UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

CARINA CALIXTO RIBEIRO DE ARAUJO

Seleção e geração de características utilizando regras de associação para o problema de ordenação de resultados de máquinas de buscas

Goiânia
2014

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

AUTORIZAÇÃO PARA PUBLICAÇÃO DE DISSERTAÇÃO EM FORMATO ELETRÔNICO

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos "reproduzir" e "publicar" conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Seleção e geração de características utilizando regras de associação para o problema de ordenação de resultados de máquinas de buscas

Autor(a): Carina Calixto Ribeiro de Araujo

Goiânia, 29 de Agosto de 2014.

Carina Calixto Ribeiro de Araujo – Autor

Thierson Couto Rosa – Orientador

CARINA CALIXTO RIBEIRO DE ARAUJO

Seleção e geração de características utilizando regras de associação para o problema de ordenação de resultados de máquinas de buscas

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Thierson Couto Rosa

Goiânia
2014

CARINA CALIXTO RIBEIRO DE ARAUJO

Seleção e geração de características utilizando regras de associação para o problema de ordenação de resultados de máquinas de buscas

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 29 de Agosto de 2014, pela Banca Examinadora constituída pelos professores:

Prof. Thierson Couto Rosa
Instituto de Informática – UFG
Presidente da Banca

Prof. Marcos André Gonçalves
Departamento de Ciência da Computação – UFMG

Prof. Humberto José Longo
Instituto de Informática – UFG

Carina Calixto Ribeiro de Araujo

Graduou-se em Ciências da Computação na UFG - Universidade Federal de Goiás. Durante a sua graduação, foi monitora no instituto de informática pelo período de um ano e meio (18 meses) nas disciplinas de Algoritmos e Programação e Introdução à Computação. Durante o mestrado, na UFG, foi bolsista da CAPES com a bolsa REUNI e posteriormente Demanda Social e realizou estágio docência por um ano na disciplina de Inteligência Artificial. Atualmente trabalha na área de recuperação da informação com o problema de ordenação de documentos.

Dedico esse trabalho ao meu namorado Victor Flávio que foi meu companheiro nas horas de solidão, irmão quando precisei chorar, conselheiro quando pensei em desistir de tudo, amigo quando precisei de apoio e psicólogo nos dias ruins.

Dedico aos meus pais Cristina Calixto e Wilson Antônio, aos meus irmãos Juliana e Rafael Calixto pelo apoio, amor, pela minha formação e meu caráter, sem eles eu não seria quem eu sou. Dedico à minha mãe por me ajudar com conselhos e experiência. E aos meus pais por acreditarem em meu potencial.

Dedico aos meus avós e aos meus tios que mesmo morando em outra cidade me apoiaram e sempre buscaram o melhor para mim, sejam com conselhos ou apenas com palavras de incentivo.

Dedico a todos meus amigos de mestrado que estavam lá presentes nos melhores e piores dias.

Agradecimentos

Desejo agradecer ao meu orientador Thierson Couto Rosa por me orientar nessa pesquisa de mestrado e acreditar em meu potencial, por ter sempre uma conduta de orientador e amigo, por auxiliar no meu crescimento profissional e pessoal durante sua orientação e pela força, incentivo e apoio necessários na realização deste trabalho.

À minha mãe, Cristina Calixto, por pacientemente me fazer companhia enquanto eu escrevia a dissertação, pela paciência em ler a dissertação e me ajudar nas correções.

Aos meus colegas de mestrado por toda ajuda e suporte nessa estrada durante o mestrado.

À CAPES por incentivar na minha pesquisa através das bolsas.

Hello, hunters. Congratulations. You have just discovered the secret message. Please send your answer to Old Pink, care of the Funny Farm, Chalfont.

Pink Floyd,
Empty Spaces.

Resumo

Araujo, Carina. **Seleção e geração de características utilizando regras de associação para o problema de ordenação de resultados de máquinas de buscas**. Goiânia, 2014. 116p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Recuperação de Informação é a área da informática que lida com o armazenamento de documentos e a recuperação de informação desses documentos. Com o advento da internet a quantidade de documentos produzidos aumentou, bem como a necessidade de recuperar a informação de forma mais mais precisa. Muitas abordagens surgiram para suprir essa requisição e uma delas é a abordagem *Learning to Rank* (L2R). Apesar de obtidos grandes avanços na precisão dos documentos retornados, ainda há espaço para melhorias. Esse trabalho de mestrado propõe a utilização de seleção e geração de características utilizando regras de associação para conseguir uma melhoria na acurácia dos métodos de L2R.

Palavras–chave

Recuperação da Informação, *Learning to Rank*, Regras de associação

Abstract

Araujo, Carina. **Feature selection and generation using assossiation rules for the ranking problem of searches machines**. Goiânia, 2014. 116p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Information Retrieval is an area of IT that deals with document storage and the information retrieval in these documents. With the advent of the Internet, the number of documents produced has increased as well as the need to retrieve the information more accurately. Many approaches have been proposed to meet these requirements and one of them is Learning to rank (L2R). Despite major advances achieved in the accuracy of retrived documents, there is still considerable room for improvement. This master thesis proposes the use of feature selection and generation using association rules to improve the accuracy of the L2R methods.

Keywords

Information retrieval, Learning to rank, Assossiation rules

Lista de Siglas e Abreviaturas

RI - Recuperação da Informação
L2R - *Learning to Rank*
LRAR - *Learning to Rank using Association Rules*
tf - *Term Frequency*
idf - *Inverse Document Frenquency*
tf-idf - *Term Frenquency - Inverse Document Frequency*
MAP - *Mean Average Precision*
NDCG - *Normalized Discounted Cumulative Gain*
CG - *Cumulative Gain*
DCG - *Discounted Cumulative Gain*
IDCG - *Ideal Discounted Cumulative Gain*
TREC - *Text REtrieval Conferenfe*
TD - *Topic Destilation*
HP - *HomePage finding*
NP - *Name Page finding*
URL - *Uniform Resource Locator*
MEDLINE - *Medical Literature Analysis and Retrieval System Online*
SIGIR - *Special Interest Group on Information Retrieval*
SVM - Support Vector Machine

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

CAPÍTULO 1

Introdução

1.1 Contextualização

O interesse em se ordenar e classificar documentos, imagens, vídeos e quaisquer outros tipos de informação não é recente. Muito antes dos computadores sequer existirem já havia a necessidade de se classificar a informação. O início da arquivística não é conhecido, apesar de ser datada anterior a 3000 AC quando os Sumérios guardavam seus documentos formados por placas de argila em locais especialmente designados a fim de evitar roubos e facilitar futura busca pela informação.

A necessidade de armazenar e buscar uma informação de forma mais precisa foi intensificada ao longo dos séculos. Após utilização do papel para o armazenamento de informações vieram as técnicas de arquivamento que até hoje são utilizadas.

Em 1945, foi publicado o artigo "*As We May Think*" do Dr. Vannevar Bush [6], o qual introduziu a ideia de armazenar grandes quantidades de informações em sistemas computacionais de modo a atingir muitas pessoas ao mesmo tempo e com alta velocidade. Assim, com a criação do computador, naturalmente muitos documentos passaram a ser armazenados digitalmente, conseguindo desta forma diminuir o espaço utilizado para se armazenar e melhorar a velocidade de acesso à informação armazenada.

Durante a década de 50 surgiram vários trabalhos que elaboraram a ideia básica da busca de informação de forma automática utilizando-se de um computador. O termo "Recuperação da informação" foi descrito pela primeira vez por Moores em 1951 [36]. A definição de recuperação da informação cunhada por Moores foi a seguinte:

Definição 1.1 *Recuperação da informação é o nome para o processo ou método pelo qual um usuário de informação em potencial é capaz de converter a sua necessidade de informação em uma lista de citações a documentos armazenados que contêm informações úteis para ele. É a constatação ou processo de descoberta com relação às informações armazenadas.*

Um dos trabalhos mais significativos em RI na década de 50 foi escrito por Luhn [30] em que ele descreve um modelo estatístico para automatizar o armazenamento e

a busca de informações. Outra obra igualmente importante foi a de Maron et al [34] em que testaram uma abordagem em que cada documento de uma coleção recebia um escore (pontuação) indicando a sua relevância para uma dada consulta. Os documentos eram então ordenados e os melhores eram retornados ao usuário. Na abordagem testada foram atribuídas manualmente palavras-chave para duzentos documentos e pesos foram determinados para cada uma das palavras-chave dependendo da importância dessas palavras no documento. Ainda em 1958, Luhn descreve pela primeira vez o conceito de *term frequency* (tf) em [31] sugerindo que a frequência da ocorrência das palavras em um documento resulta em uma métrica útil para medir a significância das palavas.

Na década de 1960 foram realizados diversos estudos para melhorias em sistemas de recuperação da informação. Salton criou e liderou um grupo de RI na Universidade de Harvard e posteriormente na Universidade de Cornell. Este grupo produziu diversos relatórios técnicos e o sistema SMART [43], os quais estabelecem conceitos e ideias que são utilizados até a atualidade. O Sistema SMART possibilitou também a melhoria na qualidade de busca. Outro estudo importante dessa década foi a sugestão de Maron e Kuhns [33] de como atribuir pesos aos termos em modelos vetoriais.

Um conceito que foi estabelecido pelo grupo de Salton foi o modelo de espaço vetorial. Essa abordagem considera documentos e consultas como vetores em um espaço dimensional de tamanho N (sendo N a quantidade de termos únicos presentes na coleção de documentos) e posteriormente é calculada a similaridade entre o documento e a consulta a fim de identificar a relevância do documento para aquela consulta. A similaridade entre um documento e uma consulta são medidos pelo cosseno do ângulo entre esses vetores [43].

Entre outros avanços na recuperação da informação realizados nesse período incluem o agrupamento de documentos com conteúdo similar e a associação de termos semanticamente similares a fim de aumentar a quantidade de documentos mostrados ao usuário por meio da expansão da consulta com variações léxicas.

Uma inovação também realizada na década de 1960 foi a introdução do conceito de *feedback* de relevância. A ideia visa envolver o usuário na saída das consultas a fim de melhorar o resultado final do conjunto retornado. O usuário dá sua opinião sobre a relevância dos documentos inicialmente retornados e o sistema utiliza essa informação dada pelo usuário para melhorar o processo de ordenação de documentos.

Na década de 70 o conceito de tf introduzido por Luhn foi complementado por Spärck Jones dando a contribuição do *inverse document frequency* ou idf [46]. Ainda durante essa década houve o crescimento dos estudos no modelo vetorial.

Nas décadas subsequentes houve o grande avanço em modelos probabilísticos e na conscientização da comunidade acadêmica de recuperação da informação em criar coleções a fim de permitir comparação entre modelos e facilitar testes de sistemas de RI.

A partir da década de 90 surgiram grandes avanços em técnicas de aprendizado de máquina juntamente com o crescimento da web. As técnicas de aprendizado de máquina são aquelas que visam desenvolver algoritmos que permitam o computador a aprender a partir de um conjunto de dados e desta forma executar tarefas com melhor desempenho. Aos sistemas de recuperação de informação que ordenam documentos utilizando aprendizado de máquinas dá-se o nome de sistemas de "*learning to rank*" (L2R).

As características das buscas na web são que, em sua maioria, os usuários utilizam dois termos em uma consulta e verificam apenas os vinte primeiros resultados [47]. Desta forma, percebe-se que normalmente uma consulta conterá poucos termos e que há a necessidade de que os documentos mais relevantes ocupem as vinte primeiras posições da lista de documentos retornados. O crescimento da web trouxe o aumento da quantidade de documentos e consequentemente a utilização dos sistemas de *learning to rank* tomaram maior destaque.

1.2 Learning to Rank - L2R

Nas últimas décadas cada vez mais tecnologias de aprendizado de máquina têm sido utilizadas para criar um modelo de ordenamento e uma nova área chamada de *learning to rank* tem gradualmente emergido. Nos últimos anos as áreas de pesquisa em L2R têm sido as mais ativas na recuperação da informação.

Métodos *learning to rank* são aqueles que utilizam tecnologias de aprendizado de máquina para resolver problemas de ordenação de documentos. Alguns exemplos utilização de L2R são: sistemas com feedback de relevância e o ajuste automático de parâmetros em modelos de recuperação da informação.

Assim como outros métodos de aprendizado de máquina, os algoritmos de L2R precisam ser alimentados com dados (ou experiências) para que o computador possa aprender e resolver os problemas. Em L2R a base de dados utilizada para aprender é chamada de conjunto de treino. O conjunto de treino é utilizado pelo sistema para aprender um modelo. Esse modelo pode ou não conter ajustes de parâmetros realizados através do conjunto de validação. Após realizada a etapa de aprendizagem, é de praxe verificar a qualidade das predições realizadas pelo sistema de L2R, para tanto é utilizado o conjunto de teste aplicado ao modelo aprendido durante a fase de treino.

Vários métodos de abordagens L2R foram propostos. Alguns se utilizam de redes neurais [3], programação evolutiva (ou programação genética) [15] e máquina de vetores de suporte (SVM) [24] [51]. Em trabalhos mais recentes, há também algoritmos baseados em regras de associação em nível de consulta [1].

1.2.1 Regras de Associação

Algoritmos baseados em regras de associação encontram regras no documento de treino. Regras de associação são definidas como uma implicação da forma $A \rightarrow C$, onde A é o antecedente e C o consequente. Nas abordagens L2R utilizando regras de associação o antecedente é um conjunto de características de um documento. O consequente é um grau de relevância r_i, para os problemas de ordenação de documentos ou, para os problemas de classificação de documentos, o consequente é uma classificação. A figura 1.1 ilustra exemplos de regras de associação para o problema de classificação e ordenação de documentos

$\{C_1 \cap C_2 \cap \ldots \cap C_n\} \rightarrow$ Relevância Ex: {Café ∩ Farinha ∩ Ovo } → 1

$\{C_1 \cap C_2 \cap \ldots \cap C_n\} \rightarrow$ Classificação {Bola ∩ Partida ∩ Vence ∩ Futebol } → Esporte

Figura 1.1: *Modelo e exemplo de regras de associação*

Uma regra de associação possui também medidas de avaliações úteis como *suporte* e *confiança*. O suporte de uma regra $\sigma(A \rightarrow C)$ é o número de casos (relativo ou absoluto) em que uma regra é correta, ou seja, em que a presença do antecedente coincide com a presença do consequente. O suporte quando absoluto, consiste apenas de uma contagem da ocorrência da regra. O suporte quando relativo, consiste da razão da ocorrência da regra em toda coleção, ou seja, o total de vezes que essa regra ocorre na coleção dividido pelo total de regras da coleção. Conforme fórmula 1-1:

$$\sigma(A \rightarrow C) = \frac{\text{total de ocorrências da regra}}{\text{total de regras}} \tag{1-1}$$

A confiança de uma regra $\theta(A \rightarrow C)$ é definida como a probabilidade condicional de um consequente dado o seu antecedente, ou seja, o total de vezes que essa regra ocorre dividido pelo total de regras em que o antecedente ocorre. A confiança pode ser interpretada como a probabilidade condicional de encontrar o consequente dado que o antecedente ocorreu, ou seja $P(C|A)$. A confiança pode ser representada conforme a fórmula 1-2:

$$\theta(A \rightarrow C) = \frac{\text{suporte da regra}}{\text{suporte do antecedente } A} \tag{1-2}$$

Para os problemas de ordenação de documentos, o conjunto de treino formula um modelo R contendo regras da forma $f_1 \wedge f_2 \wedge f_3 \wedge \ldots \wedge f_T \rightarrow r_i$. Essas regras contêm características dos documentos no antecedente e um grau de relevância no consequente. Uma vez que o modelo é construído, as regras serão utilizadas para estimar a relevância dos documentos de teste.

Na abordagem que utiliza regras de associação como técnica de ordenação de documentos chamada de LRAR (*Learning to Rank using Association Rules*) [48] não é aprendida uma função de ordenação. O trabalho gera regras de associação em tempo de consulta e as utiliza para ordenar documentos. A ordenação de um documento é realizada da seguinte forma: para cada documento do conjunto de teste, acumula-se a confiança da regra de acordo com seu grau de relevância formando-se um escore para aquela relevância.

Para a classificação de documento como relevante ou irrelevante, o grau de relevância que possuir maior escore será a classificação do documento. Na ordenação, o escore para o grau de relevância 0 será sempre descartado, considerando-se apenas a soma dos escores restantes. Os documentos são, então, ordenados em forma decrescente desta soma.

Porém, qualquer outra medida de avaliação de regra de associação poderia ser acumulada nesse escore como lift, suporte ou outra. Como ilustrado na figura 1.2.

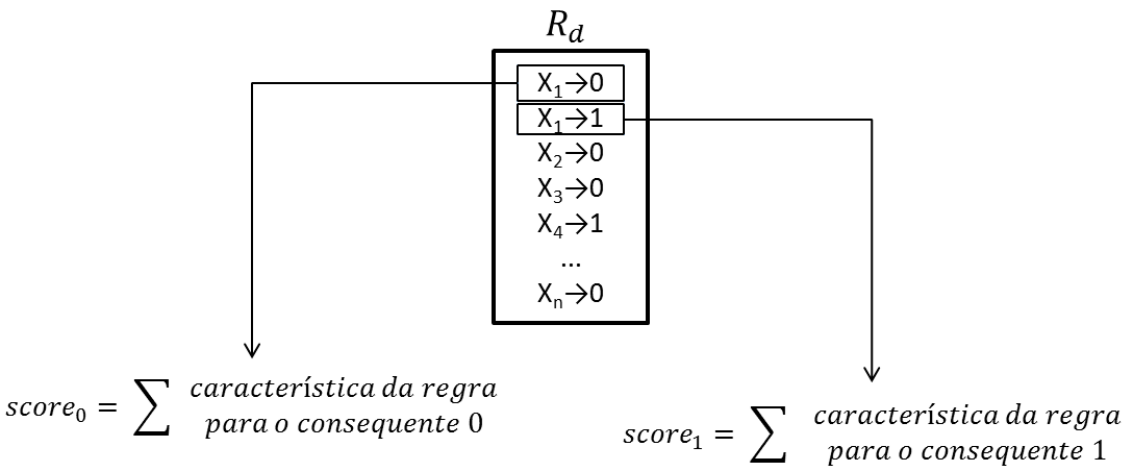


Figura 1.2: *Formação do escore em regras de associação*

Isso leva à primeira questão de pesquisa trabalhada nessa dissertação.

Questão de pesquisa 1 *A utilização das diferentes medidas de avaliação de regras de associação no algoritmo de ordenação LRAR pode trazer benefícios à acurácia dese algoritmo?*

1.2.2 Seleção e geração de características

Em RI, características (em inglês *features*) são valores numéricos que representam atributos de um documento, alguns exemplos são: frequência de termos em um documento, número de termos do documento e tamanho do título do documento.

Quando o número de características é grande, algoritmos que utilizam de aprendizado de máquina possuem maior facilidade em aprender bons modelos que aqueles

tradicionais como, por exemplo, BM25 [13]. Porém, aprender com muitas características ou com as características erradas pode, além de ser custoso, produzir um modelo pior. Sendo assim, é importante realizar uma seleção de características.

Em aprendizado de máquina, seleção de características é a tarefa de separar um subconjunto de características, para que apenas essas sejam consideradas pelo algoritmo de aprendizagem, a fim de produzir melhores resultados na classificação ou ordenação de documentos.

No problema de classificação de documentos, a seleção de características pode ser divididas em três abordagens: abordagem de filtro, abordagem *wrapper* e abordagem *embedding* [21]. Na abordagem de filtro a seleção de características se dá como um passo de pré-processamento em que um subconjunto das características é escolhido independente da métrica a ser otimizada. Na abordagem *wrapper* a seleção de características se dá escolhendo o subconjunto que produz o melhor resultado considerando uma métrica escolhida. A abordagem *embedding* embute o processo de seleção de características no processo de aprendizado.

Porém, essas abordagens são utilizadas em problemas de classificação de documentos e os problemas de ordenação comportam-se de forma diferente. Em problemas de classificação, por exemplo, a precisão e a revocação são igualmente importantes, já em problemas de ordenação a precisão é mais importante que a revocação. Sendo assim, os métodos existentes para seleção de características em problemas de classificação não são ideais para os problemas de ordenação de documentos, tornando necessária a criação de novos métodos de seleção de característica para tais problemas.

A área de aprendizagem de máquina que se preocupa com a geração de característica (ou indução construtiva) estuda métodos que dão aos computadores a capacidade de modificar ou melhorar a representação contida nas bases de dados. Técnicas que utilizam da geração de características procuram novas características que possam descrevem os conceitos alvo melhor do que os atributos advindos da base de dados sendo capazes aumentar a acurácia dos métodos de ordenação.

Desta forma tem-se a segunda questão de pesquisa averiguada no presente trabalho.

Questão de pesquisa 2 *A expansão da base de dados com novas características obtidas a partir de regras de associação podem trazer benefícios para os métodos de L2R existentes?*

1.3 Objetivos geral e específicos

O objetivo geral desse trabalho foi investigar o uso de regras de associação no problema de ordenação de documentos. Essa investigação foi conduzida com base em

dois objetivos específicos, descritos a seguir.

1.3.1 Primeiro objetivo específico

O primeiro objetivo foi o de avaliar o desempenho de outras medidas de avaliação de regras de associação (em adição às medidas de suporte e confiança) quando utilizadas em um algoritmo de ordenação de documentos semelhante ao algoritmo descrito na seção 1.2. Esse algoritmo é o *Learning to Rank using Association Rules* (LRAR) proposto por Veloso et al. em [48].

O referido algoritmo gera regras de associação específicas para cada documento de uma consulta de teste, ou seja, as regras de associação são computadas sob demanda. Com isso, evita-se a geração de um grande conjunto de regras que se obteria se a geração de regras fosse feita durante uma fase prévia de treinamento. O algoritmo LRAR é apresentado na seção 4.1.1.

1.3.2 Segundo objetivo específico

O segundo objetivo específico é o de investigar e avaliar o uso de conjuntos de características (*features*) que aparecem como antecedentes em regras de associações relevantes como novas características no processo de ordenação.

A motivação para esse objetivo é a seguinte. Seja uma regra x do tipo $f_1 \wedge f_2 \wedge \ldots \wedge f_n \rightarrow r_i$, onde r_i corresponde a um valor de um conjunto discreto de valores de relevância que podem ser atribuídos a um documento. Considere que a regra x é importante para determinar o grau de relevância r_i de um documento em relação a uma dada consulta, de acordo com alguma medida de avaliação de regra (suporte, confiança, ou outra). Nesse casso, a coocorrência das características $f_1, f_2, \ldots, f_n$ é importante para caracterizar o grau de relevância r_i para o documento. Logo, essa coocorrência de $f_1 \wedge f_2 \wedge \ldots \wedge f_n \rightarrow r_i$ pode ser vista como uma nova característica para o documento, em adição às características já existentes.

O valor da medida de avaliação de regra usada para determinar a importância da regra pode ser usado como o peso dessa nova característica. O segundo objetivo corresponde à geração de novas características para os conjuntos de treino e de teste e avaliação dessas destas novas características. O novo conjunto de características (formado pelas características existente e pela novas) pode ser utilizado em algoritmos tradicionais de L2R, visando melhorar suas eficácias.

Os dois objetivos específicos consistem, portanto, em propor e avaliar soluções para as questões de pesquisa previamente discutidas:

Questão de pesquisa 1: *a utilização das diferentes medidas de avaliação de regras de associação no algoritmo de ordenação LRAR pode trazer benefícios à acurácia desse algoritmo?*

Questão de pesquisa 2: *a expansão da base de dados com novas características obtidas a partir de regras de associação podem trazer benefícios para os métodos de L2R existentes?*

1.4 Principais contribuições

O presente trabalho tem as seguintes contribuições para o problema de ordenação de documentos:

- Comprovar a eficácia da utilização de regras de associação para os problemas de ordenação de documentos.
- Demonstrar que a partir da geração de novas características é possível trazer benefícios para os métodos de L2R existentes, desde que a característica gerada tenha qualidade para ordenação.

1.5 Organização da Dissertação

O capítulo 2 apresenta uma revisão bibliográfica dos conceitos e termos utilizados no decorrer dessa dissertação. Nesse capítulo são expostos conceitos básicos em recuperação da informação, aprendizado de máquina e *learning to rank*. No capítulo 3 são apresentados os trabalhos que possuem correlação direta com o tema abordado por este trabalho. No capítulo 4 são apresentadas todas as abordagens de seleção e geração de características que foram propostas e avaliadas. O capítulo 5 possui os resultados alcançados por algoritmos de ordenação de documentos conceituados com a utilização das características selecionadas ou geradas propostas por esse trabalho. Por último o capítulo 6 apresenta as conclusões acerca dos resultados obtidos e ao final do capítulo propõe-se possíveis trabalhos futuros.

CAPÍTULO 2

Conceitos Básicos

Nesse capítulo são abordados os conceitos básicos necessários para as definições relacionadas ao entendimento da dissertação. Na primeira seção são descritos os conceitos relacionados a recuperação da informação. Primeiro é descrito um sistema de informação típico, posteriormente são definidos os conceitos de documento, consultas e relevância. Em uma subseção são mostrados os principais modelos de representação de documentos e consultas em sistemas de RI. Em seguida, é discorrido sobre as medidas de avaliação de performance de um sistema de reconhecimento da informação. Posteriormente, são apresentados os conceitos relacionados a aprendizado de máquina e *Learning to Rank*. Por último, é apresentado o conceito de coleções de referência e a coleção LETOR que foi utilizada nesse trabalho.

2.1 Recuperação da Informação

A recuperação de informação é um conceito simples. Suponha que haja uma livraria e que um usuário da loja formule um pedido ou consulta de acordo com a necessidade de informação desejada. A resposta para tal pedido é um conjunto de livros que satisfaçam essa necessidade de informação expressa por sua pergunta. Esse usuário poderia ler todos os livros da loja, mantendo aqueles que considera relevante e descartando todos os outros. A isso chamamos de recuperação "perfeita", pois todos os livros relevantes seriam mantidos e além disso nenhum livro irrelevante seria mantido. Porém, esta solução é impraticável, pois um usuário não tem ou, muitas vezes, não deseja passar grande tempo lendo toda uma coleção de livros para separar apenas aqueles que considera relevante.

Sendo assim, de acordo com [32], recuperação da informação (RI) é a tarefa de encontrar materiais (normalmente documentos) de natureza não estruturada (geralmente em formato textual) que satisfaçam uma necessidade de informação de um usuário dentro de uma coleção armazenadas em computadores.

Como mostra a figura 2.1, um sistema de recuperação de informação é uma tríade composta por: entradas, processamento de recuperação de informação e saída. A

entrada de um sistema de RI é composta por um conjunto de documentos e por consultas. Em uma primeira fase do funcionamento, o sistema de RI lê apenas os documentos e cria uma representação interna para os mesmos. Em seguida, o sistema passa a aceitar consultas. Para cada consulta de entrada, o sistema pesquisa os documentos lidos na primeira fase e identifica aqueles documentos que são relevantes à consulta. Em seguida, o sistema gera uma ordenação dos documentos, segundo a relevância dos mesmos em relação à consulta. A saída do sistema consiste nessa ordenação, a qual denominamos *ordenação por relevância ou ranking*. Um sistema de RI é considerado online se nele for possível ao usuário selecionar os documentos que ele considera mais relevantes em uma ordenação por relevância previamente gerada pelo sistema. Essa reordenação dos documentos feita pelo usuário pode ser utilizada como aprendizado pelo sistema para melhorar a recuperação posterior de documentos. A esse procedimento dá-se o nome de realimentação ou *feedback* [40].

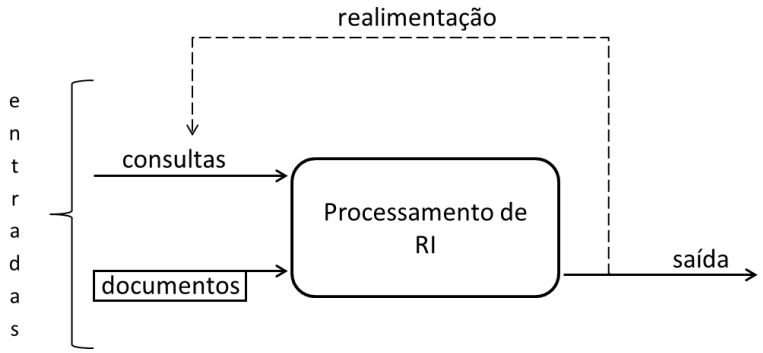


Figura 2.1: *Um sistema de RI típico*

2.1.1 Documento, consulta e relevância

Em RI o termo *documento* corresponde a um texto ou qualquer porção de um texto, podendo ser uma frase, um parágrafo, uma página de livro ou artigo, um livro ou até mesmo uma coleção de livros. Ao grupo de documentos que o sistema de RI utiliza como entrada para realizar a recuperação de informação damos o nome de "*coleção*" ou "*corpus*". [32]

Para os usuários o processo de recuperação de informação parte de uma necessidade de informação. Essa necessidade de informação é passada ao sistema através de uma consulta formulada pelo usuário. Necessidade de informação é um conceito bastante diferente de uma consulta.

A necessidade de informação é o tema sobre o qual o usuário deseja ter mais conhecimento, enquanto uma consulta (ou em inglês "*query*") é a forma com que o usuário converte a sua necessidade de informação a um modelo que o computador compreenda e possa realizar a tarefa de recuperação da informação. [32]

Digamos que um usuário deseje saber quais peças de Shakespeare contêm as palavras “Brutus” e “Caesar” e não possua a palavra “Calpurnia”. Um exemplo de consulta para tal necessidade de informação poderia ser “Brutus AND Caesar AND NOT Calpurnia”. Muitas vezes, por inexperiência ou desatenção, um usuário pode formular uma consulta não muito representativa à sua necessidade de informação e esse é um dos problemas enfrentados por sistemas de RI.

Relevância é uma característica atribuída ao documento de acordo com a necessidade de informação de um usuário. Sendo assim, um documento é considerado relevante se ele contiver a informação que supre a necessidade de informação pessoal de um usuário. O problema maior enfrentado pelos sistemas de RI em relação à relevância é que esse conceito é dependente da necessidade de informação e da noção específica de relevância do usuário que necessita da informação e, portanto, não pode ser explicitamente formalizado. [2]

Portanto, a ordenação por relevância gerada por um sistema de RI corresponde a uma ordenação segundo uma suposição do sistema quanto à relevância de cada documento em relação à consulta correspondente e pode não coincidir com aquela considerada pelo usuário.

Em um sistema de RI não é comum se armazenar o texto completo de um documento ou tratar uma consulta pela linguagem natural, pois não é uma tarefa fácil para o computador. O principal desafio é obter uma representação de cada documento e consulta tal que seja possível o computador utilizar-se desses dados. Existem diferentes modelos para a representação dos documentos e consultas. Alguns sistemas de RI, por exemplo, tratam os documentos e as consultas como conjuntos de palavras-chave, desconsiderando a sequência com que as palavras aparecem nas consultas ou documentos. Os sistemas de RI adotam modelos de representação de documentos e consultas com o objetivo de atribuir um valor de importância a um documento em relação a uma consulta que sejam o mais semelhante possível ao valor subjetivo de relevância atribuído pelo usuário ao documento em relação à mesma consulta.

2.1.2 Modelos de Representação

Em sistemas de RI, existem várias estratégias de se recuperar os documentos mais relevantes. Cada uma dessas estratégias possue um modelo específico de representação dos documentos. Essas diferentes representações dos documentos são realizadas para adequar o documento à estratégia. Os três modelos clássicos de representação são: Booleano, Vetorial e Probabilístico. No modelo booleano, documentos e consultas são representados como um conjunto de termos. No modelo vetorial, documentos e consultas são representados como vetores em um espaço n-dimensional. Assim, diz-se que o mo-

delo é vetorial. No modelo probabilístico, as representações para modelar o documento e a consulta são baseados em teorias probabilísticas. Assim, diz-se que o modelo é probabilístico [2].

Antes de discutir sobre os modelos clássicos faz-se necessário definir formalmente um modelo de recuperação de informação. Segue abaixo a definição de acordo com [2].

Definição 2.1 *Um modelo de representação de recuperação de informação é uma quádrupla* $[C, Q, \mathcal{F}, R(q_i, d_j)]$ *onde:*

1. *C é um conjunto composto de visões lógicas ou representações para os documentos da coleção.*
2. *Q é um conjunto composto de visões lógicas ou representações para a necessidade de informação do usuário. Tal representação é chamada de consulta.*
3. *$\mathcal{F}$ é um framework para representação da modelagem de representações dos documentos, consultas e a relação entre eles.*
4. *$R(q_i, d_j)$ é uma função de ranking que associa um valor a uma consulta $q_i \in Q$ e uma representação de documento $d_j \in D$. Tal valor define uma ordenação entre os documentos em relação à consulta q_i.*

Modelo booleano

Este modelo representa os documentos e as consultas como um conjunto de palavras ou termos. Nessa técnica de recuperação da informação, os documentos são tratados como um saco de palavras (em inglês *bag of words*). [32]

No processo de formação da consulta, são utilizadas operações booleanas com os termos desejados pela necessidade de informação. No modelo de conjunto são retornados apenas os documentos que contêm exatamente a descrição booleana da consulta. Uma descrição booleana é aquela que se utiliza da álgebra booleana, ou seja, que utiliza dos operadores lógicos (E, OU, NÃO e outros). [20]

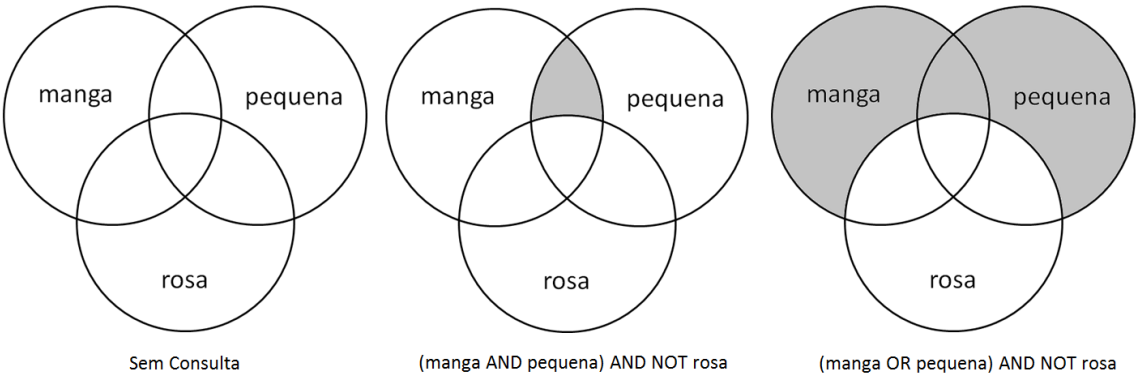


Figura 2.2: *Exemplo de consultas no modelo booleano e os conjuntos resultantes*

Por exemplo, na figura 2.2, pode-se ver os conjuntos de documentos retornados por um sistema de RI que opere com o modelo booleano. Na primeira parte da imagem tem-se os conjuntos de documento que possuem os termos “manga”, “pequena” e “rosa”. Na segunda parte tem-se, na área sombreada, o conjunto resultante da consulta “(manga AND pequena) AND NOT rosa”. Nessa área, apenas os documentos que possuírem ambos “manga” e “pequena”, mas não o termo “rosa” serão retornados como relevantes. Na terceira parte tem-se, na área sombreada, o conjunto resultante da consulta “(manga OR pequena) AND NOT rosa”. Nessa área, apenas os documentos que possuírem os termos “manga” ou “pequena”, mas não o termo “rosa” serão retornados.

Esse modelo possui alguns problemas. Muitas vezes pode não retornar as melhores informações possíveis, pois podem existir palavras homônimas, ou seja, que possuam mesma grafia, mas significam coisas totalmente diferentes, como por exemplo a palavra “canto” que pode significar o ato de cantar ou o local onde duas paredes se encontram.

Outro problema desse modelo é que exige de quem efetua a consulta um grande conhecimento de lógica booleana. Como o modelo utiliza de operadores lógicos, para obter melhores resultados é necessário que o usuário possua grande conhecimento de lógica booleana. Ou seja, é um modelo não intuitivo, pois não utiliza de linguagem natural ou apenas palavras-chave para formular uma consulta.

Além de todos esses problemas, o modelo booleano não faz distinção entre a relevância dos documentos. Se um documento possuir as palavras descritas na consulta este será relevante, caso contrário irrelevante. O modelo não considera que alguns documentos são mais relevantes que outros. Tome por exemplo a consulta “casa AND alugar AND Goiânia”. Um documento que possua as palavras “manga, pequena e rosa” é claramente não relevante para tal consulta. Porém, um documento com as palavras “casa e alugar” e não possuir a palavra “Goiânia” é considerado irrelevante para o modelo booleano e este é mais relevante que o documento anterior, pois possui mais palavras afins com a consulta. Porém, o modelo booleano considera ambos igualmente irrelevantes não fazendo distinção do grau de relevância entre cada um. Assim, esse modelo não é exatamente um modelo de recuperação de informação, mas um modelo de recuperação de dados, pois não atribui um valor de relevância aos documentos.

Modelo vetorial

Este modelo representa os documentos e as consultas como vetores, matrizes ou tuplas. Um exemplo de modelo vetorial é o modelo de espaço vetorial. Peter Luhn foi o primeiro a sugerir uma abordagem estatística para buscar informações [30]. Foi sugerido por ele que, para buscar em uma coleção de documentos, o usuário primeiramente separasse um documento que é similar ao documento que ele deseja buscar. O grau de

similaridade entre o documento separado pelo usuário e as representações do documento da coleção é utilizado para ordenar a lista de documentos retornados.

Luhn definiu o grau de similaridade como: "O quanto mais duas representações se assemelhassem nos seus elementos e suas distribuições, maior seria a probabilidade de representarem informações similares."[30]

Posteriormente Maron e Kuhns [33] foram além, sugerindo como atribuir pesos aos termos e incluindo alguns experimentos com pesos de termos.

O conceito de vetor de documento e vetor de consulta é bastante simples. Assuma que uma coleção $C = \{d_1, d_2, ..., d_{|C|}\}$ possui um conjunto $T = \{t_1, t_2, ..., t_{|T|}\}$ de termos únicos. Um documento d_i pode então ser representado por um vetor $d_i = (t_1, t_2, t_3, ..., t_{|T|})$ em que t_i terá um valor igual a 1 se o termo t_i estiver presente no documento, e um valor 0 caso contrário. Uma consulta pode ser representada da mesma maneira. A figura 2.3 abaixo mostra como é realizada essa representação.

Lista de termos da coleção: T = {fatores, informação, ajuda, humano, operação, sistemas, recuperação}	
Consulta Vetor consulta	Fatores humanos em sistemas de recuperação da informação (1, 1, 0, 1, 0, 1, 1)
Documento1 Vetor documento1	humano, fator, informação, recuperação (1, 1, 0, 1, 0, 1, 0)
Documento2 Vetor documento2	humano, fator, ajuda, sistema (1, 0, 1, 1, 0, 0, 1)
Documento3 Vetor documento3	fator, operação, sistema (1, 0, 0, 0, 1, 0, 1)

Figura 2.3: *Exemplo simples de vetor de documento e consulta*

O modelo de vetor que representa as consultas e documentos como vetores é denominado modelo vetorial e foi proposto por Salton [43]. Nele, cada dimensão do vetor corresponde à presença de um termo do documento ou da consulta, e adicionalmente ao indicador da presença de um termo no documento ou consulta há um peso associado para cada termo.

De acordo com [2] a definição de modelo vetorial é:

Definição 2.2 *O peso $p_{i,j}$ associado com o par (t_i, d_j), palavra t_i e documento d_j, é positivo e não binário. A cada termo de uma consulta também é associado um peso. O vetor para uma consulta q é definido como $\vec{V}(q) = (p_{1,q}, p_{2,q}, ..., p_{|T|,q})$, onde $p_{i,q} \geq 0$ e $|T|$ é o número total de termos do sistema. Um vetor para um documento d_j é representado por $\vec{V}(d_j) = (p_{1,j}, p_{2,j}, ..., p_{|T|,j})$, onde $p_{i,j} \geq 0$.*

No modelo vetorial, é normalmente utilizado o tf-idf (*term frequency–inverse document frequency* ou $tf\text{-}idf_{t,d}$) como peso dos termos nos vetores. Existem várias modificações de tf-idf e a mais comum é encontrada em [32].

Para definir tf-idf, deve-se primeiro entender o conceito de *term frequency* ou $tf_{t,d}$ e posteriormente o conceito de *inverse document frequency* ou idf_t. *Term frequency* é um peso que corresponde ao número de vezes que o termo está presente em um documento. É denotado por $tf_{t,d}$, onde t denota o termo e d o documento em questão [32].

Porém, a utilização apenas do $tf_{t,d}$ não é suficiente para indicar a importância do termo em um documento. Nem todas as palavras que tiverem grande ocorrência no documento podem ser consideradas igualmente importantes. Suponha que exista um livro sobre recuperação da informação. A palavra "de" pode ocorrer nesse livro muito mais vezes que a palavra "informação", porém a palavra "informação" é mais importante nesse contexto. A essas palavras que ocorrem inúmeras vezes, mas não possuem força para a ordenação de documentos dá-se o nome de *stopwords*.

Para suprir o problema das *stopwords* foi criado um mecanismo de atenuar o seu efeito. A ideia é reduzir os pesos dos termos que possuírem uma alta *document frequency* ou df. *document frequency* ou df_t é definido como o número de documentos na coleção que contêm o termo t. O df_t é utilizado na forma de *inverse document frequency* ou idf_t. Considerando $|C|$ o número de documentos em uma coleção, idf_t é definido por [32] como a seguir.

$$idf_t = \log \frac{|C|}{df_t} \tag{2-1}$$

Dessa forma o idf_t de um termo raro na coleção será alto e o idf_t de um termo muito comum será baixo. Combinando os dois pesos $tf_{t,d}$ e idf_t podemos produzir um novo peso para cada termo em cada documento. Esse novo peso é chamado *term frequency–inverse document frequency* (tf-idf) ou $tf\text{-}idf_{t,d}$ e é definido como a seguir.

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t \tag{2-2}$$

O $tf\text{-}idf_{t,d}$ dá um peso a um termo t em um documento d e se comporta da seguinte forma:

- O $tf\text{-}idf_{t,d}$ terá valor alto se o termo t ocorrer várias vezes no documento d e ocorrer em poucos documentos da coleção de documentos.
- O $tf\text{-}idf_{t,d}$ terá valor baixo se o termo t ocorrer poucas vezes no documento d ou se t ocorre em vários documentos da coleção.

No modelo de espaço vetorial, a fim de retornar os documentos relevantes a uma consulta, compara-se a semelhança entre o vetor da consulta com os vetores dos

documentos presentes na coleção. Os documentos que possuírem maior semelhança são selecionados como os mais relevantes. Na prática, a maneira mais comum de comparar a semelhança entre uma consulta e um documento é calculando o cosseno do ângulo formado entre os vetores que representam o documento e a consulta.

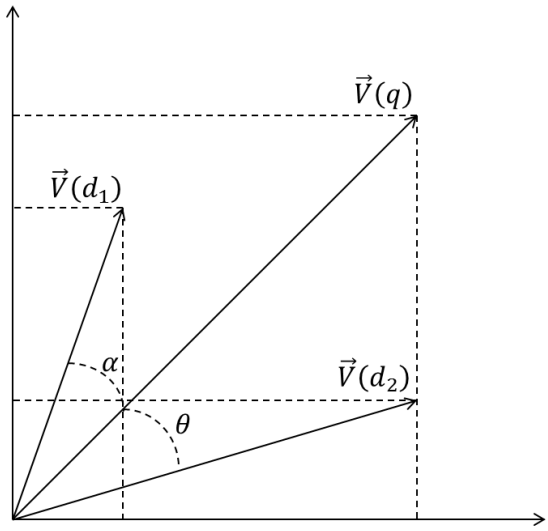


Figura 2.4: *Modelo de espaço vetorial*

A figura 2.4 mostra um exemplo de vetor-consulta $\vec{V}(q)$ e dois vetores-documentos $\vec{V}(d_1)$ e $\vec{V}(d_2)$ no modelo de espaço vetorial bidimensional, isto é, cada documento ou consulta pode ser composto por no máximo dois termos. O primeiro componente de cada vetor denota o peso do primeiro termo do vetor e correspondente à projeção do vetor no eixo x. O segundo componente correspondente ao peso do segundo termo (projeção do vetor no eixo y).

Tomando a figura 2.4 como exemplo. A fórmula utilizada para o calculo do cosseno entre o vetor-consulta $\vec{V}(q)$ e um vetor-documento $\vec{V}(d)$ é a seguinte [32]:

$$\cos\theta = \frac{\vec{V}(d) \cdot \vec{V}(q)}{|\,\vec{V}(d)\,||\,\vec{V}(q)\,|} \tag{2-3}$$

O numerador da fórmula 2-3 representa o produto escalar entre os vetores $\vec{V}(d)$ e $\vec{V}(q)$. Sendo $|T|$ o número de dimensões dos vetores, o produto escalar entre dois vetores é definido pela equação (2-4):

$$\vec{V}(d) \cdot \vec{V}(q) = \sum_{i=1}^{|T|} p_{i,d} \times p_{i,q} \tag{2-4}$$

O denominador da fórmula 2-3 representa o comprimento ou norma de cada um dos vetores. A norma de um vetor $\vec{V}(d)$ é calculada de acordo com a equação 2-5.

$$|\vec{V}(d)| = \sqrt{\sum_{i=1}^{|T|} p_{i,d}{}^2} \tag{2-5}$$

Sendo assim, a fórmula para calcular a semelhança entre dois vetores pode ser reescrita na forma:

$$sem(\vec{V}(d), \vec{V}(q)) = \frac{\vec{V}(d)}{|\vec{V}(d)|} \times \frac{\vec{V}(q)}{|\vec{V}(q)|} \tag{2-6}$$

Substituindo a fórmula 2-5 na equação anterior, tem-se:

$$sem(\vec{V}(d), \vec{V}(q)) = \frac{\vec{V}(d)}{\sqrt{\sum_{i=1}^{n} p_{i,d}{}^2}} \times \frac{\vec{V}(q)}{\sqrt{\sum_{i=1}^{n} p_{i,q}{}^2}} \tag{2-7}$$

Com a fórmula dessa forma pode-se comparar mais rapidamente um vetor consulta com vários vetores documentos, pois um dos termos da multiplicação, a porção da direita $\frac{\vec{V}(q)}{\sqrt{\sum_{i=1}^{n} p_{i,q}{}^2}}$, terá sempre o mesmo valor para uma mesma consulta.

Modelo Probabilístico

Este modelo foi introduzido por [42] e trata a recuperação da informação como um processo probabilístico. Os modelos probabilísticos são utilizados para ordenar os documentos em ordem decrescente de probabilidade de relevância de acordo com uma consulta formulada pelo usuário [12]. Serão abordados nessa seção apenas os elementos chaves para o entendimento do modelo probabilístico.

A ideia fundamental do modelo probabilístico é a seguinte. Dada uma consulta formulada pelo usuário, há um conjunto de documentos que contêm exatamente todos os documentos relevantes. Esse conjunto é chamado de conjunto resposta ideal [2].

Para determinar tal conjunto é necessário conhecer as características que o definem. Porém, essas características não são conhecidas em tempo de consulta, assim deve ser feito um processo para inicialmente determinar que características são essas.

O modelo probabilístico é baseado na definição 2.3 abaixo:

Definição 2.3 *Dada uma consulta q e um documento d_j da coleção, o modelo probabilístico tenta estimar a probabilidade de um usuário considerar o documento d_j relevante à consulta q. O modelo probabilístico assume que essa probabilidade de relevância depende apenas das representações dos documentos e da consulta. O modelo assume que haverá dois possíveis conjuntos:*

1. *R que é o conjunto com todos os documentos relevantes à consulta q;*
2. *$\overline{R}$ que é o conjunto com todos os documentos não relevantes à consulta q.*

O modelo probabilístico então calcula a probabilidade de um usuário considerar o documento d_j relevante à consulta como a razão $\frac{P(R|\vec{V}(d_j))}{P(\overline{R}|\vec{V}(d_j))}$. Posteriormente utiliza as probabilidades como *ranking* dos documentos visando minimizar a chance de julgamentos errôneos [41].

Os pesos dos termos de um documento e de uma consulta, em um modelo probabilístico, são binários, por exemplo $p_{i,j}$ (peso do termo i no documento j) e $p_{i,q}$ (peso do termo i na consulta q). Com o peso $p_{i,j} = 1$ se o termo i estiver presente no documento j e $p_{i,j} = 0$ caso contrário [18]. A consulta q é um subconjunto do conjunto de termos. Seja $P(R|\vec{V}(d_j))$ a probabilidade que o documento d_j seja relevante para q e $P(\overline{R}|\vec{V}(d_j))$ a probabilidade que o documento d_j seja não relevante para q. A medida de similaridade, ou relevância, de um documento d_j a uma consulta q é definida como a razão:

$$sim(d_j, q) = \frac{P(R|\vec{V}(d_j))}{P(\overline{R}|\vec{V}(d_j))} \tag{2-8}$$

Utilizando-se do Teorema de Bayes:

$$sim(d_j, q) = \frac{P(\vec{V}(d_j)|R) \times P(R)}{P(\vec{V}(d_j)|\overline{R}) \times P(\overline{R})} \tag{2-9}$$

De acordo com a equação (2-9):

- $P(\vec{V}(d_j)|R)$ é a probabilidade condicional de se selecionar aleatoriamente o documento d_j dentro do conjunto de documentos relevantes R.
- $P(\vec{V}(d_j)|\overline{R})$ é a probabilidade condicional de se selecionar o documento d_j dentro do conjunto de documentos não relevantes $\overline{R}$.
- $P(R)$ é a probabilidade de que um documento selecionado aleatoriamente dentro da coleção total seja relevante.
- $P(\overline{R})$ é a probabilidade de que um documento selecionado aleatoriamente dentro da coleção total seja não relevante.

Como $P(R)$ e $P(\overline{R})$ possuem valores fixos para todos os documentos da coleção, a equação 2-9 pode ser reescrita como a seguir:

$$sim(d_j, q) \sim \frac{P(\vec{V}(d_j)|R)}{P(\vec{V}(d_j)|\overline{R})} \tag{2-10}$$

Assumindo que haja independência entre os termos da coleção [25],

$$sim(d_j, q) \sim \prod_{i=1}^{|T|} \frac{P(t_i|R)}{P(t_i|\overline{R})} \tag{2-11}$$

Onde $|T|$ é igual ao total de termos distintos da coleção de documentos e t_i corresponde a cada um dos termos distintos da coleção. $P(t_i|R)$ é a probabilidade condicional de haver um termo t_i dado que o documento é relevante e pode ser obtida dividindo-se o total de documentos que possuem o termo t_i e são relevantes pelo total de documentos relevantes. $P(t_i|\overline{R})$ é a probabilidade condicional de haver um termo t_i dado que o documento é não relevante e pode ser obtida dividindo-se o total de documentos que possuem o termo t_i e são não relevantes pelo total de documentos não relevantes.

As principais vantagens do modelo probabilístico são [2, 9]:

- Facilidade na ordenação dos documentos de acordo com seu grau de relevância. No modelo probabilístico, os documentos são ordenados de forma decrescente de acordo com a probabilidade de relevância.
- Maior precisão (ver subseção 2.1.3) na recuperação de documentos em relação aos outros modelos de representação clássicos.

As principais desvantagens do modelo probabilístico são [2, 9]:

- Necessidade de conhecer previamente a separação de documentos da coleção em dois subconjuntos (documentos relevantes R e documentos não relevantes $\overline{R}$) através de hipótese.
- O método probabilístico clássico não utiliza a frequência dos termos no documento, utilizando-se apenas de pesos binários.
- Assume a total independência dos termos nos documentos.

Um exemplo de modelo probabilístico mais utilizado e com maior sucesso que o modelo probabilístico clássico é o BM25 [32]. Sua fórmula mais comum, apresentada na equação 2-12, faz o uso da frequência dos termos no documento ($tf_{t,d}$) e leva em consideração o tamanho dos documentos. Dada uma consulta q com T_Q termos.

$$sim(d_j,q) = \sum_{i=1}^{|T_Q|} idf_t(q_i) \times \frac{tf(q_i,d_j) \times (k_1+1)}{tf(q_i,d_j) + k_1 \times (1 - b + b \times \frac{|T|}{avgDoclen})} \tag{2-12}$$

Onde Q é a quantidade de termos da consulta q. q_i é o i-ésimo termo de uma consulta, $tf(q_i,d_j)$ é o *term frequency* do termo q_i no documento d_j. $|T|$ é o tamanho do documento, expresso pelo total de termos de d_j, e $avgDoclen$ é a média de tamanho dos documentos da coleção. k_1 e b são parâmetros livres, porém comumente escolhidos como $k_1 \in [1.2, 2.0]$ e $b = 0.75$ [32].

2.1.3 Medidas de avaliação de eficácia de um sistema de RI

Para medir a eficácia de um sistema de RI, ou seja, a qualidade de um sistema de RI, o usuário irá precisar das medidas de avaliação de eficácia de um sistema de RI. As medidas mais importantes são: precisão (*precision*), revocação (*recall*), precisão média (*average precision*), MAP (*Mean Average Precision*) e NDCG (*Normalized Discounted Cumulative Gain*).

Precisão

Precisão é a fração dos documentos retornados que são relevantes ao usuário. A equação 2-13 define o cálculo de precisão de um sistema de RI [2, 32]:

$$precisão = \frac{DocRelRet}{DocRet} = P(RelevRet|Ret) \tag{2-13}$$

Onde *DocRelRet* é o número de documentos relevantes retornados pelo sistema de RI, *DocRet* é a quantidade total de documentos retornados pelo sistema de RI. *RelevRet* é o conjunto dos documentos relevantes e retornados e *Ret* é o conjunto de documentos retornados. A figura 2.5 abaixo ilustra esses conceitos.

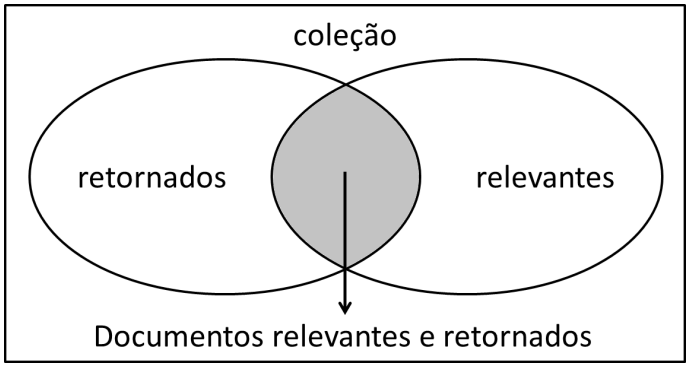


Figura 2.5: *Conjuntos dos documentos retornados e relevantes*

Por exemplo, suponha que uma lista de documentos retornados por um sistema de RI tenha tamanho 10. Suponha também que apenas 5 documentos dessa lista foram relevantes. Assim, a precisão desse sistema de RI de exemplo será de $0,5$.

Precisão em um ponto

A precisão em um ponto, ou precisão em n, mede a relevância dos n primeiros documentos de uma lista ordenada. Sua fórmula é expressa como [2, 32]:

$$P@n = \frac{DocRelRet_n}{n} \tag{2-14}$$

Onde $DocRelRet_n$ é o número de documentos relevantes retornados até a posição n e n é a posição da lista a que se deseja calcular a precisão.

Por exemplo, suponha que em uma lista de documentos retornados por um sistema de RI o segundo documento é relevante, porém o primeiro não é relevante. Assim, a precisão da lista no segundo documento será de $0,5$ pois há 1 documento relevante em 2 documentos retornados.

Revocação (*recall*)

A revocação é a fração dos documentos relevantes que foram retornados do total de documentos relevantes. A equação 2-15 mostra como calcular a revocação de um sistema de RI [2, 32]:

$$revocacão = \frac{DocRelRet}{DocRel} = P(RelevRet|Rel) \tag{2-15}$$

Onde $DocRelRet$ é o número de documentos relevantes retornados pelo sistema de RI, $DocRel$ é a quantidade total de documentos relevantes. $RelevRet$ é o conjunto dos documentos relevantes e retornados e Rel é o conjunto de documentos relevantes. A figura 2.5 da subseção anterior ilustra esses conceitos.

Por exemplo, suponha que, para uma consulta realizada ao sistema de RI, haja 200 documentos relevantes na coleção, mas apenas 100 documentos relevantes foram retornados, assim a revocação nesse sistema de RI de exemplo seria $0,5$.

A figura 2.6 abaixo mostra as relações entre revocação, precisão, falsos positivos e negativos e verdadeiros positivos e negativos.

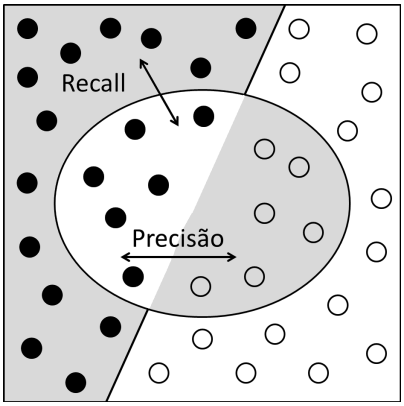


Figura 2.6: *Relação entre precisão e revocação.*

A porção da esquerda (bolas pretas) são todos os documentos relevantes, enquanto a porção da direita (bolas brancas) são os documentos não relevantes. A região correspondente ao círculo interno à figura corresponde ao documentos retornados pelo sistema. As bolas pretas dentro do círculo correspondem verdadeiros positivos, isto é,

documentos ordenados como relevantes pelo sistema e que são realmente considerados relevantes. As bolas não pretas dentro do círculo correspondem a falsos positivos, documentos que o sistema retornou como sendo relevantes quando na verdade não são. As bolas brancas fora do círculo correspondem a verdadeiros negativos: o sistema não os ordenou como sendo relevantes e realmente não são relevantes. As bolas pretas externas ao círculo correspondem aos falsos negativos, documentos relevantes mas que o sistema não os relacionou na ordenação produzida.

Um valor alto de revocação significa que o sistema de RI retornou a maioria dos documentos relevantes, minimizando assim os falsos negativos.

Um valor alto de precisão significa que o sistema de RI retornou uma pequena quantidade de documentos não relevantes, minimizando o número de falsos positivos.

Precisão Média

Precisão média é calculada como a média das precisões em um ponto para cada documento relevante de uma lista. Ou seja, dada uma lista de documentos recuperados para uma consulta, calcula-se a precisão em um ponto para cada documento relevante dessa lista e posteriormente realiza-se a média aritmética dessa $P@n$. A fórmula abaixo ilustra o cálculo da precisão média [2, 32]:

$$Pm = \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{DocRelRet} \tag{2-16}$$

Ná fórmula acima, n é o total de documentos retornados pelo sistema de RI, $P(k)$ é a precisão em um ponto k no documento, $rel(k)$ é um indicador binário da relevância do documento k, ou seja, se o documento k for relevante $rel(k) = 1$ e $rel(k) = 0$ caso contrário; e $DocRelRet$ é a quantidade de documentos relevantes retornados pelo sistema de RI a uma consulta.

MAP

É comum avaliar um sistema de RI utilizando-se várias consultas para avaliar sua eficácia em diferentes tipos de consultas. O MAP (*Mean Average Precision*) é uma medida de eficácia que visa resumir em um único valor as precisões médias de cada consulta. MAP é calculado como uma média aritmética das precisões médias. Dessa forma, para mensurar a qualidade de um sistema de RI utilizando-se o MAP, calculam-se as precisões média de várias consultas e posteriormente realiza-se uma média aritmética dessas precisões médias. Os valores possíveis de MAP encontram-se no intervalo de 0 a 1. A fórmula abaixo descreve como calcular um valor de MAP [2, 32]:

$$MAP = \frac{\sum_{q=1}^{|Q|} Pm(q)}{|Q|} \tag{2-17}$$

Na fórmula acima $|Q|$ é a cardinalidade do conjunto de consultas realizadas para se calcular o valor de MAP. Essa quantidade é definida pela pessoa que deseja calcular um valor de MAP para um sistema de RI. Valores muito pequenos podem não representar um valor correto do sistema.

NDCG

A medida *Normalized Discounted Cumulative Gain* (NDCG) mede a eficácia de um sistema de RI baseado no grau de relevância dos documentos posicionados no início da lista de ordenação. A medida varia entre 0 e 1, com 1 representando a ordenação ideal dos documentos, ou seja, documentos mais relevantes ocupando as posições superiores da lista.

Em máquinas de busca para a Web a maioria dos usuários analisa apenas as primeiras páginas da ordenação. Logo, é importante nesse contexto a utilização de uma medida que privilegie métodos de RI capazes de ordenar os documentos mais relevantes no início da lista de ordenação e penalize aqueles que, mesmo trazendo documentos relevantes, não sejam capazes de colocá-los mais próximo do início da lista de ordenação. Por esse motivo a métrica NDCG é comumente utilizada em máquinas de busca web.

A relevância de um documento pode ser binária ou discreta dependendo do modelo utilizado pelo sistema. Em uma relevância binária, assume-se apenas dois graus de relevância, por exemplo: 0, 1. Nela um documento é classificado apenas como relevante (1) ou irrelevante (0) para uma dada consulta. Em uma relevância discreta, assume-se mais de um grau de relevância, por exemplo: $0,1,2,3,4$.

A medida de avaliação NDCG foi criada para situações em que os sistemas de RI possuem medidas não binárias de relevância. Com sistemas nessas condições é necessário que a maioria dos documentos com maior relevância sejam retornados no topo da lista. A medida de avaliação MAP não consegue prever esse tipo de problema, pois considera apenas a relevância de um documento de forma binária.

O ganho acumulado (CG) é calculado pela equação 2-18:

$$CG(n) = \sum_{i=1}^{n} rel_i \tag{2-18}$$

Onde rel_i é a relevância do documento na posição i da lista e n é o tamanho da lista de documentos retornados.

Por exemplo, considere a lista $L = \{3,2,3,0,0,1,2,2,3,0\}$ de relevância dos documentos retornados. Assim para essa lista de relevância teremos, a seguinte lista de

ganho cumulativo: $CG_L = \{3,5,8,8,8,9,11,13,16,16\}$.

A medida DCG (*Discounted Cumulative Gain*) é fundada em duas regras. Documentos com relevância máxima são mais importantes que documentos de relevância grande. Quanto mais longe da primeira posição da lista um documento estiver, menos valioso o documento será para o usuário. Tendo em vista essas duas regras, a fórmula do DCG é como a seguir:

$$DCG(n) = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2(i)} \tag{2-19}$$

Observando a fórmula podemos perceber que quanto maior o valor de n, maior será o desconto dado à relevância do documento e menor será o valor acumulado. Para a lista "L" acima o DCG será: $DCG_L = \{3,5,6.89,6.89,6.89,7.28,7.99,8.66,9.61,9.61\}$, percebe-se que os valores no DCG são iguais ou menores que os valores de ganho acumulado devido ao desconto dado cada vez que o documento está mais longe do início da lista.

As listas de documentos retornados podem variar de comprimento dependendo de cada consulta. Assim, a comparação do desempenho de um sistema de RI utilizando várias consultas (assim como realizado no MAP) não pode ser consistentemente feita utilizando apenas DCG. Para tanto é utilizada a medida de avaliação NDCG. Nessa medida, o DCG de cada consulta é igualado em um tamanho n. Calcula-se também um DCG Ideal (IDCG) até essa mesma posição e realiza-se a razão entre o DCG e o IDCG. O IDCG seria a lista ideal de relevâncias dos documentos retornados. Para encontrar o IDCG faz-se de forma gulosa. Colocam-se nas primeiras posições os documentos de relevância máxima, se ainda houver espaço na lista, colocam-se os documentos de relevância máxima menos 1 e assim por diante até que se preencha todas as p posições possíveis da lista.

Sendo assim a fórmula para se calcular o NDCG é:

$$NDCG_n = \frac{DCG_n}{IDCG_n} \tag{2-20}$$

Cada uma das medidas de avaliação vistas nessa seção são utilizadas para diferentes contextos. A precisão e revocação foram criados para situações em que havia a necessidade de resultados completos, ou seja, retornar a maioria dos documentos relevantes e que se retornasse a menor quantidade de documentos não relevantes possível. A precisão em um ponto, a média das precisões em um ponto e o MAP são medidas que vieram a fim de avaliar sistemas de RI cuja ordenação dos documentos retornados fosse importante. Porém outros sistemas de RI, como por exemplo, máquinas de busca web não possuem tal necessidades. Máquinas de busca web necessitam apenas que os melhores documentos sejam retornados no topo da lista. Nesses tipos de sistemas de RI a medida

mais adequada é a NDCG.

2.2 Aprendizado de máquina

Aprendizado de máquina pode ser definido como o emprego de métodos computacionais que usam a experiência adquirida para melhorar sua performance ou para fazer predições com maior acurácia. Experiência, nesse cenário, significa a informação conhecida pelo computador. Normalmente, ela está de forma eletrônica e de modo com que o computador possa reconhecer informações para fazer análises. Essa informação poderia ser, por exemplo, um conjunto de documentos digitalizados e com palavras-chaves escolhidas por humanos [35].

Os sistemas que utilizam aprendizado de máquina são normalmente utilizados para resolver tarefas que:

- estão além da capacidade humana, como por exemplo analisar uma grande quantidade de dados ou encontrar fenômenos interessantes em grandes quantidades de dados.
- não possuem algoritmo preciso para sua solução, como por exemplo um sistema que simula a forma como as formigas trabalham para encontrar comida.

Assim como na classificação manual, o sucesso de um sistema que utiliza aprendizado de máquina depende do tamanho e qualidade das experiências adquiridas. Se os dados utilizados para aprender forem muito pequenos ou possuírem erros nas classificações, o sistema que utiliza aprendizado de máquina pode não possuir uma boa acurácia.

Algoritmos de aprendizado de máquina possuem grande sucesso em uma vasta quantidade de aplicações. Essas aplicações incluem:

- máquinas de busca;
- classificação de documentos;
- sistemas de recomendação;
- reconhecimento ótico de caracteres;
- visão computacional;
- biologia computacional;
- química computacional;
- inteligência artificial de jogos;
- reconhecimento de fala;
- processamento de linguagem natural;
- detecção de invasão ou fraude;

- diagnósticos médicos; e
- pilotos automáticos.

Essa lista de aplicações não abrange todos os exemplos de usos práticos de aprendizado de máquina. As aplicações que se utilizam de aprendizado de máquina podem estar dentro das seguintes classes de problemas: Classificação, Regressão, *Clustering*, *Ranking* e Redução de dimensionalidade. Nas seções a seguir são introduzidos os conceitos de cada uma dessas classes de problemas.

2.2.1 Classificação

Em aprendizagem de máquina, o problema de classificação consiste em automaticamente identificar a qual categoria uma informação pertence, dentre um conjunto de categorias. Por exemplo, uma classificação de documentos em um sistema poderia tratar cada tema dos documentos como uma categoria do problema de classificação, haveria categorias como política, esportes, clima, tecnologia ou economia. Uma classificação de imagens poderia tratar o conteúdo da imagem como uma categoria, sendo assim haveriam categorias como retrato, paisagem, animal ou rosto [35].

As categorias podem ser do tipo categórico (esporte, economia, política, clima e outros), ordinal (pequeno, médio, grande, imenso), valores inteiros (número filhos de uma família), ou valores reais (dimensões de uma caixa, salário de funcionários).

Alguns algoritmos que resolvem problemas de classificação trabalham apenas com dados discretizados e necessitam que os valores inteiros e reais sejam convertidos em grupos. Por exemplo, o valor real salário de funcionário poderia ser convertido nos grupos:

- salário até R$999,99,
- salário de R$1.000,00 até R$2.499,99,
- salário de R$2.500,00 até R$4.999,99,
- salário de R$5.000,00 até R$7.999,99, e
- salário acima de R$8.000,00.

Outro exemplo seria com as dimensões de uma caixa, ao invés de ser considerado o volume em centímetros cúbicos, apenas categorizá-las em “pequena”, “média”, “grande” e “imensa”. Ao ato de converter ou particionar valores contínuos em grupos dá-se o nome de discretização.

2.2.2 Regressão

Diferentemente dos problemas de classificação que visam prever uma categoria, os problemas de regressão visam prever um valor real. Exemplos de aplicações que

resolvem problemas com regressão incluem a previsão do índice de bolsa de valores, previsão do volume da precipitação em uma determinada região e a idade de um fóssil de acordo com a quantidade de carbono-14 presente [35].

A definição de regressão é: tarefa de aprender uma função alvo f que mapeie cada conjunto de atributos x em uma saída de valores contínuos y.

O objetivo em ambos os problemas de regressão e classificação é aprender um modelo que minimize os erros entre o valor previsto e o real, porém nos problemas de regressão, o tamanho do erro de uma predição incorreta depende da diferença entre o valor desejado e a predição realizada.

2.2.3 *Clustering*

Clustering é o agrupamento de dados em conjunto de elementos similares. É geralmente realizado na análise de um grande conjunto de dados. Apesar de ser muito semelhante ao problema de classificação, o problema de *clustering* visa detectar grupos com indivíduos semelhantes, diferentemente dos problemas de classificação que visam prever a qual grupo um determinado item ou indivíduo pertence [35].

Um exemplo de aplicação que utiliza algoritmos de *clustering* é: sobre dados de uma população sob tratamento de uma doença causada por um novo vírus pode-se detectar grupos de indivíduos que possuam sintomas similares.

2.2.4 *Ranking*

Os problemas de *ranking* possuem como objetivo central ordenar itens de acordo com algum critério. Máquinas de busca web utilizam-se de algoritmos de *ranking* e possuem a finalidade de retornar uma lista de páginas relevantes ordenadas de forma decrescente de acordo com sua relevância para uma dada consulta [35].

2.3 *Learning to Rank*

A fim de resolver o problema da recuperação de documentos em RI, muitos modelos de ordenação têm sido propostos e utilizados. Porém, devido à quantidade crescente de documentos disponíveis, tornou-se necessário o desenvolvimento de tecnologias com aprendizado de máquina na construção de modelos mais eficazes. A esses métodos que aprendem como combinar características predefinidas para realizar a ordenação de documentos damos o nome de métodos *learning to rank* (L2R).

Utilizando-se do exemplo de ordenação decrescente em ordem de relevância de documentos, alguns conceitos são descritos abaixo.

- Item: Exemplos ou instâncias de dados utilizados para a aprendizagem ou predição. No problema de exemplo escolhido, esses itens correspondem a uma 3-upla da forma: $<q,d,r>$, onde "q" representa uma consulta, "d" representa um documento e "r" o grau de relevância de "d" dada a consulta "q".
- Características (*Features*): O conjunto de atributos, normalmente representado como um vetor, associados a um item. No caso da ordenação de documentos *features* são as características que descrevem um documento "d". Geralmente essas características correspondem a valores que relacionam a consulta com o documento. Por exemplo, a soma das frequências ($tf_{t,d}, tf_{t,q}$) dos termos que aparecem na consulta q e no documento d, ou o valor do BM25 (ver equação 2-12) entre q e d.
- Relevância: Valor dado ao item previsto. A relevância de um documento pode variar a sua estrutura dependendo do modelo utilizado. Ela pode ser binária, por exemplo: $0,1$, ou seja, um documento é classificado apenas como relevante (1) ou irrelevante (0) para uma dada consulta ou a relevância pode ser discreta, por exemplo: $0,1,2,3,4$, ou seja, um documento é classificado por níveis de relevância para uma dada consulta.
- Documento: Porção de texto a que se deseja classificar ou ordenar. É composto por um conjunto de características.
- Consulta: Representação de uma necessidade de informação. Pode ser composta por uma lista de termos ou por apenas um identificador da consulta.

As abordagens L2R fazem uso de um conjunto de treino como entrada para o processo de aprendizagem de máquina. Como saída desse processo, é gerada uma função de ordenamento ou também chamada de modelo. Durante a realização de uma consulta, o sistema de RI utiliza-se dessa função de ordenamento para ordenar os documentos, realizando uma predição [29]. Esse esquema do funcionamento das abordagens L2R pode ser visualizado na figura 2.7.

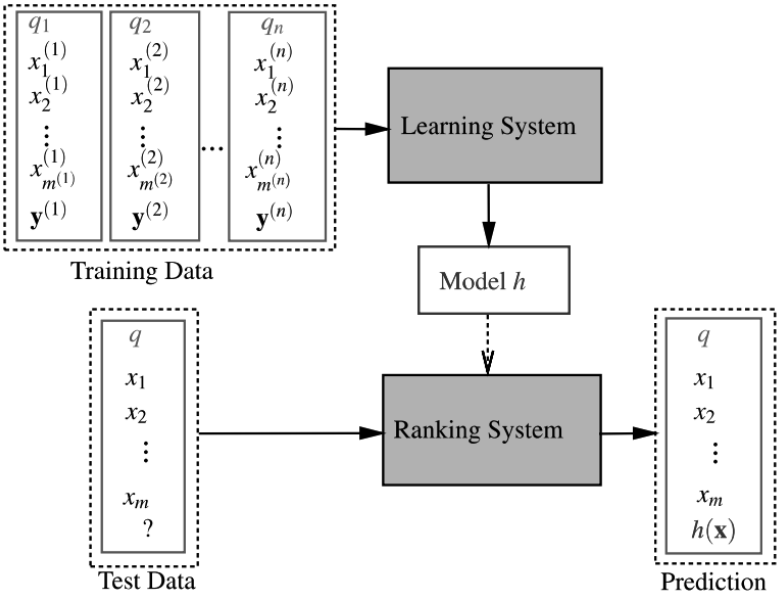


Figura 2.7: *Esquema de funcionamento do framework L2R*

Nas seções seguintes será descrito os conceitos dos documentos de treino, teste e validação envolvidos no processo de *learning to rank*.

2.3.1 Conjunto de treino

Como entrada para o algoritmo de L2R tem-se o conjunto de treino. Este conjunto geralmente consiste em consultas e documentos. Cada consulta está associada com um conjunto de documentos. Na prática o conjunto de treino é um arquivo texto em que cada linha representa um trio $<q,d,r>$. Onde q representa uma consulta, o dado d representa um documento e r a relevância do documento d para a consulta q.

A definição de um documento de treino abaixo pode ser encontrada em [22].

Definição 2.4 *Suponha que Q é um conjunto de consultas e D é o conjunto de documentos. Suponha que $\mathcal{Y} = \{1,2,...,|\mathcal{Y}|\}$ é o conjunto de rótulos que representam graus de relevância. Suponha ainda que $\{q_1,q_2,q_3,...,q_{|Q|}\} \subset Q$ é o conjunto de consultas para o treino e q_i é a i-ésima consulta. $D_i = \{d_{i,1},d_{i,2},...,d_{i,n_i}\}$ é o conjunto de documentos associados a consulta q_i e $\boldsymbol{Y}_i = \{y_{i,1},y_{i,2},...,y_{i,n_i}\}$ é o conjunto de rótulos relacionados a consulta q_i, onde n_i representa o tamanho de D_i; $d_{i,j}$ representa o j-ésimo documento em D_i; e $Y_{i,j} \in Y$ representa o j-ésimo grau de relevância em $\boldsymbol{Y}_i$, e representa o grau de relevância do documento $d_{i,j}$ a respeito da consulta q_i. O documento de treino original pode ser representado como $S = \{(q_i,D_i),\boldsymbol{Y}_i\}_{i=1}^{m}$.*

Em documentos de treino típicos, a representação de um documento é geralmente realizada por um vetor de características $X = \{x_1,x_2,...,x_{|X|}\}$ em que $|X|$ é o número de características contidas no documento de treino.

2.3.2 Conjunto de validação

Assim como o conjunto de treino, o conjunto de validação também é uma tripla que consiste em consultas, documentos e relevâncias. Ambos possuem a mesma definição, porém usos diferentes.

Muitos algoritmos de *learning to rank* possuem parâmetros internos que são determinados utilizando-se do documento de treino. Porém, para obter uma melhor performance na tarefa de *ranking*, alguns algoritmos utilizam-se do documento de validação (também chamado de conjunto de validação) para ajustar seus parâmetros internos.

Na prática o conjunto de validação, assim como o conjunto de treino, é um arquivo texto em que cada linha representa um trio $<q,d,r>$. O dado q representa uma consulta, o dado d representa um documento e r a relevância do documento d para a consulta q.

2.3.3 Conjunto de teste

O conjunto de treino é utilizado a fim de construir um modelo que relaciona as características de um documento a sua relevância para uma dada consulta e o conjunto de validação é utilizado para ajustar parâmetros aprendidos durante a fase de treino. Porém, além desses documentos, a abordagem L2R utiliza-se também de um conjunto de teste.

Assim como os conjuntos de treino e validação, os dados de teste consistem em uma tripla contendo consultas q, documentos associados as consultas D_i e relevâncias associadas Y_i. O conjunto de teste pode ser definido como: $Te = \{(q_i, D_i), \mathbf{Y}_i\}_{i=1}^{m}$.

Da mesma forma que no conjunto de treino, os documentos no conjunto de teste são representados por um vetor de características.

Após aprendido o modelo para ordenação de documentos, o conjunto de teste é utilizado para predizer escores aos documentos D_i presentes no conjunto de teste. Após a predição, os documentos são ordenados em forma decrescente de acordo com o valor do escore, resultando assim em um *ranking* de documentos. Apesar de existir, no documento de teste, os dados de relevância, os mesmos não são utilizados na fase de teste para a predição da relevância de um documento a uma dada consulta. Esse dado é apenas utilizado com a finalidade de realizar as medidas de avaliação de um sistema de RI, como MAP, NDCG e outras.

2.3.4 Abordagens L2R

Muitos algoritmos em L2R encaixam-se no *framework* ilustrado na figura 2.7. Para melhor entendê-los, Tie-Yan Liu [29] realiza uma categorização desses algoritmos em três abordagens: Abordagem *Pointwise*, Abordagem *Pairwise* e Abordagem *Listwise*.

As próximas subseções tratam dos aspectos fundamentais de cada uma dessas abordagens em *Learning to rank*.

Abordagem *Pointwise*

Nas abordagens *pontwise*, presume-se que cada par consulta-documento no conjunto de treino tem uma pontuação numérica ou ordinal. Assim, problemas *learning to rank* podem ser aproximados a um problema de regressão, ou seja, dado um único par consulta-documento, prever o seu escore. Existem vários algoritmos de aprendizado de máquina supervisionados que podem ser utilizados para abordagens *pointwise*. Métodos existentes de classificação, regressão ou classificação ordinal podem ser utilizados na abordagem *pointwise* [22].

Alguns exemplos de algoritmos da abordagem *pointwise* são: *Subset Ranking* [10], *McRank* [27], *PRanking* [11] e *OC SVM* [44].

O espaço de entrada dos algoritmos da abordagem *pointwise* contém um vetor de características para um único documento. O espaço de saída dos algoritmos dessa abordagem contém o grau de relevância de cada documento. Portanto, nesses algoritmos é necessária uma nova etapa para geração do *ranking*, que consiste em ordenar os documentos de acordo com o escore atribuído pela função aprendida.

O espaço de hipótese contém funções que recebem o vetor de características de um documento como entrada e preveem o grau de relevância do documento. Tais funções são normalmente chamadas funções de escore. É baseada na função de escore da hipótese que pode-se ordenar todos os documentos e produzir uma lista ordenada pelo grau de relevância. Para se encontrar a melhor função de escore é normalmente utilizada uma função de perda.

A função de perda tem o objetivo de medir o grau em que a predição gerada pela hipótese está em acordo com a relevância real do documento a fim de otimizar a função de escore para que a mesma realize melhores predições. Em diferentes algoritmos de classificação *pointwise*, o *ranking* é modelado como regressão, classificação ou regressão ordinal. Assim, as correspondentes perda de regressão, perda de classificação, e perda de regressão ordinal são usadas como as funções de perda nas abordagens *pointwise* [29].

A ideia geral por trás da abordagem *pointwise* é que, dado um vetor de características de cada documento do treino como entrada, o grau de relevância de cada um desses documentos pode ser previsto por uma função de escore, e todos esses escores podem ser utilizados para ordenar os documentos e produzir a lista final de classificação.

Considerando que a maioria das medidas de avaliação para recuperação de informação são baseadas em consulta e consideram a posição do documento no ranking, a abordagem *pointwise* possui suas limitações [29].

Percebe-se que na abordagem *pointwise* não é considerado a inter-dependência entre os documentos, e assim, a posição de um documento na lista de classificação não é visível para a função de perda. A função de perda na abordagem *pointwise* pode super enfatizar documentos não relevantes.

Além disso, os algoritmos dessa abordagem não fazem uso do fato de que alguns documentos são, na verdade, relacionados com a mesma consulta. Dessa forma, quando o número de documentos associados às consultas variam muito, a função de perda será dominada por aquelas consultas com grande quantidade de documentos.

A vantagem da utilização da abordagem *pointwise* é que essa abordagem suporta diretamente a aplicação de teorias e algoritmos existentes em regressão ou classificação.

Abordagem *Pairwise*

A abordagem *pairwise* não foca em predizer o grau de relevância de cada documento, mas na ordem relativa entre dois documentos. Nos algoritmos da abordagem *pairwise*, os problemas *learning to rank* são tratados como um problema binário de classificação ou regressão. Deve-se aprender um classificador binário que pode dizer qual documento é mais relevante em um determinado par de documentos a fim de ordenar os documentos de acordo com o grau de relevância binário. O objetivo principal dessa abordagem é minimizar o número médio de inversões no ranking.

Alguns exemplos de algoritmos da abordagem *pairwise* são: Ranking SVM [23], RankBoost [17], RankNet [3], GBRank [52], IR SVM [7], Lambda Rank [4], LambdaMART [50].

O espaço de entrada dos algoritmos da abordagem *pairwise* contém pares de documentos, ambos representados por vetores de características. O espaço de saída dos algoritmos dessa abordagem contém a preferência emparelhada (com valores de $\{-1,1\}$) entre cada par de documentos. Na abordagem *pairwise*, quando um par de documento é comparado, aquele que for mais relevante à consulta terá uma referência emparelhada 1 enquanto o outro terá uma referência de -1.

O espaço de hipótese contém funções bi-variadas h que recebem um par de documentos como entrada e retornam a ordem relativa entre eles.

A função de perda na abordagem *pairwise* mede a inconsistência entre a função da hipótese h e o grau de relevância y. Em muitos algoritmos *pairwise*, a ordenação é modelada como uma classificação emparelhada, e a correspondente perda de classificação em um par de documentos é usada como a função de perda. Quando utilizada uma função de escore, a perda de classificação é expressa pela diferença entre o escore do par de documentos [29].

A ideia geral por trás da abordagem *pairwise* é que, dado um par de documentos representados por vetores de características do conjunto de treino como entrada, o grau de

relevância entre cada um desses pares de documentos pode ser predito por uma função de escore que tem como objetivo principal minimizar a média de pares de documentos não corretamente classificados. A função de perda utilizada na abordagem *pairwise* leva em consideração a ordem de relevância relativa entre cada par de documentos.

Pode-se perceber que o problema que a abordagem *pairwise* visa não é o mesmo que na abordagem *pointwise*. Em *pointwise* o processo de aprendizagem opera em cada documento enquanto nas abordagens *pairwise* o processo de aprendizagem opera em pares de documentos do conjunto de treino.

O algoritmo *pairwise Rankboost* adota o *AdaBoost* para a classificação sobre pares de documentos. A diferença principal entre *RankBoost* e *AdaBoost* é que a distribuição em *RankBoost* é definida nos pares de documentos enquanto em *AdaBoost* é definido em documentos individuais.

A ideia básica do algoritmo *RankBoost* é moldar o processo de *learning to rank* como um problema de classificação binária em pares de documentos e então adotar a abordagem de *boosting*. Assim como todos os algoritmos de *boosting*, o *RankBoost* treina um *ranker* fraco a cada rodada de iteração e combina-os para formar a função de *ranking* final. Após cada rodada os documentos são rebalanceados de forma que os pesos dos pares de documentos que forem corretamente ordenados são diminuídos e os pesos dos pares de documentos serão aumentados caso contrário [17].

Abordagem *Listwise*

Diferentemente da abordagem *pointwise* que retorna um escore para cada documento e da abordagem *pairwise* que retorna um grau de relevância entre dois documentos, a abordagem *listwise* recebe o conjunto completo de documentos associados com uma consulta do documento de treino como espaço de entrada e prediz as relevâncias dos documentos para aquela consulta.

Os algoritmos da abordagem *listwise* podem ser divididos em duas categorias. Na primeira categoria estão aqueles que tentam, em sua função de perda, otimizar diretamente o valor de uma medida de avaliação, em média de todas as consultas de um conjunto de treino. Esse processo não é simples pois a maioria das medidas de avaliação não são funções contínuas e então aproximações contínuas são utilizadas. Devido à grande relação entre a função de perda e as medidas de avaliação, esses algoritmos dessa categoria são referidos como métodos de otimização direta. Nos algoritmos da segunda categoria, a função de perda não está relacionada diretamente com as medidas avaliação de sistemas de recuperação da informação.

A abordagem *listwise* busca minimizar uma função de perda que leva em consideração a ordem relativa de todos os itens da lista. As penalidades por errar itens em uma posição mais elevada são maiores do que quando o erro em uma posição menor.

Alguns exemplos de algoritmos da abordagem *listwise* são: ListNet [8], RankCosine [39], Relational Ranking [38], LambdaRank [5], LambdaMART [49].

A entrada dos algoritmos da abordagem *listwise* contém um conjunto de documentos associados a uma consulta q. A saída dos algoritmos dessa abordagem contém a lista de ordenação de documentos.

A hipótese contém funções h multivariadas que operam sobre um conjunto de documentos e prevêem sua permutação. Por razões práticas, a hipótese h é normalmente implementada com uma função de escore f. Ou seja, primeiro uma função de escore f é utilizada para dar uma pontuação para cada documento, e, em seguida, estes documentos são classificados em ordem decrescente dos escores para produzir a permutação desejada.

Há dois tipos de função de perda comumente utilizadas na abordagem *listwise*. Para o primeiro tipo, a função de perda é explicitamente relacionada com a medida de avaliação. No segundo tipo a função de perda não está relacionada com uma medida de avaliação. Sendo assim, nesse tipo de abordagem, a função de perda pode ser explicitamente relacionada com as medidas de avaliação ou não, o que torna difícil a classificação de uma função de perda em uma função de perda *listwise*. Em [29], uma função de perda é *listwise* de acordo com os seguintes critérios:

- Se for definida levando em consideração todos os documentos do treino associados a uma consulta.
- Se não puder ser totalmente decomposta em uma simples somatória sobre documentos individuais ou em pares.
- Se enfatiza o conceito de uma lista ordenada de documentos e a posição final dos documentos é visível para a função de perda.

A ideia principal dos algoritmos da abordagem *listwise* é que, dado um vetor de características que representa um documento do conjunto de treino como entrada, o grau de relevância de cada um desses documentos será predito com uma função de escore que busca otimizar diretamente o valor de uma medida de avaliação (precisão, revocação, MAP, outros), em média para todas as consultas do conjunto de treino. A função de perda usada nessa abordagem leva em consideração a posição dos documentos na lista ordenada para todos documentos de uma mesma consulta [28].

A abordagem *listwise* tem uma certa vantagem sobre os algoritmos das outras abordagens, pois com essa abordagem é possível distinguir no processo de aprendizagem a qual consulta um documento pertence e sua ordem na lista ordenada de documentos, o que não ocorre com as outras abordagens.

2.4 Coleções de referência

Nessa seção será introduzido sobre a coletânea de referência LETOR 3.0 [37] que foi a utilizada neste trabalho de dissertação. Será explicitado sobre os tipos de coleções presentes na coletânea LETOR, como são selecionados os conjuntos de documentos e consultas e a descrição das características (*features*) presentes nas coleções.

Para facilitar o processo de avaliação de um experimento com *learning to rank* é necessária uma coleção com documentos indexados, consultas selecionadas para conjunto de treino, teste e validação, características extraídas dos documentos entre outros fatores. Entretanto, antes não existiam tais coleções e os pesquisadores utilizavam suas próprias coleções. Mas isso não ajudava a comparar um experimento a outro, pois as coleções utilizadas não eram as mesmas. Com essa motivação criou-se a coletânea LETOR.

Na coletânea LETOR existem duas coleções de documentos: A coleção "Gov" e a coleção "OHSUMED".

2.4.1 Coleção Gov

Na "Text REtrieval Conference" de 2003 e de 2004 (TREC 2003 / TREC 2004) havia uma parte para recuperação da informação *web* chamada "*web track*". Nessa parte da conferência foram utilizados dados minerados por *crawlers* do domínio "gov" do ano de 2002 (domínio de uso restrito do governo norte americano). Existe aproximadamente um milhão de páginas html nessa coleção de documentos [28, 37].

Para tal coleção, foram separados três problemas da recuperação de documentos *web*: *Topic Destilation* (TD), *HomePage finding* (HP) e *Named Page finding* (NP).

A tarefa de *Topic Destilation* consiste em encontrar uma lista de sites relevantes a primariamente um certo tópico. O foco desta tarefa está em encontrar uma boa página de entrada (aquela ao qual um usuário utilizou para chegar a página com conteúdo relevante) ao invés daquela que contem o conteúdo relevante propriamente, pois as páginas de entrada contêm melhor informação geral do que está contido no site. Assim, a tarefa de *Topic Destilation* consiste em encontrar um documento (*website*) que descreva algo pedido na consulta.

A tarefa *HomePage finding* visa encontrar a página inicial para uma dada consulta. Assim, a tarefa de *HomePage finding* consiste em encontrar um documento (*website*) que possua a URL exata que foi pedida na consulta.

A tarefa de *Name Page finding* consiste em encontrar a página cujo nome é idêntico ao da consulta.

Uma característica interessante sobre essas tarefas é que normalmente haverá apenas um documento relevante nas tarefas de *HomePage finding* e *Name Page finding* enquanto para a tarefa de *Topic Destilation* pode haver vários documentos relevantes.

Para a coleção Gov da TREC 2003 existem 50 consultas para a tarefa de *Topic Destilation*, 150 consultas para a tarefa de *HomePage finding* e mais 150 consultas para a tarefa *Name Page finding*. Para a coleção Gov da TREC 2004 existem 75 consultas para cada uma das tarefas. Para melhor visualização da quantidade de consultas veja a tabela 2.1 abaixo [37].

Tarefa	TREC 2003	TREC 2004
Topic Destilation	50	75
HomePage finding	150	75
Name Page finding	150	75

Tabela 2.1: *Número de consultas Web track da conferência TREC*

A coletânea LETOR utiliza as seguintes abreviações na coleção gov: se estiver referindo à tarefa *Topic Destilation* então utiliza TD, se referindo à tarefa *HomePage finding* então utiliza HP, enquanto para *Name Page finding*, NP. Como, na coleção gov, existem duas TRECs possíveis (2003 e 2004), então quando se estiver referenciando a tarefa de *Topic Destilation* da TREC 2003 será abreviado da forma TD2003. Se ao invés da TREC 2003, fosse a do ano de 2004, então abreviaria-se na forma TD2004. De forma análoga pode-se abreviar as demais tarefas.

2.4.2 Coletânea OHSUMED

A coleção OHSUMED é um subconjunto do conjunto de dados MEDLINE [2]. A MEDLINE é uma base de dados com publicações na área de saúde e atualmente conta com mais de 21,6 milhões de documentos selecionados de 5639 journals da área de saúde desde 1950 até atualmente [2].

A OHSUMED consiste em aproximadamente 0,3 milhões de documentos de 270 artigos médicos do período de 1987 a 1991. A coleção conta também com um conjunto de 106 consultas que descrevem uma necessidade de informação médica. O grau de relevância dos documentos a respeito das consultas foram classificados por humanos em três níveis: relevante, parcialmente relevante e irrelevante. Há, nessa coleção, um total de 16.140 pares de documento-consulta com as respectivas relevâncias [28, 37].

2.4.3 Seleção dos documentos

Devido ao tamanho das coleções, torna-se impraticável o julgamento das relevâncias de todos os documentos para uma dada consulta. Da mesma forma que nas prá-

[2]http://www.nlm.nih.gov/bsd/pmresources.html
[2]Dados extraídos do site http://www.nlm.nih.gov/bsd/num_titles.html em 27 de maio de 2014

ticas comuns em recuperação da informação, apenas os documentos considerados mais relevantes são selecionados para julgamento. De forma similar não faz-se necessário extrair o vetor de características para todos os documentos da coleção.

Na coleção Gov algumas pessoas selecionadas pelo comitê da TREC classificaram, para cada consulta, apenas alguns documentos como relevantes. Todos os documentos restantes foram automaticamente classificados como irrelevantes. Seguindo essa prática da TREC, a LETOR também considera todos os documentos não classificados como irrelevantes. Na coleção LETOR, para selecionar apenas os documentos mais relevantes para realizar o julgamento, foi primeiramente utilizado o modelo BM25 para ordenar todos os documentos de cada consulta, e então, foram selecionados os 1000 documentos mais relevantes para cada consulta para a extração do vetor de características [37].

Diferentemente da coleção Gov em que os documentos são apenas rotulados como relevantes e todos os demais são irrelevantes, na OHSUMED, existe explicitamente o rótulo irrelevante e todos os documentos não rotulados são ignorados na avaliação. Seguindo a mesma prática, a LETOR realiza a extração do vetor de características apenas para os documentos rotulados, ignorando os demais [37]. Como resultado, para cada consulta da coleção OHSUMED, tem-se aproximadamente 152 documentos associados para extração do vetor de características.

2.4.4 Seleção das características

A seleção das características na coletânea de referência LETOR foi realizada seguindo os seguintes princípios:

- Cobrir o maior número de características clássicas de RI possível.
- Reproduzir o maior número de características propostas nos últimos trabalhos do SIGIR (Special Interest Group on Information Retrieval).
- Estar de acordo com as definições dos documentos ou artigos originais.

Um exemplo de arquivo de dados da LETOR pode ser visto na figura 2.8 abaixo. Cada linha representa um par documento-consulta. A primeira coluna do arquivo representa o grau de relevância do par documento-consulta, ou seja, a relevância do documento em questão àquela consulta. A segunda coluna representa a consulta dada por um identificador (qid). As colunas consequentes até a penúltima representam o vetor de características do documento em questão. Cada uma dessas colunas do vetor de características representa uma característica do documento. Na última coluna há o identificador do documento que normalmente não é utilizado nos sistemas de RI.

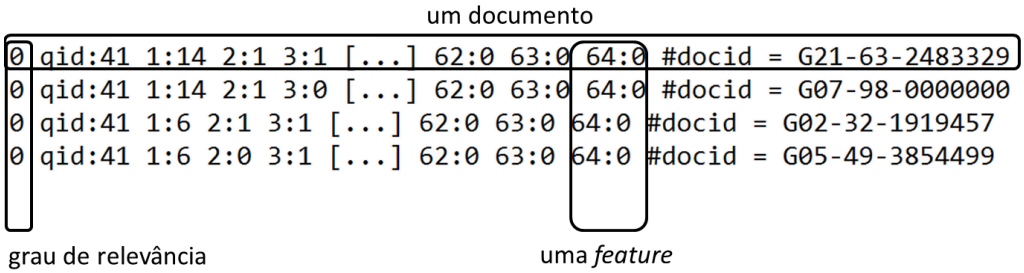


Figura 2.8: *Exemplo de dados da LETOR*

2.4.5 Seleção das características em Gov

Para a coleção Gov, foram extraídas no total 64 características para cada par documento-consulta. A tabela abaixo 2.2 mostra todas as características para um documento em Gov.

A identificação de cada característica na tabela é a mesma nos conjuntos de dados da LETOR. Além disso, na tabela há a diferenciação das categorias das características entre as que são dependentes da consulta, ou *query-dependent* (Q-D), aquelas que são dependentes apenas dos documentos (D) e as que são dependentes apenas da consulta (Q).

ID	Descrição da característica	Categoria
1	tf - body	Q-D
2	tf - anchor	Q-D
3	tf - title	Q-D
4	tf - URL	Q-D
5	tf - todo documento	Q-D
6	idf - body	Q
7	idf - anchor	Q
8	idf - title	Q
9	idf - URL	Q
10	idf - todo documento	Q
11	tf-idf - body	Q-D
12	tf-idf - anchor	Q-D
13	tf-idf - title	Q-D
14	tf-idf - URL	Q-D
15	tf-idf - todo documento	Q-D
16	Tamanho - body	D
17	Tamanho - anchor	D
18	Tamanho - title	D

19	Tamanho - URL	D
20	Tamanho - todo documento	D
21	BM25 - body	Q-D
22	BM25 - anchor	Q-D
23	BM25 - title	Q-D
24	BM25 - URL	Q-D
25	BM25 - todo documento	Q-D
26	LMIR.ABS - body	Q-D
27	LMIR.ABS - anchor	Q-D
28	LMIR.ABS - title	Q-D
29	LMIR.ABS - URL	Q-D
30	LMIR.ABS - todo documento	Q-D
31	LMIR.DIR - body	Q-D
32	LMIR.DIR - anchor	Q-D
33	LMIR.DIR - title	Q-D
34	LMIR.DIR - URL	Q-D
35	LMIR.DIR - todo documento	Q-D
36	LMIR.JM - body	Q-D
37	LMIR.JM - anchor	Q-D
38	LMIR.JM - title	Q-D
39	LMIR.JM - URL	Q-D
40	LMIR.JM - todo documento	Q-D
41	Sitemap based term propagation	Q-D
42	Sitemap based score propagation	Q-D
43	Hiperlink based score propagation: weighted in-link	Q-D
44	Hiperlink based score propagation: weighted out-link	Q-D
45	Hiperlink based score propagation: uniform out-link	Q-D
46	Hiperlink based propagation: weighted in-link	Q-D
47	Hiperlink based feature propagation: weighted out-link	Q-D
48	Hiperlink based feature propagation: uniform out-link	Q-D
49	HITS authority	Q-D
50	HITS hub	Q-D
51	PageRank	D
52	HostRank	D
53	Topical PageRank	Q-D
54	Topical HITS authority	Q-D

55	Topical HITS hub	Q-D
56	Inlink number	D
57	Outlink number	D
58	Número de barras na URL	D
59	Comprimento da URL	D
60	Número de páginas filhas	D
61	BM25 do título extraído da página HTML	Q-D
62	LMIR.ABS do título extraído da página HTML	Q-D
63	LMIR.DIR do título extraído da página HTML	Q-D
64	título extraído da página HTML	Q-D

Tabela 2.2: *Características da coleção Gov*

2.4.6 Seleção das características em OHSUMED

Para a coleção OHSUMED foram selecionadas um total de 45 características extraídas das partes: título, *abstract* e "título + *abstract*". Assim como na seleção das características para a Gov, os números de cada característica na tabela corresponde a identificação da característica na base de dados da LETOR. Nessa coletânea também são utilizadas as abreviaturas Q-D, D e Q para representar respectivamente características dependentes da consulta e do documento, dependentes apenas do documento e dependentes apenas da consulta.

A tabela 2.3 abaixo mostra a seleção das características em OHSUMED. Considere $|d|$ o tamanho do porção de texto (título, *abstract* ou "título + *abstract*") em questão e $|C|$ o total de documentos na coleção.

ID	Descrição da característica	Categoria
1	tf - título	Q-D
2	$\log(tf(d)+1)$ - título	Q-D
3	$\frac{tf(d)}{\lvert d\rvert}$ - título	Q-D
4	$\log(\frac{tf(d)}{\lvert d\rvert}+1)$ - título	Q-D
5	$\log(\frac{\lvert C\rvert}{df})$ - título	Q
6	$\log(\log(\frac{\lvert C\rvert}{df}))$ - título	Q
7	$\log(\frac{\lvert C\rvert}{tf(C)}+1)$ - título	Q
8	$\log(\frac{tf(d)}{\lvert d\rvert}\cdot\log(\frac{\lvert C\rvert}{df})+1)$ - título	Q-D
9	$tf(d)\cdot\log(\frac{\lvert C\rvert}{df})$ - título	Q-D

10	$\log(\frac{tf(d)}{\lvert d\rvert} \cdot \frac{\lvert C\rvert}{tf(C)} + 1)$ - título	Q-D
11	BM25 - título	Q-D
12	log(BM25) - título	Q-D
13	LMIR.DIR - título	Q-D
14	LMIR.JM - título	Q-D
15	LMIR.ABS - título	Q-D
16	tf - *abstract*	Q-D
17	$\log(tf(d)+1)$ - *abstract*	Q-D
18	$\frac{tf(d)}{\lvert d\rvert}$ - *abstract*	Q-D
19	$\log(\frac{tf(d)}{\lvert d\rvert} + 1)$ - *abstract*	Q-D
20	$\log(\frac{\lvert C\rvert}{df})$ - *abstract*	Q
21	$\log(\log(\frac{\lvert C\rvert}{df}))$ - *abstract*	Q
22	$\log(\frac{\lvert C\rvert}{tf(C)} + 1)$ - *abstract*	Q
23	$\log(\frac{tf(d)}{\lvert d\rvert} \cdot \log(\frac{\lvert C\rvert}{df}) + 1)$ - *abstract*	Q-D
24	$tf(d) \cdot \log(\frac{\lvert C\rvert}{df})$ - *abstract*	Q-D
25	$\log(\frac{tf(d)}{\lvert d\rvert} \cdot \frac{\lvert C\rvert}{tf(C)} + 1)$ - *abstract*	Q-D
26	BM25 - *abstract*	Q-D
27	log(BM25) - *abstract*	Q-D
28	LMIR.DIR - *abstract*	Q-D
29	LMIR.JM - *abstract*	Q-D
30	LMIR.ABS - *abstract*	Q-D
31	tf - "título + *abstract*"	Q-D
32	$\log(tf(d)+1)$ - "título + *abstract*"	Q-D
33	$\frac{tf(d)}{\lvert d\rvert}$ - "título + *abstract*"	Q-D
34	$\log(\frac{tf(d)}{\lvert d\rvert} + 1)$ - "título + *abstract*"	Q-D
35	$\log(\frac{\lvert C\rvert}{df})$ - "título + *abstract*"	Q
36	$\log(\log(\frac{\lvert C\rvert}{df}))$ - "título + *abstract*"	Q
37	$\log(\frac{\lvert C\rvert}{tf(C)} + 1)$ - "título + *abstract*"	Q
38	$\log(\frac{tf(d)}{\lvert d\rvert} \cdot \log(\frac{\lvert C\rvert}{df}) + 1)$ - "título + *abstract*"	Q-D
39	$tf(d) \cdot \log(\frac{\lvert C\rvert}{df})$ - "título + *abstract*"	Q-D
40	$\log(\frac{tf(d)}{\lvert d\rvert} \cdot \frac{\lvert C\rvert}{tf(C)} + 1)$ - "título + *abstract*"	Q-D
41	BM25 - "título + *abstract*"	Q-D
42	log(BM25) - "título + *abstract*"	Q-D
43	LMIR.DIR - "título + *abstract*"	Q-D
44	LMIR.JM - "título + *abstract*"	Q-D

45	LMIR.ABS - “título + *abstract*”	Q-D

Tabela 2.3: *Características da coleção OHSUMED*

CAPÍTULO 3

Trabalhos Relacionados

Nesse capítulo são apresentados alguns trabalhos relacionados à utilização de regras de associação para o problema de ordenação de documentos em sistemas *learning to rank*. Na primeira seção encontra-se um trabalho relacionado a regras de associação. Nele realiza-se um estudo sobre o grau de interesse das medidas de avaliação de regras de associação para o problema de classificação. Na segunda seção encontram-se algoritmos sobre seleção de características em L2R.

3.1 Regras de associação

Normalmente regras de associação são utilizadas em problemas de classificação de documentos. Recentemente foram criadas abordagens que utilizam regras de associação para os problemas de ordenação de documentos. Apesar de serem problemas com características diferentes, algumas dificuldades e abordagens de problemas de classificação com regras de associação podem ser adaptadas para os problemas de ordenação.

A seguir é apresentado um trabalho correlato à presente dissertação. Nele são discutidos fatores que influenciam o grau de importância de uma regra de associação para a classificação de um documento. Dois desses fatores foram utilizados como inspiração.

3.1.1 *On Rule Interestingness Measures*

Um aspecto crucial ao se utilizar de regras de associação é que o conhecimento descoberto dado através das regras deve ser, de algum modo, interessante (ou útil) para a classificação ou ordenação de documentos. Um grau de interesse de uma regra pode ser dividido em duas categorias: objetivo, em que é relacionado com os dados e subjetivo, que é relacionado com o usuário. Medidas de interesse de regras objetivas são baseadas em dados das regras, como por exemplo probabilidade de ocorrência de um antecedente, ou consequente. Medidas de interesse de regras subjetivas são baseadas em se comparar regras encontradas com os conhecimentos ou crenças prévios do usuário. As medidas de interesse de regras objetivas podem ser utilizadas como um primeiro filtro para a seleção

de potenciais regras. Medidas subjetivas possuem a vantagem de serem uma medida direta do interesse de um usuário a uma regra, mas possui desvantagens. Primeiro porque requerem uma especificação das crenças do usuário ou conhecimento prévio como entrada e em segundo porque são fortemente dependentes de um domínio e usuário.

O artigo [16] inicialmente apresenta diversos fatores que influenciam o grau de interesse de uma regra de associação. Os fatores que inspiraram o presente trabalho de dissertação foram os dois primeiros fatores.

O primeiro fator apresentado é o tamanho da disjunção. Um conjunto de regras pode ser considerado como uma disjunção de regras, de modo que uma dada regra pode ser considerada como uma disjunção. O tamanho de uma disjunção (regra) é o número de tuplas satisfeita pela regra. Uma disjunção é pequena quando o número de tuplas abrangidas por uma regra for pequena de acordo com algum limiar. Disjunções pequenas podem ser consideradas ruídos ou exceções, porém se todas as regras com tal característica forem removidas, a acurácia da predição pode ficar prejudicada. Ou seja, as regras com baixa abrangência de tuplas que forem exceções devem ser mantidas e apenas aquelas que forem ruídos devem ser eliminadas. Para resolver esse problema é sugerido que sejam tratados de forma diferente as disjunções pequenas das grandes.

O segundo fator apresentado é o desbalanceamento da distribuição de classes. Uma classe é considerada desbalanceada se o número de antecedentes levando a tal classe for menos frequente ou mais rara que das outras classes. Quando há o desbalanceamento de classes, é relativamente fácil encontrar regras predizendo a classe maior, mas em contrapartida é muito difícil encontrar regras predizendo a classe menor. Assim, quanto menos regras existem para predizer uma classe, mais interessante é tal regra. De forma análoga, quanto mais regras existem para predizer uma classe, menos interessante é tal regra.

3.2 Seleção de características

Existem muitas abordagens de seleção de características que são utilizadas em problemas de classificação de documentos, porém os problemas de ordenação comportam-se de forma diferente. Sendo assim os métodos existentes para seleção de características em problemas de classificação não são ideais para aqueles de ordenação de documentos tornando assim, necessária a criação de novos métodos de seleção de característica para tais problemas.

Nas seções seguintes são abordados como três trabalhos realizaram a seleção de características. O primeiro nomeado "*Feature selection for Ranking*", o segundo "*Feature Selection for Document Ranking using Best First Search and Coordinate Ascent*" e o terceiro "*FP-Rank: An Effective Ranking Approach Based on Frequent Pattern Analysis*".

Os dois primeiros trabalhos realizam uma seleção de características e testam se a base de dados apenas com as características selecionadas conseguem resultados melhores que a base de dados original. O trabalho "*FP-Rank: An Effective Ranking Approach Based on Frequent Pattern Analysis*" realiza uma seleção de características para posteriormente expandir o documento de teste e treino. Os dois primeiros trabalhos possibilitaram a elucidação de como realizar a seleção de características, já o último trabalho serviu como inspiração para a geração de características para o problema de ordenação de resultados de máquina de busca.

3.2.1 Feature selection for Ranking

O trabalho de Geng et al [19] realiza uma seleção de características do tipo filtro para o problema de ordenação de documentos. Primeiro os autores utilizam o valor de cada característica para ordenar os documentos. Posteriormente avaliam cada uma dessas ordenações realizadas pelos valores de cada característica utilizando as medidas de avaliação MAP e NDCG. O valor de MAP e NDCG encontrado será considerado o escore de importância para aquela característica.

Com os escores de importância de cada característica, os autores ordenam as características de acordo com tais escores. O artigo ressalta que se for utilizado uma medida de avaliação como por exemplo a função de perda (do inglês *loss function*) em que quanto menor o valor, melhor a característica, o tipo de ordenação (crescente ou decrescente) irá mudar, porém as características sempre serão ordenadas de tal forma que a melhor esteja no topo da lista.

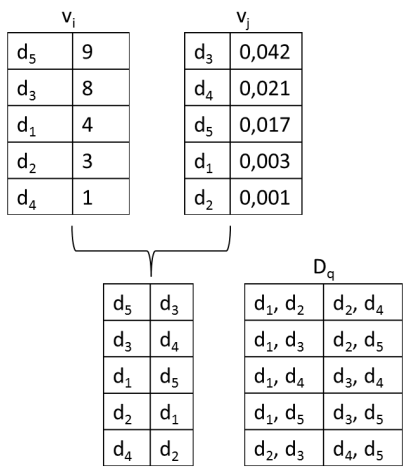
Com a ordem de importância das características, é realizada em [19] uma remoção de características similares com menores valores de importância. A lista de características ordenada por importância é percorrida, a cada novo elemento da lista, é verificado se ele não é semelhante a outra característica já percorrida, ou seja, que possui maior importância. Como não é ideal utilizar características redundantes esse processo faz-se necessário. A similaridade entre as características é percebida pela semelhança entre o resultado de ranking que elas produzem. Para tanto, é utilizado o coeficiente de similaridade de Kendall.

O coeficiente de similaridade de Kendall utilizado no artigo pode ser calculado da seguinte forma:

$$\tau_q(v_i, v_j) = \frac{|(d_s, d_t) \in D_q | d_s \prec_{v_i} d_t \; E \; d_s \prec_{v_j} d_t|}{|(d_s, d_t) \in D_q|} \tag{3-1}$$

Onde v_i e v_j são as características a ser calculada a similaridade entre elas, D_q é o conjunto de pares (d_s, d_t) em resposta a uma consulta q. $d_s \prec_{v_i} d_t$ implica que o documento d_s é menos relevante que o documento d_t para a característica v_i.

Um exemplo de como é calculada a similaridade entre as características pode ser visto na figura 3.1 a seguir.



$$\tau_q(v_i, v_j) = \frac{1+1+0+1+1+0+1+1+0+0}{10}$$

$$\tau_q(v_i, v_j) = 0{,}6$$

Figura 3.1: *Figura exemplo da fórmula de Kendall*

No topo da figura 3.1 é possível ver a tabela com os rankings das características v_i e v_j. Abaixo está a tabela comparando o ranking produzido por v_i e v_j. d_1 a d_5 são exemplos de documentos para ilustrar a utilização da fórmula de kendall 4-1. D_q possui todos os possíveis pares (d_s, d_t) resultantes da combinação dos documentos d_1 a d_5. O numerador da fórmula de kendall 4-1 é o somatório da concordância entre a ordenação de dois documentos no ranking de cada característica. Se dois documentos possuírem a mesma ordem de relevância em ambas características, será atribuído 1 à concordância entre elas, caso contrário será atribuído 0. Tome como exemplo o par d_1, d_2. Na característica v_i, o documento d_1 é mais relevante que o documento d_2. O mesmo ocorre com a característica v_j. Sendo assim, há a concordância entre a ordenação dos documentos d_1 e d_2 em ambas características e atribui-se 1 à concordância.

Como a seleção de características é apenas uma etapa de pré-processamento e não um método de ordenação de documentos, faz-se necessário utilizar, modelos de ranking. Nos experimentos de [19] foram utilizados dois modelos de ranking: Ranking SVM e RankNet.

Com os resultados dos experimentos, obteve-se como conclusão que a seleção de características pôde melhorar o ranking mais significantemente para a coleção .gov que para a OHSUMED e que o método de seleção de características proposto obteve melhores resultados que os métodos tradicionais de seleção de características para problemas de classificação.

Esse trabalho se assemelha a esse trabalho de dissertação por realizar uma seleção de características e utilizar, posteriormente um modelo de ranking para verificar a qualidade da seleção realizada.

Outro trabalho cujo tema é correlato com esse trabalho de dissertação é "*Feature Selection for Document Ranking using Best First Search and Coordinate Ascent*".

3.2.2 Feature Selection for Document Ranking using Best First Search and Coordinate Ascent

Diferentemente do trabalho da seção anterior, o trabalho de Dang et al [14] realiza uma seleção de características do tipo *wrapper*. O método proposto realiza uma modificação simples do método tradicional *wrapper* que pode ser encontrada no trabalho [26].

O objetivo do trabalho é particionar o conjunto inicial de características de forma gulosa para formar k subconjuntos não coincidentes de características (cada subconjunto com tamanho no máximo s). Juntamente com um conjunto de rankings aprendidos que maximizam uma métrica (MAP ou NDCG, por exemplo). O método de ranking utilizado nesse trabalho foi o *Coordinate Ascent* e o tamanho dos subconjuntos formados foram entre 2 e 4, ou seja, $2 \leq s \leq 4$.

Para realizar a partição das características em subconjuntos são realizadas as seguintes etapas: primeiro são colocadas todas as características em um conjunto. Depois é construído um grafo não orientado onde cada nó representa um subconjunto de características com tamanho máximo s. Uma aresta ligará dois vértices se um dos vértices puder ser obtido pelo outro vértice adicionado de exatamente uma característica. Após construído o grafo, é realizado um procedimento de busca *best-first* descrito no algoritmo 3.1 abaixo.

Algoritmo 3.1: Algoritmo Best-first para seleção de características

Entrada: Grafo não orientado G com vértice v composto por subconjunto de características, Método de ranking $\mathcal{A}$, Conjunto de treino $\mathcal{T}$.

Saída: Vértice que contem o melhor subconjunto de características.

```
1  P = ∅
2  melhor = null
3  Randomicamente escolha um vértice v.
4  Treine A em T usando v para maximizar uma métrica
5  Adicione v a P
6  enquanto |P| > 0 faça
7      v ← o vértice com melhor valor de métrica em P
8      Remove v de P
9      se O valor da métrica de v >valor da métrica de melhor então
10         melhor ← v
11     fim
12     se melhor não modificou por 5 rodadas então
13         PARE e RETORNE melhor
14     fim
15     para cada vizinho de v, u faça
16         Treine A em T usando u para maximizar uma métrica
17         Adicione u a P
18     fim
19 fim
20 RETORNE melhor
```

Após encontrado o vértice com melhor métrica, todas as características contidas nele são removidas do conjunto de características total e um novo grafo não orientado é criado. Esse processo é repetido até que o conjunto de características total seja vazio. O resultado dessa etapa é o particionamento do conjunto de características em subconjuntos com 2 a 4 características. Uma vez que o processo de particionamento das características termina, é definida uma nova característica correspondente para cada vértice (subconjunto de características) de tal forma que, para cada documento, a característica receberá o valor do escore atribuído pelo modelo de ranking ao documento. Assim, o vetor original de um documento $\{x_1, x_2, ..., x_{|X|}\}$ ficaria $\{x'_1, x'_2, ..., x'_k\}$ com k igual ao número de particionamentos e $k < |X|$. Esse novo vetor de características é então utilizado no processo de aprendizado.

Assim que a fase de pré-processamento termina, para cada tamanho de subconjunto, são selecionados os j melhores subconjuntos, $j \in 1..5$, para treinar o modelo de

ranking e utilizado o conjunto de validação para selecionar j. São utilizados os modelos de *ranking RankNet*, *RankBoost*, *AdaRank* e *Coordinate Ascent* para avaliar as mudanças com a seleção de características.

Os resultados dos experimentos, obteve como conclusão que a seleção de características com o método *wrapper* pode melhorar significantemente os valores de métricas comparando-se com o original sem a seleção de características, e também com apenas uma seleção gulosa das características. Os melhores resultados, ou seja, com maior ganho de valores de métrica, foram obtidos utilizando o método de ranking RankNet, pois por se tratar de um método baseado em redes neurais é difícil de ser treinado, portanto a seleção de características ajuda bastante em sua eficácia.

Outra conclusão importante do trabalho foi a de que a seleção gulosa das características nem sempre resulta em um bom resultado. Apenas o fez para o método RankNet.

Por último percebe-se com os resultados que os melhores valores para o tamanho do subconjunto de características foram 3 e 4.

3.2.3 FP-Rank: An Effective Ranking Approach Based on Frequent Pattern Analysis

A maioria dos esforços foram gastos para encontrar algoritmos de treinamento de modelos de classificação mais eficientes, enquanto poucos estudos foram realizados para melhorar a qualidade das características dos documentos utilizadas nas abordagens de L2R. Além do algoritmo de treinamento de modelos de classificação, a performance de um sistema de recuperação da informação depende das características utilizadas para classificar. Um algoritmo treinado com características ruins terá pior performance que se for treinado com características boas. Assim, pode-se verificar a importância de se selecionar boas características para os algoritmos de treinamento.

Em [45] é investigada a possibilidade de selecionar as características mais frequentes no treino que possuem maior significância e menor redundância. É proposto um novo método de ordenação FP-Rank que otimiza o conjunto de características utilizadas em algoritmos de L2R para melhorar a precisão dos métodos de ordenação.

Considere um padrão α. D_α é o subconjunto de documentos da coleção (D) em que o padrão α ocorre. Dado um limiar θ_0, um padrão é considerado frequente se $P(\alpha) = \frac{|D_\alpha|}{|D|} \geq \theta_0$.

O trabalho enfatiza que nem todos os padrões frequentes contribuem igualmente para a classificação. *Stopwords* são um bom exemplo de que nem sempre padrões muito frequentes são úteis. Assim é necessário que haja uma nova seleção após a seleção

dos padrões mais frequentes a fim de selecionar nestes padrões frequentes aqueles que deixarão o conjunto com uma alta significância e uma baixa redundância.

Após a seleção das características mais frequentes e mais significantes, o trabalho realiza uma expansão dos conjuntos de treino e teste com tais características e, utilizando os dados expandidos, aplica-se o algoritmo RankSVM e RankNet.

O estudo confirma que padrões frequentes oferecem características de alta qualidade que podem ser utilizadas para melhorar a performance de um modelo de ranking. Comparado com as abordagens comumente utilizadas de seleção de características, o FP-Rank é capaz de encontrar um subconjunto padrão que é específico para o problema de ordenação.

3.3 Paralelo com os trabalhos relacionados

O trabalho apresentado por [48] foi utilizado como inspiração para o presente trabalho por se utilizar de regras de associação em problemas de ordenação de resultados de máquinas de busca. O artigo [16] elucidou sobre os diversos fatores que influenciam o grau de interesse de uma regra de associação. Os trabalhos [19], [14] e [26] abordaram formas de se realizar seleção de características para o problema de ordenação. Em [26] foi realizada uma expansão dos conjuntos de treino e teste com características selecionadas que serviu de inspiração para a realização do presente trabalho.

CAPÍTULO 4

Utilização de regras de associação em Learning to Rank

Nesse capítulo apresentam-se as soluções que foram investigadas para os problemas de pesquisa que o trabalho de dissertação propôs a analisar, apresentados na seção 1.3.

Na primeira seção serão apresentadas as propostas que foram investigadas para a solução do problema de pesquisa 1. E na segunda seção serão apresentadas as soluções para o problema de pesquisa 2.

4.1 Uso do algoritmo LRAR com diferentes medidas de avaliação de regras de associação

Nesta seção serão apresentadas as diferentes alternativas apresentadas pelo presente trabalho para gerar ranking usando regras de associação com o objetivo de obter possíveis soluções para o problema de pesquisa 1 abordados na seção 1.3.

Questão de pesquisa 1 *A utilização das diferentes medidas de avaliação de regras de associação no algoritmo de ordenação LRAR pode trazer benefícios à acurácia desse algoritmo?*

Serão utilizadas outras medidas de avaliação de regras de associação (em adição às medidas de confiança) em um algoritmo de ordenação de documentos semelhante ao algoritmo descrito na seção 1.2. Esse algoritmo é o Learning to Rank using Association Rules (LRAR) proposto por Veloso et al. em [48] e é explicado na seção a seguir.

4.1.1 Learning to rank at query-time using association rules

Nessa seção será abordado o trabalho "Learning to rank at query-time using association rules" que utiliza regras de associação para o problema de ordenação de documentos através do algoritmo LRAR (Learning to Rank using Association Rules).

A maioria dos métodos *learning to rank* tentam otimizar uma métrica (MAP, NDCG, precisão e outros), porém neste trabalho são utilizadas regras de associação para criar um modelo R que será posteriormente utilizado para estimar a relevância dos documentos no conjunto de teste.

Dado um conjunto de treino $\mathcal{D}$, as regras de associação são criadas mapeando-se as características dos documentos às relevâncias. Assim, uma regra de associação teria o formato $\mathcal{X} \rightarrow r_i$ onde $\mathcal{X}$ é um conjunto de características e r_i é um grau de relevância.

Nesse trabalho, duas medidas são utilizadas para quantificar a qualidade de uma regra. São elas: o suporte e a confiança. O Suporte de uma regra é a fração de exemplos em $\mathcal{D}$ que contem a regra. A confiança de uma regra é a probabilidade condicional de r_i dado $\mathcal{X}$. Para garantir que exista uma forte correlação entre o antecedente e o consequente, uma confiança mínima é exigida na criação das regras. Se a regra não possuir esse mínimo a mesma será descartada. Da mesma forma, para evitar uma grande quantidade de regras um suporte mínimo também é exigido.

Como cada modelo gerado é específico de um documento d, apenas regras úteis para tal documento serão criadas. A geração de regras é adiada até que um conjunto de documentos seja considerado para uma dada consulta no documento de teste. Então, cada documento do teste é utilizado como um filtro para remover características e exemplos irrelevantes do documento de treino. Esse processo gera um subconjunto D_d do treino que é uma projeção do documento de treino D. Essa projeção foca apenas nas características úteis para ranquear um documento específico d. Para cada projeção, é então criado um modelo específico R_d para cada documento d associado a uma consulta. Como resultado, regras importantes que possuíam um valor baixo de confiança em D agora podem possuir um valor maior de confiança e dessa forma não seriam descartadas.

Para estimar a relevância de um documento, o método realizado por [48], utiliza o conjunto R_d. Cada regra de R_d é considerada como um voto para medir a relevância de um documento. Porém, como cada regra possui uma confiança diferente, cada voto também possuirá um peso diferente, de forma que regras com maior confiança possuam um maior peso na ordenação de um documento.

Para cada grau de relevância contido na coleção é computado um escore $s(r_i)$ da seguinte forma: Somam-se as confianças de cada regra cujo grau de relevância (especificado no consequente da regra) seja idêntico (r_i). Isto é, para cada grau de relevância haverá um escore $s(r_i)$ e esse será computado como o somatório das confianças das regras cujos graus de relevância é aquele do escore (r_i). Posteriormente divide-se esse somatório pela quantidade de regras com aquele grau de relevância. Assim o escore é calculado como uma média aritmética das confianças para cada grau de relevância como na fórmula 4-2:

$$s(r_i) = \frac{\sum_{X \to r_i \in R_d} \theta(X \to r_i)}{|R_d(r_i)|} \tag{4-1}$$

Onde $s(r_i)$ é o escore para o grau de relevância r_i, $\theta(\mathcal{X} \to r_i)$ é a confiança da regra $\mathcal{X} \to r_i$ e $|R_d(r_i)|$ é a quantidade de regras no conjunto R_d que possuem como antecedente o grau de relevância r_i

O escore é utilizado pelo método tanto para a classificação quanto para a ordenação. O grau de relevância que obtiver maior escore será considerada a classificação do documento para a consulta em questão. O peso para realizar a ordenação de um documento d é estimado pela combinação linear dos escores normalizados associados com cada relevância. A fórmula para o peso da ordenação é descrita abaixo:

$$rank = \sum_{i=0}^{k} r_i \times \frac{s(r_i)}{\sum_{j=0}^{k} s(r_j)} \tag{4-2}$$

O valor de ordenação é um valor estimado para a relevância real do documento d utilizando as regras do conjunto R_d.

Os resultados dos experimentos reportados em [48] indicaram que métodos que utilizam aprendizado em tempo de consulta conseguem melhor performance que muitos métodos de ordenação mais avançados como o RankingSVM, RankBoost, FRank e ListNet. Tem-se como resultado também que o tempo de execução é bastante veloz, dando margem a outros modelos utilizando tempo de consulta.

4.1.2 Medidas de avaliação de regras de associação

Para selecionar as regras de associação mais úteis (interessantes) dentro de um conjunto de regras, existem várias medidas de significância que podem ser utilizadas. As mais conhecidas entre elas são o suporte de uma regra e a confiança que já foram abordadas no capítulo 1 de introdução. Além dessas, existem outras como o suporte do antecedente, suporte do consequente e *lift*.

Suporte do antecedente

O suporte do antecedente se assemelha ao suporte de uma regra. Assim como seu semelhante, possui também um valor relativo e um valor absoluto. O suporte do antecedente absoluto é apenas uma contagem da ocorrência do antecedente na coleção. O suporte do antecedente relativo consiste da razão da ocorrência do antecedente da regra em toda a coleção, ou seja, o total de vezes que o antecedente da regra ocorre na coleção dividido pelo total de regras da coleção, conforme a equação 4-3:

$$\sigma(A) = \frac{\text{total de ocorrências do antecedente}}{\text{total de regras}} \tag{4-3}$$

Suporte do consequente

O suporte do consequente é muito parecido com o suporte do antecedente. Assim como seu semelhante, possui também um valor relativo e um valor absoluto. O suporte do consequente absoluto é apenas uma contagem da ocorrência do consequente na coleção. O suporte do consequente relativo consiste da razão da ocorrência do consequente da regra em toda a coleção, ou seja, o total de vezes que o consequente da regra ocorre na coleção dividido pelo total de regras da coleção, conforme a equação 4-4:

$$\sigma(C) = \frac{\text{total de ocorrências do consequente}}{\text{total de regras}} \tag{4-4}$$

Lift

Em regras de associação, *lift* é uma medida que indica o quanto mais importante se torna um antecedente quando um consequente ocorre. *Lift* pode ser obtido dividindo-se a confiança de uma regra pela probabilidade do consequente ou ainda pela divisão do suporte da regra pela multiplicação da probabilidade do antecedente e a probabilidade do consequente. Como pode ser vistas respectivamente pelas Equações 4-5 e 4-6:

$$lift(A \rightarrow C) = \frac{\theta(A \rightarrow C)}{P(C)} \tag{4-5}$$

$$lift(A \rightarrow C) = \frac{\sigma(A \rightarrow C)}{P(A) \times P(C)} \tag{4-6}$$

A probabilidade do antecedente $P(A)$ pode ser obtida dividindo-se o total de ocorrências do antecedente pelo total de regras da coleção. A probabilidade do consequente $P(C)$ pode ser obtida dividindo-se o total de ocorrências do consequente pelo total de regras da coleção.

Confiança invertida

O artigo de Freitas [16] discute uma série de fatores que influenciam a avaliação do grau de qualidade de uma regra de associação descoberta por algoritmos de mineração de dados. Apesar do artigo ser focado em problemas de classificação de documentos, algumas das observações expostas também podem ser aplicadas ao problema de ordenação de documentos com abordagem *learning to rank*.

A primeira observação diz respeito ao valor pequeno de suporte do antecedente de algumas regras. Se todas as regras que possuem suporte do antecedente pequeno

fossem desconsideradas, a acurácia da predição pode ser significantemente prejudicada. Nas coleções de problemas de ordenação encontra-se o mesmo tipo de problema, isto é, existem poucas regras com um grande valor de suporte do antecedente e muitas regras com um pequeno valor de suporte do antecedente. Para tratar tal problema o autor sugere que seja feita uma medida que possa avaliar de forma diferente as regras que possuam baixo e alto suporte do antecedente.

A segunda observação realizada em [16] diz respeito ao desbalanceamento da distribuição de classes. Quanto menor for a frequência relativa de uma classe, mais difícil será encontrar uma regra que consiga predizê-la, e, dessa forma, maior será a qualidade de tal regra. De forma análoga, regras pertencentes a uma classe mais frequente possuem uma menor qualidade de predizê-la e, assim, uma qualidade menor. Esse fator é geralmente ignorado em sistemas que se utilizam de regras de associação. Para tratar tal problema o autor sugere que seja realizada uma medida que privilegie regras de uma classe menos frequente em detrimento das regras de classes mais frequentes.

Visando buscar adequação ao problema de ordenação de documentos e preocupando-se com as observações expostas por [16], além das métricas tradicionais já exibidas, o presente trabalho de mestrado utiliza uma outra métrica. Tal métrica foi batizada de confiança invertida.

A confiança invertida é a probabilidade condicional de um antecedente dado o seu consequente $P(A|C)$. A confiança invertida é obtida pela divisão do total de ocorrências da regra pelo total de regras com o consequente C, como mostra a equação 4-7.

$$\theta_{inv} = \frac{\text{total de ocorrências da regra}}{\text{total de regras com o consequente } C} \tag{4-7}$$

4.1.3 Utilização de medidas de avaliação de regras no algoritmo LRAR

Nesta seção são apresentadas várias soluções para o problema de pesquisa 1. Elas modificam a função de escore do algoritmo LRAR, dada pela equação 4-2, de modo a substituir a medida de confiança da regra por outras medidas de avaliação de regras de associação.

Também foram propostas algumas variações no cálculo da função de escore que independem da medida de avaliação de regra utilizada. Por exemplo, a alteração do fator de normalização (denominador) do escore da equação 4-1. Essas alterações na função escore foram denominadas de rankings. Foram propostos 10 rankings que são descritos a seguir.

Além disso, foram feitos experimentos comparando os resultados em termos de MAP com a função $rank()$ tal como descrita na equação 4-2 e sua versão não normalizada, apresentada na equação 4-8:

$$rank = \sum_{i=0}^{k} r_i \times s(r_i) \tag{4-8}$$

Os experimentos não apresentaram perdas ou ganhos significativos. Portanto, adotou-se para todos os rankings propostos a seguir e também para o LRAR original a função $rank()$ descrita pela equação 4-8.

Ranking 1 - somatório das confianças dividido pelo total de regras

O objetivo desse ranking foi avaliar o efeito de se normalizar o escore pelo total de regras. O escore será dado pelo somatório das confianças das regras dividido pelo total de regras. Com isso, obtém-se o escore $s_1(r_i)$ definido pela equação 4-9:

$$s_1(r_i) = \frac{\sum_{X \to r_i \in R_d} \theta(X \to r_i)}{|R_d|} \tag{4-9}$$

Repare que a equação 4-9 difere-se da equação 4-1 apenas quanto ao valor do denominador.

Ranking 2 - somatório das confianças

O objetivo desse ranking foi o de testar variações em relação a normalização do escore de cada grau de relevância. Neste caso específico, não é feita nenhuma normalização do somatório das confianças. A função de escore $s_2(r_i)$ é dada pela equação 4-10:

$$s_2(r_i) = \sum_{X \to r_i \in R_d} \theta(X \to r_i) \tag{4-10}$$

Ranking 3 - somatório dos suportes absolutos das regras dividido pelo total de regras

O objetivo desse ranking foi de testar o suporte como medida de avaliação das regras de associação no algoritmo LRAR. Portanto, a medida da confiança na equação original do escore (equação 4-1) foi substituída pela medida de suporte, obtendo-se assim o novo escore $s_3(r_i)$ definido na equação 4-11, sendo $cont()$ o suporte absoluto:

$$s_3(r_i) = \frac{\sum_{X \to r_i \in R_d} cont(X \to r_i)}{|R_d|} \tag{4-11}$$

Ranking 4 - somatório dos suportes absolutos das regras

Esse ranking foi proposto com o objetivo de avaliar o uso do suporte absoluto como medida de avaliação de regra de associação no algoritmo LRAR. O escore utilizado nesse ranking é $s_4(r_i)$, definido pela equação 4-12:

$$s_4(r_i) = \sum_{X \to r_i \in R_d} cont(X \to r_i) \tag{4-12}$$

Ranking 5 - somatório da razão entre as confianças

Esse ranking foi proposto com o objetivo de avaliar o desempenho de uma nova medida de avaliação de regra de associação no algoritmo LRAR, dada pela razão entre a confiança de uma regra que tem como consequente o valor "um" (relevante) pela confiança de uma regra que tem o mesmo antecedente, com consequente igual a "zero" (não relevante). Essa medida obviamente somente pode ser utilizada nos casos em que os documentos da coleção possuem apenas dois níveis de relevância (1 relevante, 0 não relevante). Contudo, essa é a situação mais comum em problemas de recuperação de informação.

O objetivo de se usar a razão entre confianças é o de discriminar melhor documentos relevantes. Se um documento é relevante, deve haver regras do tipo $X \to 1$ e regras do tipo $X \to 0$ associadas a esse documento, tais que as regras do primeiro tipo têm valor de confiança maior do que as regras do segundo tipo.

A expectativa é que o somatório dessas razões deve ter um valor maior para documentos relevantes do que para documentos não relevantes. Com isso, os documentos com maior valor desse somatório têm maiores chances de serem documentos relevantes e aparecerão no início do ranking. O escore $s_5(r_i)$ utilizado nesse ranking é dado pela equação 4-13:

$$s_5(r_i) = \sum_{X \to r_i \in R_d} \frac{\theta(X \to 1)}{\theta(X \to 0)} \tag{4-13}$$

Ranking 6 - divisão do somatório das razões de confiança pelo número de regras com consequente i

Um problema com o uso de regras de associação para ordenar documentos é que o número de regras que possuem consequente igual a zero (indicam irrelevância) é muito maior que as regras com consequente igual a um. Com isso, o número de regras para as quais a razão $\frac{X \to 1}{X \to 0} \geq 1$ é muito pequeno.

Um modo de fazer com que essas regras contribuam mais com o escore para valor de relevância 1 $(s(1))$ do que para o escore de relevância 0 $(s(0))$, é utilizar um peso

maior para esse tipo de regra no cálculo do escore para o nível de relevância 1. Isso pode ser obtido dividindo-se a razão pelo número de regras cujo consequente coincide com o valor de relevância para o qual se quer computar o escore. Assim, o escore $s_6(r_i)$ para esse ranking é dado pela equação 4-14:

$$s_6(r_i) = \sum_{X \to r_i \in R_d} \frac{\frac{\theta(X \to 1)}{\theta(X \to 0)}}{|R_d(r_i)|} \tag{4-14}$$

Ranking 7 - somatório da diferença das confianças invertidas

Nesse ranking a confiança invertida é utilizada. Conforme explicado na seção 4.1.2, a confiança invertida corresponde a razão entre o suporte da regra e o suporte do consequente da regra. No problema de ordenação, documentos relevantes são raros, o que implica que o suporte do consequente 1 é muito pequeno em relação ao suporte do consequente 0. Portanto, o escore proposto neste ranking visa valorizar regras que possuem consequente 1 e que ocorrem com frequência no conjunto R_d de regras associadas ao documento d. O escore $s_7(r_i)$ utilizado neste ranking é apresentado na equação 4-15:

$$s_7(r_i) = \sum_{X \to r_i \in R_d} \theta_{inv}(X \to 1) - \theta_{inv}(X \to 0) \tag{4-15}$$

Ranking 8 - somatório da razão entre as confianças invertidas

Este ranking se assemelha ao Ranking 5, porém em vez de utilizar a razão entre as medidas de confiança de duas regras com mesmo antecedente, utiliza a razão entre os valores de confianças invertidas correspondentes. O escore $s_8(r_i)$ utilizado nesse ranking é dado pela equação 4-16:

$$s_8(r_i) = \sum_{X \to r_i \in R_d} \frac{\theta_{inv}(X \to 1)}{\theta_{inv}(X \to 0)} \tag{4-16}$$

Assim como ocorre no Ranking 5, a expectativa é que o somatório dessas razões deve ter um valor maior para documentos relevantes do que para documentos não relevantes. Com isso, os documentos com maior valor desse somatório têm maiores chances de serem documentos relevantes e aparecerão no início do ranking.

Ranking 9 - combinação dos escores utilizados no Ranking 7 e no Ranking 8

Os resultados apresentados no Capítulo 5 mostram que os rankings 8 e 9 obtiveram bons resultados. Portanto, o objetivo deste ranking é o de combinar os escores utilizados por aqueles dois rankings, visando melhorar a qualidade dos resultados dos

rankings individuais. Assim, o escore $s_9(r_i)$ representa essa combinação e é definido pela equação 4-17.

$$s_9(r_i) = \frac{s_7(r_i)+1}{2} + s_8(r_i) \tag{4-17}$$

A função de escore nesse experimento de ranking consistiu em modificar o intervalo da equação 4-15 para [0,1] e somar o valor obtido ao escore do ranking 8.

Ranking 10 - somatório dos *lifts*

A medida de avaliação *lift* é calculada pela divisão da confiança da regra pelo suporte do consequente, então quanto maior a confiança de uma regra (permanecendo o mesmo valor do suporte do consequente) maior será o valor de *lift*. Analogamente, quanto menor o valor do suporte do consequente (permanecendo o mesmo valor da confiança da regra), maior também será o valor de *lift*. Dessa forma, pode-se perceber que a medida *lift* ajuda a priorizar regras importantes.

Tome por exemplo duas regras: regra$_1$ e regra$_2$. Suponha que ambas regras possuam uma confiança de 0 e 9, porém o suporte do consequente da regra$_1$ seja maior que o da regra$_2$. Como mostrado em [16], regras que pertençam a uma classe desbalanceada maior tendem a ser menos interessantes que regras que pertençam a uma classe menor. Os problemas de ordenação, geralmente possuem muitos documentos que levam a um grau de relevância baixo e poucos com altos graus de relevância, e então tendem a possuir classes desbalanceadas. Dessa forma percebe-se que o *lift* consegue medir o grau de interesse de uma regra de associação para os problemas de ordenação.

A equação 4-18 mostra como é calculado o valor de escore utilizando-se do *lift*. Este escore e, por consequência, este ranking podem ser utilizados em coleções em que valor de relevância dos documentos pertence a um conjunto com mais de dois elementos.

$$s_{10}(r_i) = \sum_{X \to r_i \in R_d} lift(X \to r_i) \tag{4-18}$$

4.2 Uso de regras de associação para gerar novas características

Nesta seção serão apresentadas diferentes alternativas para gerar ranking usando regras de associação com o objetivo de obter possíveis soluções para o problema de pesquisa 2.

Problema de pesquisa 2: *a expansão da base de dados com novas características obtidas a partir de regras de associação podem trazer benefícios para os métodos de L2R existentes?*

4.2.1 Expansão da base de dados e utilização nos modelos Rankboost e Randomforest

Assim como mostrado na seção 3.2.3, o algoritmo FP-Rank realiza uma seleção de padrões de características (características únicas ou combinadas) frequentes, dentro desse grupo frequente é realizada uma seleção dos padrões mais significantes que não sejam semelhantes entre si. Após selecionados os melhores padrões de acordo com [45] é realizada uma expansão das bases de dados de teste e treino e executado o método de ordenação RankSVM para a base de dados expandida.

De forma análoga, foi criada a abordagem de expansão da base de dados com novas características derivadas de regras de associação. Essa abordagem consiste em selecionar as melhores medidas de avaliação de regras de associação e comparar os resultados obtidos no experimento utilizando o algoritmo LRAR. Uma vez obtidas as melhores características, as bases de dados seriam expandidas e algoritmos de aprendizagem como Rankboost e Randomforest foram utilizados para aprender a ordenar os documentos.

A expansão da base de dados foi realizada de duas maneiras diferentes. Essas duas abordagens diferentes de expansão são explicadas nas seções seguintes.

Expansão das características mais interessantes

A expansão das características mais interessantes foi realizada em quatro etapas:

1. geração das regras de associação;
2. ordenação das regras geradas;
3. poda das regras;
4. realimentação das bases de dados de treino e teste.

Para a realização da primeira etapa, fez-se necessário discretizar os dados utilizando a ferramenta Weka, assim como na solução para a questão de pesquisa 1), pois para gerar regras de associação com as características de um documento, tais características não podem possuir valores reais, mas discretos.

A primeira etapa consistiu em gerar todas as possíveis regras de uma base de dados de treino utilizando o algoritmo Apriori [1]. Foram geradas regras sem limite mínimo de suporte e confiança. O intuito de gerar regras sem limite mínimo de suporte e confiança foi o de tentar utilizar as regras interessantes que se enquadram no primeiro problema do artigo [16], ou seja, aquelas que possuam suporte pequeno, mas não são consideradas ruídos.

[1] A implementação do algoritmo Apriori utilizada nesse trabalho de dissertação pode ser encontrada em: http://www.borgelt.net/apriori.html

A segunda etapa consistiu em realizar a ordenação das regras geradas pelo algoritmo Apriori de acordo com algumas medidas de avaliação de regra de associação.

Foram realizadas as expansões com quatro medidas de avaliação de regras de associação que obtiveram os melhores resultados nos experimentos executados utilizando o algoritmo LRAR (na análise das soluções para a questão de pesquisa 1). Tais resultados são reportados no Capítulo 5 de resultados experimentais.

As quatro melhores medidas que foram utilizadas para expandir características das coleções foram: confiança, diferença da confiança invertida, *lift* e suporte da regra. Sendo assim, nessa etapa, foram geradas quatro ordenações distintas, cada uma utilizando uma medida de avaliação diferente.

A terceira etapa dessa abordagem realizou uma poda das regras. Tal poda era realizada de forma gulosa. As regras eram adicionadas apenas se não houvesse uma regra mais interessante que ela (ou seja, mais ao topo da lista de ordenação) em que seus antecedentes fossem semelhantes. Um antecedente foi considerado semelhante se houvesse ao menos um padrão idêntico. Considera-se um padrão idêntico um conjunto com uma ou mais características idênticas. A etapa de poda das regras foi realizada para cada uma das diferentes ordenações da etapa anterior.

Após a poda das regras, foi realizada a quarta etapa que consistiu na expansão da base de dados de treino, de teste e de validação. Nessa etapa, o conjunto dos antecedentes de uma regra é considerado como uma nova característica. Cada nova coluna gerada na base de dados significa um conjunto de antecedentes e o valor dessa nova característica gerada seria igual a zero se o conjunto de antecedentes não estivesse presente no documento (do arquivo de teste, treino ou validação) em questão ou adquiria valor da medida de avaliação da regra que continha o conjunto de antecedentes caso contrário.

Essa etapa foi realizada expandindo-se os “k” melhores conjuntos dos antecedentes. Os valores atribuídos a “k” nesse trabalho foram 10, 100 e 200.

Como descrito na seção anterior, após realizada a expansão, foram utilizados os algoritmos Rankboost e Randomforest para confrontar se a base de dados expandida obteria melhores resultados que a base de dados sem expansão.

Expansão com o somatório das características mais interessantes

A expansão com o somatório dos padrões mais interessantes foi realizada nas mesmas quatro etapas descritas acima, porém diferenciando-se apenas na quarta etapa que corresponde à expansão das bases de dados de treino, teste e validação.

A etapa de expansão nessa abordagem gera apenas uma nova característica nas bases de dados de teste, treino e validação. Ao contrário da expansão feita com as características mais interessantes nas bases de dados, essa abordagem gera apenas uma

nova característica que consiste do somatório dos valores das cem melhores características usadas na expansão adotada na abordagem anterior.

O objetivo com esse método foi o de eliminar a quantidade de valores iguais a zero que ocorrem nas características expandidas pela abordagem de expansão com as melhores características, visando obter melhores resultados de MAP.

CAPÍTULO 5

Resultados Experimentais

Nesse capítulo são apresentados os resultados dos experimentos com a utilização de regras de associação para o problema de ordenação de documentos para responder às questões de pesquisas expostas em 1.3.2.

Na seção 5.1 são apresentados os resultados de experimentos com todos os rankings discutidos na seção 4.1.3, propostos como possíveis soluções da questão de pesquisa 1.

Na seção 5.2 encontram-se os resultados para as abordagens de expansão das bases de dados utilizando regras de associação. Ou seja, experimentos com soluções para a questão de pesquisa 2. Na subseção 5.2.1 são apresentados os resultados da abordagem que realiza a expansão da base de dados com novas características mais interessantes utilizando uma medida de avaliação de regra de associação para ordenar as regras de associação. Na subseção 5.2.2 encontram-se os resultados da abordagem que realiza uma expansão da base de dados com o somatório das 100 características mais interessantes baseadas em uma medida de avaliação de regra de associação.

Por questões de testes e para facilitar a comparação entre todas as abordagens, os experimentos do presente trabalho de dissertação foram realizados utilizando as bases de dados TD2003 e HP2003 da LETOR 3.0. Tais coleções foram escolhidas devido ao grau de dificuldade de conseguir bons valores de MAP. A base de dados HP2003 possui um baixo grau de dificuldade enquanto a base de dados TD2003 possui um alto grau de dificuldade.

5.1 Resultados da modificação da função de ordenamento no algoritmo LRAR

Nessa seção são apresentados os resultados das abordagens apresentadas na seção 4.2.1. Em todos os experimentos dessa seção foi utilizado o algoritmo LRAR [1]. Na

[1] A implementação do algoritmo pode ser encontrada em: http://homepages.dcc.ufmg.br/~adrianov/software/ltar

implementação original do algoritmo LRAR foram modificadas as funções de ordenação como explicado na seção 4.2.1 de acordo com cada medida de avaliação de regra de associação desejada. Como mostrado em [16] se forem removidas todas as regras com suportes baixos a acurácia da predição pode ser afetada, pois além de ruídos podem ser eliminadas também regras que se tratam de exceções. Portanto, como a intenção do trabalho é de buscar as melhores alternativas de priorizar regras interessantes para a ordenação de documentos, foram retiradas as restrições de suporte e confiança mínimos do algoritmo LRAR utilizados em [48]. É importante ressaltar que limiares de suporte e confiança mínimos beneficiam a confiança como medida de avaliação de regra, pois no referido artigo os valores de MAP são superiores aos reportados pelo presente trabalho que desconsidera tais limiares.

Para medir o impacto do tamanho de regra na acurácia da predição, o algoritmo LRAR foi executado três vezes, para cada função nova de ordenação. A primeira execução utilizando apenas regras de tamanho 2, a segunda apenas regras de tamanho 3 e a terceira apenas regras de tamanho 4.

5.1.1 Resultados do Ranking 1 - somatório das confianças dividido pelo total de regras

A tabela 5.1 permite visualizar os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 1 (somatório das confianças dividido pelo total de regras) para a ordenação de documentos. Os resultados desse Ranking são comparados com o ranking original com as restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 1	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1417	0,4849
3	0,2992	0,7677	0,2430	0,5580
4	**0,3021**	**0,7748**	**0,2610**	**0,6073**

Tabela 5.1: *Resultados de MAP para o Ranking 1*

O tamanho de regra que obteve os melhores resultados com o ranking em questão foi o tamanho de regra 4. Apesar de não ter ultrapassado os valores originais do algoritmo LRAR, deve ser levado em consideração que o ranking foi executado sem limites mínimos de suporte e confiança.

5.1.2 Resultados do Ranking 2 - somatório das confianças

A tabela 5.2 mostra os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 2 (somatório das confianças) e para o algoritmo LRAR original ambos com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 2	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1418	0,4849
3	0,2992	0,7677	**0,2616**	**0,6101**
4	**0,3021**	**0,7748**	0,2425	0,5574

Tabela 5.2: *Resultados de MAP para o Ranking 2*

Diferente da maioria dos resultados dos rankings, o ranking 2 obteve seu melhor resultado com regras de tamanho 3, porém não chegaram a ultrapassar os valores obtidos com o algoritmo original LRAR com as restrições de suporte e confiança mínimos.

5.1.3 Resultados do Ranking 3 - somatório dos suportes absolutos das regras dividido pelo total de regras

A tabela 5.3 mostra os resultados de MAP obtidos da execução do algoritmo LRAR com a função de ordenação alterada para o Ranking 3 (somatório dos suportes absolutos das regras dividido pelo total de regras) comparados com os do ranking original do algoritmo LRAR com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 3	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1649	0,4925
3	0,2992	0,7677	0,2292	0,5735
4	**0,3021**	**0,7748**	**0,2534**	**0,6342**

Tabela 5.3: *Resultados de MAP para o Ranking 3*

5.1.4 Resultados do Ranking 4 - somatório dos suportes absolutos das regras

A tabela 5.4 permite visualizar os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 4 (somatório dos suportes absolutos das regras) para a ordenação de documentos. Os resultados do Ranking 4 são comparados com o ranking original com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 4	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1649	0,4925
3	0,2992	0,7677	0,2302	0,5735
4	**0,3021**	**0,7748**	**0,2539**	**0,6345**

Tabela 5.4: *Resultados de MAP para o Ranking 4*

5.1.5 Resultados do Ranking 5 - somatório da razão entre as confianças

A tabela 5.5 mostra os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 5 (somatório da razão entre as confianças) e os valores de MAP obtidos para o algoritmo LRAR original ambos sem as restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 5	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,2509	0,5194
3	0,2992	0,7677	0,2180	0,5867
4	**0,3021**	**0,7748**	0,1983	**0,6362**

Tabela 5.5: *Resultados de MAP para o Ranking 5*

Diferente da maioria dos resultados dos rankings, o ranking 5 obteve seu melhor resultado com regras de tamanho diferentes para cada base de dados. Para a base TD2003, o melhor resultado foi obtido utilizando apenas regras de tamanho 2, enquanto para a base de dados HP2003, o melhor resultado foi obtido utilizando apenas regras de tamanho 4, como a maioria dos rankings.

Os resultados desse ranking não chegaram a ultrapassar os valores obtidos com o algoritmo original LRAR com as restrições de suporte e confiança mínimos.

5.1.6 Resultados do Ranking 6 - divisão do somatório das razões de confiança pelo número de regras com consequente i

Na tabela 5.6 é apresentado os resultados obtidos através da execução do algoritmo LRAR utilizando o Ranking 6 (divisão do somatório das razões de confiança pelo número de regras com consequente i) como função de ordenação comparado com a execução original do algoritmo ambos sem restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 6	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	**0,2509**	0,5194
3	0,2992	0,7677	0,2168	0,5871
4	**0,3021**	**0,7748**	0,1983	**0,6362**

Tabela 5.6: *Resultados de MAP para o Ranking 6*

Assim como nos resultados do ranking 5, o ranking 6 obteve seu melhor resultado com regras de tamanho diferentes para cada base de dados. Também obteve para a base TD2003 o melhor resultado utilizando apenas regras de tamanho 2 e para a base de dados HP2003 utilizando apenas regras de tamanho 4.

5.1.7 Resultados do Ranking 7 - somatório da diferença das confianças invertidas

A tabela 5.7 permite visualizar os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 7 (somatório da diferença das confianças invertidas) para a ordenação de documentos. Os resultados do Ranking 7 são comparados com o ranking original com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 7	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1772	0,6957
3	0,2992	0,7677	0,2268	0,6975
4	**0,3021**	**0,7748**	**0,2589**	**0,7029**

Tabela 5.7: *Resultados de MAP para o Ranking 7*

5.1.8 Resultados do Ranking 8 - somatório da razão entre as confianças invertidas

A tabela 5.8 mostra os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 8 (somatório da razão entre as confianças invertidas) e para o algoritmo LRAR original ambos com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 8	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	**0,2509**	0,5194
3	0,2992	0,7677	0,2168	0,5846
4	**0,3021**	**0,7748**	0,1889	**0,6316**

Tabela 5.8: *Resultados de MAP para o Ranking 8*

Assim como nos resultados do ranking 5 e 6, o ranking 8 obteve seu melhor resultado com regras de tamanho diferentes para cada base de dados. Também obteve para a base TD2003 o melhor resultado utilizando apenas regras de tamanho 2 e para a base de dados HP2003 utilizando apenas regras de tamanho 4.

5.1.9 Resultados do Ranking 9 - combinação dos escores utilizados no Ranking 7 e no Ranking 8

A tabela 5.9 mostra os resultados de MAP obtidos da execução do algoritmo LRAR com a função de ordenação alterada para o Ranking 9 (utilização dos Rankings 7 e 8) comparados com os do ranking original do algoritmo LRAR com restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 9	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,1861	**0,7035**
3	0,2992	0,7677	0,1874	0,6878
4	**0,3021**	**0,7748**	**0,2054**	0,6840

Tabela 5.9: *Resultados de MAP para o Ranking 9*

O ranking 9 obteve melhor resultado com tamanhos de regra diferentes para cada base de dados. Assim como a maioria dos resultados, para a base de dados TD2003, o ranking 9 obteve melhor resultado utilizando apenas regras de tamanho 4. Porém, para a base de dados HP2003, o melhor resultado foi obtido utilizando apenas regra de tamanho 2.

5.1.10 Resultados do Ranking 10 - somatório dos *lift*

A tabela 5.10 permite visualizar os valores de MAP obtidos para o algoritmo LRAR utilizando o Ranking 10 (somatório dos *lift*) para a ordenação de documentos. Os resultados do Ranking 10 são comparados com o ranking original sem as restrições de suporte e confiança mínimos.

Tamanho de regra	Original		Ranking 10	
	TD2003	HP2003	TD2003	HP2003
2	0,2760	0,7614	0,2581	0,5181
3	0,2992	0,7677	0,2703	0,6137
4	**0,3021**	**0,7748**	**0,2766**	**0,6554**

Tabela 5.10: *Resultados de MAP para o Ranking 10*

O tamanho de regra que obteve os melhores resultados com o ranking em questão foi o tamanho de regra 4. Os valroes de MAP obtidos não ultrapassaram os valores originais do algoritmo LRAR.

Como pôde ser observado nos resultados da modificação da função de ordenamento no algoritmo LRAR, o tamanho de regra que gerou os melhores resultados na maioria das bases de dados e rankings foi o tamanho 4. Apesar dos resultados obtidos não terem ultrapassado os resultados originais do algoritmo LRAR, deve ser considerado que os experimentos realizados com os novos rankings não possuíram limite mínimo de suporte e confiança.

5.2 Resultados da expansão da base de dados

Nessa seção são apresentados os resultados das abordagens que realizam a expansão da base de dados de treino, teste e validação conforme explicado na seção 4.2. A expansão foi realizada de duas formas diferentes. A primeira abordagem consistiu em expandir as bases de dados de tal forma que cada nova característica criada é um conjunto de um ou mais antecedentes de uma regra de associação, sendo assim, nessa abordagem será criada uma característica para cada conjunto de antecedente. A segunda abordagem consistiu em expandir a base de dados com apenas uma característica. Tal característica consiste no somatório dos valores das medidas de regra de associação de cada padrão interessante selecionado (conjunto de antecedentes de regra interessante).

Para ordenar os antecedentes mais interessantes, foram escolhidas quatro medidas de regras de associação tendo em vista os resultados obtidos com o experimento da modificação da função de ordenamento utilizando o algoritmo LRAR presentes na Seção 5.1. A partir dos resultados obtidos nesse experimento, foram escolhidas as medidas confiança, suporte da regra, diferença da confiança invertida e *lift* por serem consideradas mais promissoras devido aos ganhos consistentes de acurácia entre as coleções e com o aumento do tamanho da regra de associação. Após a expansão da base de dados foram

utilizados os algoritmos Rankboost e Randomforest [1] para confrontar se a base de dados expandida consegue beneficiar a acurácia dos algoritmos de ordenação.

Na Seção 5.2.1 encontram-se os resultados pertinentes a primeira abordagem de expansão enquanto na Seção 5.2.2 encontram-se os resultados pertinentes a segunda abordagem de expansão.

5.2.1 Resultados da expansão das características mais interessantes

Nessa abordagem da expansão das características mais interessantes são executados três tamanhos de expansão: uma expansão com as 10 características mais interessantes, outra com as 100 características mais interessantes e uma terceira com as 200 características mais interessantes. Como explicado em 4.2.1 as características são considerados mais importantes (são ordenados de forma decrescente) de acordo com uma medida de avaliação de regra de associação.

Expansão das 10 características mais interessantes

Nessa seção encontram-se os resultados da expansão da base de dados com as 10 características mais interessantes (ordenados de forma decrescente) de acordo com uma medida de avaliação de regra de associação. Primeiro são apresentados os resultados utilizando a confiança como medida para priorizar as regras de associação, depois são apresentados os resultados utilizando o suporte da regra, posteriormente os resultados utilizando a diferença da confiança invertida e por último os resultados utilizando o *lift*. Todos esses experimentos foram confrontados com a execução do algoritmo Rankboost e Randomforest sem a expansão das características.

As tabelas 5.11 e 5.12 apresentam os resultados das expansões das 10 características mais interessantes utilizando a confiança para priorizar as regras de associação nas coleções TD2003 e HP2003. A tabela 5.11 utiliza o algoritmo Randomforest para comparar os resultados da expansão, já a tabela 5.12 utiliza o algoritmo Rankboost para comparar os resultados da expansão.

[1] A implementação dos algoritmos Rankboost e Randomforest utilizadas no presente trabalho pode ser encontrada em: http://people.cs.umass.edu/~vdang/ranklib.html

Fold	Original | Randomforest		Confiança | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,0302	0,5572	-5,29%	0,60%
2	0,3516	0,7971	0,0191	0,5767	2,27%	4,46%
3	0,2518	0,7259	0,0238	0,6724	9,77%	0,65%
4	0,2658	0,7008	0,0716	0,2512	12,00%	-1,38%
5	0,2288	0,7397	0,0556	0,2912	-3,45%	0,10%
Média	**0,2618**	**0,7332**	**0,0400**	**0,4698**	**3,47%**	**0,97%**

Tabela 5.11: *Resultados para a expansão das características de acordo com a confiança utilizando o Randomforest*

Fold	Original | Rankboost		Confiança | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	-0,55%
2	0,2053	0,6582	0,2098	0,6583	2,20%	0,02%
3	0,3091	0,7380	0,3055	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2128	0,6174	0,21%	2,70%
Média	**0,2189**	**0,6576**	**0,2304**	**0,6659**	**5,25%**	**0,96%**

Tabela 5.12: *Resultados para a expansão das características de acordo com a confiança utilizando o Rankboost*

Conforme a tabela 5.11, percebe-se um ganho da acurácia ao se utilizar a abordagem de expansão com a confiança como medida de ordenação das regras de associação para o algoritmo Randomforest. Um aumento de ganho também pode ser observado na tabela 5.12, que utiliza o algoritmo Rankboost.

Nas tabelas 5.13 e 5.14 são apresentados os resultados das expansões das 10 características mais interessantes utilizando o suporte da regra para priorizar as regras de associação nas coleções TD2003 e HP2003. Os resultados presentes na tabela 5.13 utilizam o algoritmo Randomforest para confrontar os resultados da expansão, já os resultados presentes na tabela 5.14 utilizam o algoritmo Rankboost para comparar os resultados da expansão.

Fold	Original | Randomforest		Suporte da Regra | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2047	0,7084	-3,01%	0,81%
2	0,3516	0,7971	0,3575	0,8456	1,67%	6,09%
3	0,2518	0,7259	0,3267	0,7270	29,76%	0,15%
4	0,2658	0,7008	0,3114	0,6689	17,16%	-4,55%
5	0,2288	0,7397	0,2080	0,7391	-9,07%	-0,08%
Média	**0,2618**	**0,7332**	**0,2817**	**0,7378**	**7,59%**	**0,62%**

Tabela 5.13: *Resultados para a expansão das características de acordo com o suporte da regra utilizando o Randomforest*

Fold	Original | Rankboost		Suporte da Regra | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	-0,55%
2	0,2053	0,6582	0,2198	0,6583	2,20%	0,02%
3	0,3091	0,7380	0,3155	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2492	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2338	0,6174	0,21%	2,70%
Média	**0,2189**	**0,6576**	**0,2315**	**0,6659**	**5,25%**	**0,96%**

Tabela 5.14: *Resultados para a expansão das características de acordo com o suporte da regra utilizando o Rankboost*

De acordo com os resultados obtidos na utilização do algoritmo Randomforest, presente na tabela 5.13 percebe-se um ganho superior àquele obtido ao se utilizar a confiança como medida para ordenar as características mais interessantes. Porém, de acordo com a tabela 5.14 a utilização do algoritmo Rankboost não produziu ganhos melhores dos obtidos utilizando a confiança como medida para ordenar as características mais interessantes.

A próxima abordagem utiliza a diferença da confiança invertida como medida de regra de associação para ordenar as características mais interessantes. As tabelas 5.15 e 5.16 apresentam os resultados dessa abordagem nas coleções TD2003 e HP2003. A tabela 5.15 compara os resultados obtidos da expansão utilizando o algoritmo Randomforest com os resultados sem expansão utilizando o mesmo algoritmo. A tabela 5.16 confronta os resultados obtidos com a expansão da base de dados com os resultados sem expansão para o algoritmo Rankboost.

Fold	Original | Randomforest		Dif. conf. inv. | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2036	0,6972	-3,50%	-0,79%
2	0,3516	0,7971	0,3238	0,8138	-7,91%	2,09%
3	0,2518	0,7259	0,2203	0,7232	-12,52%	-0,37%
4	0,2658	0,7008	0,2422	0,6685	-8,86%	-4,61%
5	0,2288	0,7397	0,2464	0,7557	7,71%	2,16%
Média	**0,2618**	**0,7332**	**0,2473**	**0,7317**	**-5,55%**	**-0,22%**

Tabela 5.15: *Resultados para a expansão das características de acordo com a diferença da confiança invertida utilizando o Randomforest*

Fold	Original | Rankboost		Dif. conf. inv. | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1326	0,6198	-4,22%	-3,78%
2	0,2053	0,6582	0,1849	0,6980	-9,92%	6,06%
3	0,3091	0,7380	0,3636	0,8129	17,64%	10,16%
4	0,2294	0,6561	0,2936	0,5831	27,99%	-11,12%
5	0,2124	0,5915	0,1493	0,6077	-29,69%	2,76%
Média	**0,2189**	**0,6576**	**0,2248**	**0,6643**	**2,69%**	**1,03%**

Tabela 5.16: *Resultados para a expansão das características de acordo com a diferença da confiança invertida utilizando o Rankboost*

Através dos resultados presentes na tabela 5.15 verifica-se que a abordagem utilizando a diferença da confiança invertida obteve uma perda de ganho ao se utilizar o algoritmo Randomforest. Porém, através dos resultados presentes na tabela 5.16 percebe-se um ganho de acurácia ao se utilizar a expansão das características de acordo com a diferença da confiança invertida utilizando o algoritmo Rankboost. Vale ressaltar que o ganho obtido não foi superior aos ganhos utilizando-se a confiança e o suporte da regra.

As tabelas 5.17 e 5.18 abaixo apresentam os resultados das expansões das 10 características mais interessantes utilizando *lift* para priorizar as regras de associação nas coleções TD2003 e HP2003. A tabela 5.17 utiliza o algoritmo Randomforest para comparar os resultados da expansão, já a tabela 5.18 utiliza o algoritmo Rankboost para comparar os resultados da expansão.

Fold	Original | Randomforest		*lift* | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2093	0,6406	-0,82%	-8,84%
2	0,3516	0,7971	0,2769	0,7821	-21,24%	-1,88%
3	0,2518	0,7259	0,2743	0,7138	8,96%	-1,66%
4	0,2658	0,7008	0,2979	0,6781	12,08%	-3,24%
5	0,2288	0,7397	0,2005	0,7800	-12,38%	5,45%
Média	**0,2618**	**0,7332**	**0,2518**	**0,7189**	**-3,82%**	**-1,95%**

Tabela 5.17: *Resultados para a expansão das características de acordo com lift utilizando o Randomforest*

Fold	Original | Rankboost		*lift* | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6389	0,80%	-0,82%
2	0,2053	0,6582	0,2098	0,6598	2,20%	0,25%
3	0,3091	0,7380	0,3055	0,7251	-1,15%	-1,75%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2134	0,6074	0,50%	2,70%
Média	**0,2189**	**0,6576**	**0,2305**	**0,6612**	**5,30%**	**0,56%**

Tabela 5.18: *Resultados para a expansão das características de acordo com lift utilizando o Rankboost*

Conforme a tabela 5.17 percebe-se que, assim como na utilização da diferença da confiança invertida como medida de ordenação de regra de associação, na utilização do *lift* também houve um declínio do ganho para o algoritmo Randomforest. Em contrapartida, conforme a tabela 5.18, utilizando-se o algoritmo Rankboost a abordagem com expansão das características mais interessantes adquiriu um aumento no ganho superior às abordagens anteriormente expostas.

As tabelas 5.29 e 5.30 a seguir demonstram os resultados obtidos com a ordenação utilizando cada medida de regra de associação. A tabela 5.29 remete à aplicação da expansão da base de dados ao algoritmo Randomforest, enquanto a tabela 5.30 à aplicação do algoritmo Rankboost.

Medida	Original | RF		Exp. 10 | RF		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2618	0,7332	0,2709	0,7403	**3,47%**	**0,97%**
Sup. abs. da regra	0,2618	0,7332	0,2817	0,7378	**7,59%**	**0,62%**
Dif. conf. inv.	0,2618	0,7332	0,2473	0,7317	-5,55%	-0,22%
Lift	0,2618	0,7332	0,2518	0,7189	-3,82%	-1,95%

Tabela 5.19: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Randomforest*

Medida	Original | RB		Exp. 10 | RB		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2189	0,6576	0,2304	0,6639	**5,25%**	**0,96%**
Sup. abs. da regra	0,2189	0,6576	0,2234	0,6639	2,04%	**0,96%**
Dif. conf. inv.	0,2189	0,6576	0,2248	0,6644	2,69%	**1,03%**
Lift	0,2189	0,6576	0,2306	0,6613	**5,30%**	0,56%

Tabela 5.20: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Rankboost*

De acordo com os resultados expostos para a abordagem de expansão das 10 características mais interessantes, os melhores resultados foram todos obtidos utilizando o algoritmo Rankboost, independente da medida utilizada, porém os melhores resultados foram obtidos com a utilização das medidas "confiança" e "*lift*" para a base de dados TD2003 e "diferença da confiança invertida", "confiança" e "suporte absoluto da regra" para a base de dados HP2003. Na utilização do algoritmo Randomforest, apenas as medidas "confiança" e "suporte absoluto da regra" obtiveram um ganho positivo.

Expansão das 100 características mais interessantes

Nessa seção encontram-se os resultados da expansão da base de dados com as 100 características mais interessantes, ordenados de forma decrescente de acordo com uma medida de avaliação de regra de associação. Assim como na seção anterior, primeiro são apresentados os resultados utilizando a confiança como medida para priorizar as regras de associação, depois são apresentados os resultados utilizando o suporte da regra, posteriormente os resultados utilizando a diferença da confiança invertida e por último os resultados utilizando o *lift*. Todos esses experimentos foram confrontados com a execução do algoritmo Randomforest e Rankboost sem a expansão das características.

Os primeiros resultados são da abordagem que utiliza a confiança como medida para priorizar as regras de associação. As tabelas 5.21 e 5.22 apresentam os resultados dessa abordagem para a expansão da base de dados utilizando as 100 características mais interessantes. A tabela 5.21 exibe os resultados dessa abordagem utilizando o algoritmo Randomforest, enquanto a tabela 5.22 exibe os resultados utilizando o algoritmo Rankboost.

Fold	Original | Randomforest		Confiança | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2019	0,7070	-4,32%	0,60%
2	0,3516	0,7971	0,3336	0,8178	-5,10%	2,60%
3	0,2518	0,7259	0,2658	0,7291	5,56%	0,44%
4	0,2658	0,7008	0,2885	0,6734	8,55%	-3,91%
5	0,2288	0,7397	0,2308	0,7468	0,87%	0,96%
Média	**0,2618**	**0,7332**	**0,2641**	**0,7348**	**0,89%**	**0,21%**

Tabela 5.21: *Resultados para a expansão de 100 características de acordo com a confiança utilizando o Randomforest*

Fold	Original | Rankboost		Confiança | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	-0,55%
2	0,2053	0,6582	0,2098	0,6583	2,20%	0,02%
3	0,3091	0,7380	0,3055	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2128	0,6074	0,21%	2,70%
Média	**0,2189**	**0,6576**	**0,2304**	**0,6639**	**5,25%**	**0,96%**

Tabela 5.22: *Resultados para a expansão de 100 características de acordo com a confiança utilizando o Rankboost*

De acordo com os resultados obtidos na utilização do algoritmo Randomforest, presente na tabela 5.21, observa-se um aumento de ganho ao se expandir a base de dados com 100 características. Porém na expansão com 100 características houve um declínio do ganho quando comparado com a abordagem anterior que expandia apenas 10 características.

Assim como nos resultados da expansão com 10 características, essa abordagem também conseguiu um aumento do ganho ao se utilizar o algoritmo Rankboost. Porém, quando confrontado com a abordagem que expandia apenas 10 características não percebe-se aumento do ganho.

A próxima abordagem utilizou o suporte da regra como medida de regra de associação para ordenar as características mais interessantes. As tabelas 5.23 e 5.24

apresentam os resultados dessa abordagem nas coleções TD2003 e HP2003. A tabela 5.23 compara os resultados obtidos da expansão utilizando o algoritmo Randomforest com os resultados sem expansão utilizando o mesmo algoritmo. A tabela 5.24 compara os resultados obtidos da expansão da base de dados com os resultados sem expansão para o algoritmo Rankboost.

Fold	Original | Randomforest		Suporte da Regra | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2043	0,7087	-3,17%	0,85%
2	0,3516	0,7971	0,3562	0,8405	1,33%	5,45%
3	0,2518	0,7259	0,3023	0,7155	20,07%	-1,43%
4	0,2658	0,7008	0,3064	0,6721	15,27%	-4,09%
5	0,2288	0,7397	0,2354	0,7502	2,91%	1,42%
Média	**0,2618**	**0,7332**	**0,2809**	**0,7374**	**7,32%**	**0,57%**

Tabela 5.23: *Resultados para a expansão de 100 características de acordo com o suporte da regra utilizando o Randomforest*

Fold	Original | Rankboost		Suporte da Regra | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	-0,55%
2	0,2053	0,6582	0,2091	0,6583	1,89%	0,02%
3	0,3091	0,7380	0,3057	0,7382	-1,07%	0,16%
4	0,2294	0,6561	0,2696	0,6750	17,54%	2,88%
5	0,2124	0,5915	0,2128	0,6074	0,21%	-2,25%
Média	**0,2189**	**0,6576**	**0,2274**	**0,6639**	**3,87%**	**0,10%**

Tabela 5.24: *Resultados para a expansão de 100 características de acordo com o suporte da regra utilizando o Rankboost*

De acordo com os resultados obtidos na utilização do algoritmo Randomforest, presente na tabela 5.23, houve um aumento do ganho ao se expandir as bases de dados TD2003 e HP2003 utilizando o suporte absoluto da regra como medida de ordenação. O aumento de ganho obtido nessa abordagem não foi superior ao obtido na abordagem com a expansão de apenas 10 características, porém a diferença entre as abordagens foi muito sensível.

Assim como nos resultados da expansão de 10 características utilizando o suporte da regra, essa abordagem também conseguiu um aumento no ganho ao se utilizar o algoritmo Rankboost. Quando comparada com a abordagem anterior, essa abordagem conseguiu ganhos melhores tanto para a base de dados TD2003 quanto para a base de dados HP2003.

Nas tabelas 5.25 e 5.26 são apresentados os resultados das expansões utilizando a diferença da confiança invertida para priorizar as regras de associação. Os resultados presentes na tabela 5.25 utilizam o algoritmo Randomforest para confrontar os resultados da expansão, já os resultados presentes na tabela 5.26 utilizam o algoritmo Rankboost.

Fold	Original | Randomforest		Dif. conf. inv. | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1824	0,6852	-13,56%	-2,49%
2	0,3516	0,7971	0,3153	0,7920	-10,31%	-0,63%
3	0,2518	0,7259	0,2218	0,7349	-11,89%	1,25%
4	0,2658	0,7008	0,2693	0,6822	1,33%	-2,66%
5	0,2288	0,7397	0,2481	0,7381	8,45%	-0,22%
Média	**0,2618**	**0,7332**	**0,2474**	**0,7265**	**-5,50%**	**-0,92%**

Tabela 5.25: *Resultados para a expansão de 100 características de acordo com a diferença da confiança invertida utilizando o Randomforest*

Fold	Original | Rankboost		Dif. conf. inv. | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1541	0,6198	11,29%	3,20%
2	0,2053	0,6582	0,2139	0,6980	4,20%	0,56%
3	0,3091	0,7380	0,2966	0,8129	-4,04%	-12,58%
4	0,2294	0,6561	0,2887	0,5831	25,88%	-8,86%
5	0,2124	0,5915	0,2229	0,6077	4,95%	5,87%
Média	**0,2189**	**0,6576**	**0,2352**	**0,6643**	**7,46%**	**-2,80%**

Tabela 5.26: *Resultados para a expansão de 100 características de acordo com a diferença da confiança invertida utilizando o Rankboost*

Conforme a tabela 5.25 percebe-se que, assim como na expansão de 10 características mais interessantes utilizando a mesma medida, houve um declínio do ganho em ambas as bases de dados na utilização do algoritmo Randomforest. Porém nessa abordagem o ganho foi sensivelmente superior à abordagem com a expansão de apenas 10 características.

Na tabela 5.26, observa-se que, ao se utilizar o algoritmo Rankboost, essa abordagem obteve um aumento no ganho para a base TD2003, porém um declínio para a base HP2003. Quando confrontados os resultados dessa abordagem com a abordagem anterior, percebe-se que o ganho da base TD2003 foi superior na presente abordagem.

Os últimos resultados da expansão com 100 características mais interessantes utiliza a medida *lift* para ordenar as regras de associação. As tabelas 5.27 e 5.28 respec-

tivamente confrontam os resultados dessa abordagem com os algoritmos Randomforest e Rankboost sem a expansão.

Fold	Original | Randomforest		*lift* | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2035	0,6468	-3,59%	-7,95%
2	0,3516	0,7971	0,2382	0,7363	-32,24%	-7,62%
3	0,2518	0,7259	0,2456	0,7407	-2,47%	2,05%
4	0,2658	0,7008	0,2978	0,7121	12,03%	1,61%
5	0,2288	0,7397	0,1923	0,6850	-15,94%	-7,40%
Média	**0,2618**	**0,7332**	**0,2355**	**0,7042**	**-10,06%**	**-3,96%**

Tabela 5.27: *Resultados para a expansão de 100 características de acordo com lift utilizando o Randomforest*

Fold	Original | Rankboost		*lift* | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1389	0,6389	0,32%	-0,91%
2	0,2053	0,6582	0,2000	0,6598	-2,56%	-0,17%
3	0,3091	0,7380	0,3152	0,7251	1,98%	-1,34%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2134	0,6074	0,50%	2,70%
Média	**0,2189**	**0,6576**	**0,2303**	**0,6612**	**5,23%**	**0,55%**

Tabela 5.28: *Resultados para a expansão de 100 características de acordo com lift utilizando o Rankboost*

Conforme a tabela 5.27 percebe-se que, da mesma forma como na expansão de 10 características utilizando *lift*, houve uma perda do ganho nessa abordagem para o algoritmo Randomforest. Comparando-se com a abordagem anterior, essa abordagem teve um considerável declínio no ganho.

Em contrapartida, conforme a tabela 5.28, utilizando-se o algoritmo Rankboost a abordagem com expansão das 100 características mais interessantes obteve um aumento no ganho em ambas bases de dados.

As tabelas 5.29 e 5.30 a seguir demonstram os resultados obtidos com a ordenação utilizando cada medida de regra de associação. A tabela 5.29 remete à aplicação da expansão da base de dados ao algoritmo Randomforest para a base de dados expandida com as 100 características mais interessantes, enquanto a tabela 5.30 remete à aplicação do algoritmo Rankboost para as mesmas bases de dados.

Medida	Original | RF		Exp. 10 | RF		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2618	0,7332	0,2641	0,7348	**0,89%**	**0,21%**
Sup. abs. da regra	0,2618	0,7332	0,2809	0,7374	**7,32%**	**0,57%**
Dif. das conf. inv.	0,2618	0,7332	0,2474	0,7265	-5,50%	-0,92%
Lift	0,2618	0,7332	0,2355	0,7042	-10,06%	-3,96%

Tabela 5.29: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Randomforest*

Medida	Original | RB		Exp. 10 | RB		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2189	0,6576	0,2304	0,6639	**5,25%**	**0,96%**
Sup. abs. da regra	0,2189	0,6576	0,2274	0,6583	3,87%	0,10%
Dif. conf. inv.	0,2189	0,6576	0,2353	0,6392	**7,46%**	-2,80%
Lift	0,2189	0,6576	0,2304	0,6612	5,23%	**0,55%**

Tabela 5.30: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Rankboost*

De acordo com os resultados expostos para a abordagem de expansão das 100 características mais interessantes, os melhores resultados, assim como na abordagem anterior, foram majoritariamente obtidos utilizando o algoritmo Rankboost. Na utilização do algoritmo Randomforest, apenas a utilização das medidas "confiança" e "suporte absoluto da regra" obtiveram ganho positivo. Comparando-se os resultados obtidos na expansão com 10 características e a com 100 características, percebe-se o declínio do ganho na utilização do algoritmo Randomforest, enquanto para o algoritmo Rankboost houve um aumento do ganho.

Expansão das 200 características mais interessantes

Nessa seção encontram-se os resultados da expansão da base de dados com as 200 características mais interessantes, ordenados de forma decrescente de acordo com uma medida de avaliação de regra de associação. Assim como na seção anterior, primeiro são apresentados os resultados utilizando a confiança, depois os resultados que utilizam o suporte da regra, posteriormente aqueles com a diferença da confiança invertida e por último os resultados utilizando o *lift*. Todos esses experimentos foram confrontados com a execução do algoritmo Randomforest e Rankboost sem a expansão das características.

Os primeiros resultados são da abordagem que utiliza a confiança como medida para priorizar as regras de associação. As tabelas 5.31 e 5.32 apresentam os resultados dessa abordagem para a expansão da base de dados com as 200 características mais interessantes utilizando, respectivamente, os algoritmos Randomforest e Rankboost.

Fold	Original | Randomforest		Confiança | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1899	0,6983	-10,03%	-0,63%
2	0,3516	0,7971	0,3079	0,8392	-12,41%	5,29%
3	0,2518	0,7259	0,2766	0,7280	9,85%	0,29%
4	0,2658	0,7008	0,2488	0,6805	-6,39%	-2,90%
5	0,2288	0,7397	0,2298	0,7286	0,46%	-1,50%
Média	**0,2618**	**0,7332**	**0,2506**	**0,7349**	**-4,27%**	**0,23%**

Tabela 5.31: *Resultados para a expansão de 200 características de acordo com a confiança utilizando o Randomforest*

Fold	Original | Rankboost		Confiança | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	-0,55%
2	0,2053	0,6582	0,2098	0,6583	2,20%	0,02%
3	0,3091	0,7380	0,3055	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2128	0,6074	0,21%	2,70%
Média	**0,2189**	**0,6576**	**0,2304**	**0,6639**	**5,25%**	**0,96%**

Tabela 5.32: *Resultados para a expansão de 200 características de acordo com a confiança utilizando o Rankboost*

De acordo com os resultados obtidos na utilização do algoritmo Randomforest presente na tabela 5.31, houve um ganho negativo para a base de dados TD2003, porém obteve-se um aumento do ganho para a base de dados HP203. Comparando-se os resultados obtidos nessa abordagem com as anteriores, percebe-se grande declínio do ganho com o aumento das características expandidas nas bases de dados.

Assim como nos resultados das outras abordagens de expansão, essa também conseguiu um aumento no ganho ao se utilizar o algoritmo Rankboost. Tais resultados de ganho obtidos foram semelhantes àqueles com a expansão de 100 características.

A próxima abordagem utilizou o suporte da regra como medida de regra de associação para ordenar as características mais interessantes. As tabelas 5.33 e 5.34 apresentam os resultados dessa abordagem para as coleções TD2003 e HP2003. A tabela 5.33 compara os resultados obtidos da expansão utilizando o algoritmo Randomforest com os resultados sem expansão utilizando o mesmo algoritmo. A tabela 5.34 compara

os resultados obtidos da expansão da base de dados com os resultados sem expansão para o algoritmo Rankboost.

Fold	Original | Randomforest		Suporte da Regra | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2066	0,7128	-2,10%	1,44%
2	0,3516	0,7971	0,3634	0,8618	3,37%	8,12%
3	0,2518	0,7259	0,2315	0,7445	-8,05%	2,56%
4	0,2658	0,7008	0,2625	0,6753	-1,23%	-3,63%
5	0,2288	0,7397	0,1971	0,7597	-13,84%	2,70%
Média	**0,2618**	**0,7332**	**0,2522**	**0,7508**	**-3,65%**	**2,40%**

Tabela 5.33: *Resultados para a expansão de 200 características de acordo com o suporte da regra utilizando o Randomforest*

Fold	Original | Rankboost		Suporte da Regra | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6406	0,80%	1,76%
2	0,2053	0,6582	0,2101	0,6583	2,34%	0,28%
3	0,3091	0,7380	0,3057	0,7382	-1,07%	0,16%
4	0,2294	0,6561	0,2666	0,6750	16,23%	2,88%
5	0,2124	0,5915	0,2128	0,6074	0,21%	-2,58%
Média	**0,2189**	**0,6576**	**0,2269**	**0,6639**	**3,68%**	**0,55%**

Tabela 5.34: *Resultados para a expansão de 200 características de acordo com o suporte da regra utilizando o Rankboost*

De acordo com os resultados obtidos na utilização do algoritmo Randomforest presente na tabela 5.33, houve um ganho negativo para a base de dados TD2003, mas obteve-se um aumento do ganho para a base de dados HP203. Assim como no resultado com a confiança como medida para ordenar as regras de associação, percebe-se grande declínio do ganho com o aumento das características expandidas nas bases de dados.

Assim como nos resultados das outras abordagens de expansão utilizando o suporte da regra como medida de ordenação, essa também conseguiu um aumento no ganho ao se utilizar o algoritmo Rankboost para ambas bases de dados. Tais resultados de ganho obtidos foram semelhantes àqueles com a expansão de 100 características.

Nas tabelas 5.35 e 5.36 são apresentados os resultados das expansões utilizando a diferença da confiança invertida para priorizar as regras de associação. Os resultados presentes na tabela 5.35 utilizam o algoritmo Randomforest para confrontar os resultados da expansão, já os resultados presentes na tabela 5.36 utilizam o algoritmo Rankboost.

Fold	Original | Randomforest		Dif. conf. inv. | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1906	0,6882	-9,70%	-2,07%
2	0,3516	0,7971	0,3081	0,7908	-12,36%	-0,79%
3	0,2518	0,7259	0,2837	0,7370	12,68%	1,53%
4	0,2658	0,7008	0,2722	0,7122	2,40%	1,63%
5	0,2288	0,7397	0,2372	0,7378	3,67%	-0,27%
Média	**0,2618**	**0,7332**	**0,2583**	**0,7332**	**-1,32%**	**-0,01%**

Tabela 5.35: *Resultados para a expansão de 200 características de acordo com a diferença da confiança invertida utilizando o Randomforest*

Fold	Original | Rankboost		Dif. conf. inv. | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1490	0,6198	7,59%	3,20%
2	0,2053	0,6582	0,2180	0,6980	6,23%	-6,60%
3	0,3091	0,7380	0,3021	0,8129	-2,25%	-12,58%
4	0,2294	0,6561	0,2849	0,5831	24,20%	-13,69%
5	0,2124	0,5915	0,2039	0,6077	-3,97%	2,99%
Média	**0,2189**	**0,6576**	**0,2316**	**0,6643**	**5,79%**	**-5,71%**

Tabela 5.36: *Resultados para a expansão de 200 características de acordo com a diferença da confiança invertida utilizando o Rankboost*

Conforme a tabela 5.35 percebe-se que, assim como na expansões anteriores utilizando a mesma medida, houve um declínio da acurácia em ambas as bases de dados utilizando-se do algoritmo Randomforest. Porém, quando confrontado com as abordagens anteriores, a atual abordagem obteve menores perdas de MAP.

De acordo com a tabela 5.36, percebe-se que, ao se utilizar o algoritmo Rankboost, essa abordagem adquiriu um aumento de MAP para a coleção TD2003, porém um declínio para a coleção HP2003. Quando confrontado o resultado dessa abordagem com as abordagens anteriores (expansão de 10 características e 100 características) pode-se perceber um declínio do ganho.

Os últimos resultados utilizam a medida *lift* para ordenar as regras de associação. As tabelas 5.37 e 5.38 respectivamente confrontam os resultados dessa abordagem com os algoritmos Randomforest e Rankboost sem a expansão.

Fold	Original | Randomforest		*lift* | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1806	0,6245	-14,40%	-11,13%
2	0,3516	0,7971	0,2207	0,7742	-37,22%	-2,87%
3	0,2518	0,7259	0,2536	0,7173	0,71%	-1,18%
4	0,2658	0,7008	0,2811	0,7111	5,77%	1,47%
5	0,2288	0,7397	0,1927	0,7153	-15,77%	-3,30%
Média	**0,2618**	**0,7332**	**0,2258**	**0,7085**	**-13,76%**	**-3,38%**

Tabela 5.37: *Resultados para a expansão de 200 características de acordo com lift utilizando o Randomforest*

Fold	Original | Rankboost		*lift* | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1356	0,6389	-2,03%	-0,91%
2	0,2053	0,6582	0,2000	0,6598	-2,56%	-0,17%
3	0,3091	0,7380	0,3040	0,7251	-1,64%	-2,14%
4	0,2294	0,6561	0,2843	0,6750	23,95%	2,88%
5	0,2124	0,5915	0,2141	0,6074	0,80%	2,70%
Média	**0,2189**	**0,6576**	**0,2276**	**0,6612**	**3,97%**	**0,37%**

Tabela 5.38: *Resultados para a expansão de 200 características de acordo com lift utilizando o Rankboost*

Conforme a tabela 5.37 percebe-se que, da mesma forma como nas expansões anteriores utilizando *lift*, houve um declínio do ganho dessa abordagem para o algoritmo Randomforest para ambas bases de dados. Quando comparados os resultados dessa abordagem com as anteriores pode-se perceber um declínio da acurácia.

Porém, conforme a tabela 5.38, utilizando-se o algoritmo Rankboost essa abordagem adquiriu um aumento de MAP para ambas bases de dados.

As tabelas 5.39 e 5.40 a seguir exibem os resultados obtidos com a ordenação utilizando cada medida de regra de associação. A tabela 5.39 remete à aplicação da expansão da base de dados ao algoritmo Randomforest, enquanto a tabela 5.40 à aplicação do algoritmo Rankboost.

Medida	Original | RF		Exp. 10 | RF		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2618	0,7332	0,2506	0,7349	-4,27%	**0,23%**
Sup. abs. da regra	0,2618	0,7332	0,2522	0,7508	**-3,65%**	**2,40%**
Dif. das conf. inv.	0,2618	0,7332	0,2583	0,7332	**-1,32%**	-0,01%
Lift	0,2618	0,7332	0,2258	0,7085	-13,76%	-3,38%

Tabela 5.39: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Randomforest*

Medida	Original | RB		Exp. 10 | RB		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2189	0,6576	0,2304	0,6639	**5,25%**	**0,96%**
Sup. abs. da regra	0,2189	0,6576	0,2270	0,6612	3,68%	**0,55%**
Dif. conf. inv.	0,2189	0,6576	0,2316	0,6200	**5,79%**	-5,71%
Lift	0,2189	0,6576	0,2276	0,6600	3,97%	0,37%

Tabela 5.40: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Rankboost*

De acordo com os resultados expostos para as abordagem de expansão com as 200 características mais interessantes, os melhores resultados, foram em sua maioria obtidos utilizando o algoritmo Rankboost. Na utilização do algoritmo Randomforest, apenas a utilização da medida "suporte absoluto da regra" na coleção HP2003 obteve melhoria. Comparando-se os resultados obtidos nas expansões com 10 características, com 100 características e com 200 características percebe-se o declínio de acurácia na utilização da expansão com 200 características.

5.2.2 Resultados da expansão com o somatório das características mais interessantes

Como explicado em 4.2.1, essa abordagem realiza a soma de todos os valores das características expandidas para a abordagem de expansão das cem características mais interessantes e realimenta as bases de dados de teste, treino e validação com apenas uma característica que compõe esse somatório. O objetivo desse experimento é de obter melhores resultados que a expansão das melhores características realizando a diminuição da quantidade de valores iguais a zero das bases de dados.

Assim como nas seções anteriores, são apresentados primeiro os resultados utilizando a confiança como medida de ordenação das regras de associação, depois os

resultados que utilizam o suporte da regra, posteriormente aqueles com a diferença da confiança invertida e por último os resultados utilizando o *lift*. Todos esses experimentos foram confrontados com a execução do algoritmo Randomforest e Rankboost sem a expansão das características.

As tabelas 5.41 e 5.42 apresentam os resultados das expansões do somatório utilizando a confiança para priorizar as regras de associação nas coleções TD2003 e HP2003. A tabela 5.41 utiliza o algoritmo Randomforest para comparar os resultados da expansão, já a tabela 5.42 utiliza o algoritmo Rankboost para comparar os resultados da expansão.

Fold	Original | Randomforest		Confiança | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1411	0,7031	-33,14%	0,06%
2	0,3516	0,7971	0,2888	0,7830	-17,86%	-1,77%
3	0,2518	0,7259	0,3322	0,8001	31,93%	10,23%
4	0,2658	0,7008	0,2175	0,7129	-18,18%	1,73%
5	0,2288	0,7397	0,1902	0,7147	-16,86%	-3,38%
Média	**0,2618**	**0,7332**	**0,2339**	**0,7428**	**-10,64%**	**1,30%**

Tabela 5.41: *Resultados para a expansão do somatório de características de acordo com a confiança utilizando o Randomforest*

Fold	Original | Rankboost		Confiança | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1396	0,6555	0,80%	1,76%
2	0,2053	0,6582	0,2460	0,6583	19,83%	0,02%
3	0,3091	0,7380	0,3055	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2843	0,6584	23,95%	0,36%
5	0,2124	0,5915	0,1753	0,5935	-17,42%	0,34%
Média	**0,2189**	**0,6576**	**0,2301**	**0,6608**	**5,13%**	**0,49%**

Tabela 5.42: *Resultados para a expansão do somatório de características de acordo com a confiança utilizando o Rankboost*

De acordo com os resultados obtidos com a utilização do algoritmo Randomforest presente na tabela 5.41, houve um declínio da acurácia ao se utilizar tal algoritmo para a coleção TD2003, porém para a coleção HP2003 conseguiu-se um aumento no ganho.

Nessa abordagem, conseguiu-se um aumento de MAP ao se utilizar o algoritmo Rankboost para ambas coleções. Conforme a tabela 5.42 houve um ganho de 5,13% para a coleção TD2003 e de 0,49% para a coleção HP2003. Porém comparando-se com as abordagens anteriores o ganho não melhorou.

A próxima abordagem utilizou o suporte da regra como medida de regra de associação para ordenar as características mais interessantes. As tabelas 5.43 e 5.44 apresentam os resultados dessa abordagem para as coleções TD2003 e HP2003. A tabela 5.43 compara os resultados obtidos da expansão utilizando o algoritmo Randomforest com os resultados sem expansão utilizando o mesmo algoritmo. A tabela 5.44 compara os resultados obtidos da expansão da base de dados com os resultados sem expansão para o algoritmo Rankboost.

Fold	Original | Randomforest		Suporte da Regra | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1388	0,6661	-34,21%	-5,21%
2	0,3516	0,7971	0,2131	0,5874	-39,39%	-26,30%
3	0,2518	0,7259	0,2691	0,6729	6,87%	-7,29%
4	0,2658	0,7008	0,1601	0,6943	-39,78%	-0,93%
5	0,2288	0,7397	0,1247	0,6373	-45,49%	-13,85%
Média	**0,2618**	**0,7332**	**0,1812**	**0,6516**	**-30,80%**	**-11,13%**

Tabela 5.43: *Resultados para a expansão do somatório de características de acordo com o suporte da regra utilizando o Randomforest*

Fold	Original | Rankboost		Suporte da Regra | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1437	0,6555	1,35%	1,76%
2	0,2053	0,6582	0,2460	0,6683	19,83%	0,02%
3	0,3091	0,7380	0,3055	0,7382	-1,15%	0,03%
4	0,2294	0,6561	0,2843	0,6584	23,95%	0,36%
5	0,2124	0,5915	0,1753	0,6118	-17,42%	1,75%
Média	**0,2189**	**0,6576**	**0,2303**	**0,6664**	**5,20%**	**0,74%**

Tabela 5.44: *Resultados para a expansão do somatório de características de acordo com o suporte da regra utilizando o Rankboost*

Diferentemente das outras abordagens de expansão, percebe-se, de acordo com os resultados obtidos na utilização do algoritmo Randomforest presente na tabela 5.43, um declínio da acurácia ao se realizar a expansão com o somatório das características mais interessantes.

Essa abordagem conseguiu um aumento no ganho ao se utilizar o algoritmo Rankboost. Conforme a tabela 5.34 percebe-se um ganho de 5,20% para a coleção TD2003 e de 0,74% para a coleção HP2003. Comparando-se com os resultados das abordagens anteriores houve um aumento da acurácia na abordagem atual.

Nas tabelas 5.45 e 5.46 são apresentados os resultados das expansões utilizando a diferença da confiança invertida para priorizar as regras de associação. Os resultados presentes na tabela 5.45 utilizam o algoritmo Randomforest para confrontar os resultados da expansão, já os resultados presentes na tabela 5.46 utilizam o algoritmo Rankboost.

Fold	Original | Randomforest		Dif. conf. inv. | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,1948	0,6909	-7,68%	-1,68%
2	0,3516	0,7971	0,3515	0,7587	-0,03%	-4,81%
3	0,2518	0,7259	0,2639	0,7755	4,83%	6,84%
4	0,2658	0,7008	0,2945	0,7016	10,81%	0,11%
5	0,2288	0,7397	0,2316	0,7411	1,22%	0,18%
Média	**0,2618**	**0,7332**	**0,2673**	**0,7336**	**2,09%**	**0,04%**

Tabela 5.45: *Resultados para a expansão do somatório de características de acordo com a diferença da confiança invertida utilizando o Randomforest*

Fold	Original | Rankboost		Dif. conf. inv. | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1381	0,6286	-0,28%	-2,42%
2	0,2053	0,6582	0,2046	0,6463	-0,30%	-1,80%
3	0,3091	0,7380	0,2466	0,7483	-20,22%	1,40%
4	0,2294	0,6561	0,1456	0,6953	-36,53%	5,98%
5	0,2124	0,5915	0,1850	0,6268	-12,89%	5,97%
Média	**0,2189**	**0,6576**	**0,1840**	**0,6690**	**-15,96%**	**1,75%**

Tabela 5.46: *Resultados para a expansão do somatório de características de acordo com a diferença da confiança invertida utilizando o Rankboost*

Conforme a tabela 5.45 observa-se que houve um aumento da acurácia em ambas as bases de dados ao se utilizar o algoritmo Randomforest. Quando confrontado com os resultados das abordagens anteriores, o resultado da abordagem atual obteve aumento no ganho.

Na tabela 5.46, percebe-se que ao se utilizar o algoritmo Rankboost essa abordagem obteve declínio no ganho para a coleção TD2003, porém um aumento para a coleção HP2003.

Os últimos resultados dessa abordagem utilizam a medida *lift* para ordenar as regras de associação. As tabelas 5.47 e 5.48 respectivamente confrontam os resultados dessa abordagem com os algoritmos Randomforest e Rankboost sem a expansão.

Fold	Original | Randomforest		*lift* | Randomforest		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,2110	0,7027	0,2104	0,6071	-0,28%	-13,61%
2	0,3516	0,7971	0,3336	0,8067	-5,11%	1,21%
3	0,2518	0,7259	0,3071	0,7284	21,98%	0,35%
4	0,2658	0,7008	0,2836	0,6917	6,69%	-1,30%
5	0,2288	0,7397	0,1870	0,6938	-18,27%	-6,21%
Média	**0,2618**	**0,7332**	**0,2643**	**0,7055**	**0,97%**	**-3,78%**

Tabela 5.47: *Resultados para a expansão do somatório de características de acordo com lift utilizando o Randomforest*

Fold	Original | Rankboost		*lift* | Rankboost		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
1	0,1385	0,6442	0,1506	0,6547	8,74%	1,64%
2	0,2053	0,6582	0,2014	0,6662	-1,88%	1,23%
3	0,3091	0,7380	0,3075	0,7262	-0,51%	-1,59%
4	0,2294	0,6561	0,2260	0,6584	-1,44%	0,36%
5	0,2124	0,5915	0,2263	0,6135	6,59%	0,34%
Média	**0,2189**	**0,6576**	**0,2224**	**0,6638**	**1,58%**	**0,34%**

Tabela 5.48: *Resultados para a expansão do somatório de características de acordo com lift utilizando o Rankboost*

Conforme a tabela 5.47 percebe-se que houve um aumento do ganho nessa abordagem para a base de dados TD2003 utilizando o algoritmo Randomforest. Para a coleção HP2003 percebe-se um declínio da acurácia. Confrontando os resultados obtidos nessa abordagem com as abordagens anteriores pode-se perceber uma aumento do ganho, porém sem ganhos positivos para a base HP2003.

Utilizando o algoritmo Rankboost, percebe-se na tabela 5.48, um aumento na acurácia em ambas bases de dados. Quando comparados os resultado dessa abordagem com as anteriores percebe-se um declínio do ganho, porém contendo ainda ganhos positivos.

As tabelas 5.49 e 5.50 a seguir exibem os resultados obtidos com a ordenação utilizando cada medida de regra de associação. A tabela 5.49 remete à aplicação da expansão da base de dados ao algoritmo Randomforest, enquanto a tabela 5.50 à aplicação do algoritmo Rankboost.

Medida	Original | RF		Exp. 10 | RF		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2618	0,7332	0,2339	0,7428	-10,64%	**1,30%**
Sup. abs. da regra	0,2618	0,7332	0,1812	0,6516	-30,80%	-11,13%
Dif. das conf. inv.	0,2618	0,7332	0,2673	0,7336	**2,09%**	**0,04%**
Lift	0,2618	0,7332	0,2643	0,7055	**0,97%**	-3,78%

Tabela 5.49: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Randomforest*

Medida	Original | RB		Exp. 10 | RB		Ganho	
	TD2003	HP2003	TD2003	HP2003	TD2003	HP2003
Confiança	0,2189	0,6576	0,2302	0,6608	**5,13%**	0,49%
Sup. abs. da regra	0,2189	0,6576	0,2303	0,6625	**5,20%**	**0,74%**
Dif. das conf. inv.	0,2189	0,6576	0,1840	0,6691	-15,96%	**1,75%**
Lift	0,2189	0,6576	0,2224	0,6599	1,58%	0,34%

Tabela 5.50: *Resultado obtido com a ordenação utilizando cada medida de regra de associação para o algoritmo Rankboost*

De acordo com os resultados expostos para as abordagem de expansão com o somatório das 100 características mais interessantes, os melhores resultados, foram em sua maioria obtidos utilizando o algoritmo Rankboost. Na utilização do algoritmo Randomforest, apenas a utilização das medidas “diferença da confiança invertida” e *lift* na coleção TD2003 obtiveram melhorias, já para a base HP2003 apenas as medidas “diferença da confiança invertida” e “confiança” obtiveram melhorias. Comparando-se os resultados obtidos nas expansões com 10 características, com 100 características, com 200 características e a atual, percebe-se aumento do ganho em algumas medidas.

CAPÍTULO 6

Conclusão

Com o advento da internet a quantidade de documentos produzidos aumentou, bem como a necessidade da melhoria na qualidade do ordenamento dos documentos relevantes tomou maior destaque. Nesse trabalho estudou-se sobre a seleção e geração de características utilizando regras de associação para o problema de ordenação de documentos em máquinas de busca. Estudou-se a seleção de medidas de regras de associação, analisou-se quais medidas regras produziriam melhores resultados na ordenação de documentos e realizou-se uma geração de características nas bases de dados a fim de obter aumento da acurácia nos métodos de ordenação de documentos.

Primeiramente foi avaliado se a utilização das diferentes medidas de avaliação de regras de associação produziriam bons resultados na ordenação de documentos utilizando o algoritmo LRAR. Nessa etapa percebeu-se que a utilização de outras medidas de avaliação de regras de associação poderiam ser utilizadas nas funções de ordenação para produzir bons ganhos de acurácia. Os melhores resultados obtidos dentre os rankings propostos foram os rankings 2, 4, 7 e 10.

A tabela 6.1 mostra um comparativo entre os melhores resultados obtidos na utilização do algoritmo LRAR com a modificação dos rankings propostos pelo presente trabalho de dissertação com os algoritmos estado da arte em *learning to rank*.

	MAP	
	TD2003	HP2003
Regression	0,2409	0,4968
RankSVM	0,2628	0,7408
ListNet	0,2733	0,7659
AdaRank	0,2283	0,7710
SVM	0,2445	0,7421
RankBoost	0,2274	0,7330
Frank	0,2031	0,6640
Regress+L2reg	0,2434	0,7486
RankSVMPrimal	0,2653	0,7645
RankSVMStruck	0,2713	0,7625
SmoothRank	0,2695	0,2695
LRAR-Ranking2	0,2153	0,5508
LRAR-Ranking4	0,2163	0,5668
LRAR-Ranking7	0,2210	0,6987
LRAR-Ranking10	0,2683	0,5957

Tabela 6.1: *Comparativo entre os melhores resultados obtidos na questão de pesquisa 1 e os melhores algoritmos de L2R*

Vale salientar que os resultados mostrados pela presente dissertação que são mostrados na tabela 6.1 foram obtidos pela execução do algoritmo LRAR sem as restrições de suporte e confiança mínimos. Dessa forma, percebe-se através da tabela que os resultados obtidos são competitivos com os algoritmos em estado da arte em *learning to rank*.

Após a etapa de utilização das medidas de avaliação de regras de associação em um método de ordenação, foram selecionadas aquelas que obtiveram resultados melhores e mais consistentes entre as coleções.

Os resultados obtidos nessa etapa permitiram observar que as medidas: confiança, suporte da regra, diferença da confiança inverida e *lift* produziram crescimento de acurácia ao se aumentar o tamanho de regra e obtiveram os melhores ganhos em MAP ao se comparar com a função de ordenação original.

Sendo assim, das onze funções de ordenamento testadas, essas quatro medidas (confiança, suporte da regra, diferença da confiança invertida e *lift*) foram as escolhidas para ordenar regras de associação geradas a partir do documento de treino para responder à segunda questão de pesquisa. Os antecedentes das melhores regras ordenadas com cada uma dessas características foram considerados padrões. Tais padrões foram utilizados na expansão das bases de dados de treino, teste e validação.

A etapa de expansão da coleção com antecedentes das regras de associação

foi realizada com duas abordagens diferentes: expansão das melhores características e expansão do somatório das melhores características. A primeira consistiu em considerar cada padrão como uma nova característica do documento nas bases de dados. Essa primeira abordagem foi realizada com os dez, cem e duzentos melhores padrões. A segunda consistiu em realimentar o documento das bases de dados com uma nova característica que seu valor é resultante do somatório do valor de cada padrão mais interessante na abordagem passada.

A figura 6.1 ilustra o resultado dos ganhos médios da execução dos algoritmos Randomforest e Rankboost para cada uma das abordagens de expansão utilizando como medida de ordenação a confiança, o suporte absoluto da regra, a diferença da confiança invertida e *lift*.

Randomforest												
Ganho na média das Folds	Confiança				Suporte da regra				Diferença das confianças invertidas				Lift			
	TD2003		HP2003		TD2003		HP2003		TD2003		HP2003		TD2003		HP2003	
Exp. 10 caract.	▲	3,47%	▲	0,97%	▲	7,59%	▲	0,62%	▼	-5,55%	▼	-0,22%	▼	-3,82%	▼	-1,95%
Exp. com 100 caract.	▲	0,89%	▲	0,21%	▲	7,32%	▲	0,57%	▼	-5,50%	▼	-0,92%	▼	-10,06%	▼	-3,96%
Exp. com 200 caract.	▼	-4,27%	▲	0,23%	▼	-3,65%	▲	2,40%	▼	-1,32%	▼	-0,01%	▼	-13,76%	▼	-3,38%
Exp. Som.	▼	-10,64%	▲	1,30%	▼	-30,80%	▼	-11,13%	▲	2,09%	▲	0,04%	▲	0,97%	▼	-3,78%

Rankboost												
Ganho na média das Folds	Confiança				Suporte da regra				Diferença das confianças invertidas				Lift			
	TD2003		HP2003		TD2003		HP2003		TD2003		HP2003		TD2003		HP2003	
Exp. 10 caract.	▲	5,25%	▲	0,96%	▲	2,04%	▲	0,96%	▲	2,69%	▲	1,03%	▲	5,30%	▲	0,56%
Exp. com 100 caract.	▲	5,25%	▲	0,96%	▲	3,87%	▲	0,10%	▲	7,46%	▼	-2,80%	▲	5,23%	▲	0,55%
Exp. com 200 caract.	▲	5,25%	▲	0,96%	▲	3,68%	▲	0,55%	▲	5,79%	▼	-5,71%	▲	3,97%	▲	0,37%
Exp. Som.	▲	5,13%	▲	0,49%	▲	5,20%	▲	0,74%	▼	-15,96%	▲	1,75%	▲	1,58%	▲	0,34%

Figura 6.1: *Resultado dos ganhos médios da execução dos algoritmos Randomforest e Rankboost para cada uma das abordagens de expansão*

Os resultados obtidos nas expansões dos padrões mais interessantes permitiram observar que os melhores ganhos de MAP ocorreram na utilização do algoritmo Rankboost e sobre a expansão dos cem padrões mais interessantes. Tal resultado levou a crer que o aumento de padrões com grande quantidade de valores igual a zero resultou na deterioração da acurácia do ordenamento dos métodos testados. Essa conclusão permitiu crer que a abordagem de somatório dos cem padrões mais interessantes pudesse produzir melhores resultados em ambos métodos de ordenamento.

Observando a figura 6.1 percebe-se que na utilização da abordagem de somatório dos padrões mais interessantes foram obtidos resultados consistentes e com melhorias para as medidas de avaliação Suporte da regra e *lift* para a execução do algoritmo

Rankboost. Já para o Randomforest, apenas as medidas de avaliação *lift* e diferença da confiança invertida obtiveram resultados satisfatórios nessa abordagem.

Tais resultados ilustrados na figura 6.1 mostraram que a geração de características baseadas em regras de associação pode ser promissora e a relevância de que sejam feitos mais estudos sobre o tema.

6.1 Trabalhos Futuros

Com o presente trabalho é possível visualizar novas possibilidades de pesquisas voltadas para o problema de ordenação de documentos. Em virtude disso, como trabalhos futuros, sugere-se realizar os seguintes estudos:

- Utilização de mais de uma medida de avaliação de regra de associação para realização da expansão do somatório dos melhores padrões para verificar se há o aumento de acurácia.
- Utilização de algoritmos genéticos para realizar uma combinação entre as *features* e as métricas a fim de melhorar a acurácia.
- Propor novos métodos de utilização de regras de associação para o problema de ordenação de documentos.

Referências Bibliográficas

[1] AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. **Mining association rules between sets of items in large databases**. *SIGMOD Rec.*, 22(2):207–216, 1993.

[2] BAEZA-YATES, R. A.; RIBEIRO-NETO, B. **Modern Information Retrieval**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[3] BURGES, C.; SHAKED, T.; RENSHAW, E.; LAZIER, A.; DEEDS, M.; HAMILTON, N.; HULLENDER, G. **Learning to rank using gradient descent**. In: *Proceedings of the 22nd international conference on Machine learning*, ICML '05, p. 89–96, New York, NY, USA, 2005. ACM.

[4] BURGES, C. J.; RAGNO, R.; LE, Q. V. **Learning to rank with nonsmooth cost functions**. In: *NIPS*, volume 6, p. 193–200, 2006.

[5] BURGES, C. J.; RAGNO, R.; LE, Q. V. **Learning to rank with nonsmooth cost functions**. In: *NIPS*, volume 6, p. 193–200, 2006.

[6] BUSH, V.; THINK, A. W. M. **The atlantic monthly**. *As We May Think*, 176(1):101–108, 1945.

[7] CAO, Y.; XU, J.; LIU, T.-Y.; LI, H.; HUANG, Y.; HON, H.-W. **Adapting ranking svm to document retrieval**. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 186–193. ACM, 2006.

[8] CAO, Z.; QIN, T.; LIU, T.-Y.; TSAI, M.-F.; LI, H. **Learning to rank: from pairwise approach to listwise approach**. In: *Proceedings of the 24th international conference on Machine learning*, ICML '07, p. 129–136, New York, NY, USA, 2007. ACM.

[9] COOPER, W. S. **The formalism of probability theory in ir: A foundation or an encumbrance?** In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, p. 242–247, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[10] COSSOCK, D.; ZHANG, T. **Subset ranking using regression**. In: *Proceedings of the 19th Annual Conference on Learning Theory*, COLT'06, p. 605–619, Berlin, Heidelberg, 2006. Springer-Verlag.

[11] CRAMMER, K.; SINGER, Y. **Pranking with ranking**. In: *Advances in Neural Information Processing Systems 14*, p. 641–647. MIT Press, 2001.

[12] CRESTANI, F.; LALMAS, M.; RIJSBERGEN, C. J. V. A. N.; CAMPBELL, I. **"Is This Document Relevant ?... Probably": A Survey of Probabilistic Models in Information Retrieval**. 30(4):528–552, 1999.

[13] DANG, V.; CROFT, W. B. **Feature selection for document ranking using best first search and coordinate ascent**. In: *Sigir workshop on feature generation and selection for information retrieval*, 2010.

[14] DANG, V.; CROFT, W. B. **Feature selection for document ranking using best first search and coordinate ascent**. In: *Sigir workshop on feature generation and selection for information retrieval*, 2010.

[15] FAN, W.; GORDON, M. D.; PATHAK, P. **Genetic programming-based discovery of ranking functions for effective web search**, 2005.

[16] FREITAS, A. A. **On rule interestingness measures**. *Knowledge-Based Systems*, 12:309–315, 1999.

[17] FREUND, Y.; IYER, R.; SCHAPIRE, R. E.; SINGER, Y. **An efficient boosting algorithm for combining preferences**. *The Journal of machine learning research*, 4:933–969, 2003.

[18] FUHR, N. **Probabilistic models in information retrieval.** *Comput. J.*, 35(3):243–255, 1992.

[19] GENG, X.; LIU, T.-Y.; QIN, T.; LI, H. **Feature selection for ranking**. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, p. 407–414, New York, NY, USA, 2007. ACM.

[20] GOKER, A.; DAVIES, J. **Information retrieval: searching in the 21st century**. John Wiley & Sons, 2009.

[21] GUYON, I.; ELISSEEFF, A. **An introduction to variable and feature selection**. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[22] HANG, L. **A short introduction to learning to rank**. *IEICE TRANSACTIONS on Information and Systems*, 94(10):1854–1862, 2011.

[23] HERBRICH, R.; GRAEPEL, T.; OBERMAYER, K. **Large margin rank boundaries for ordinal regression**. *Advances in Neural Information Processing Systems*, p. 115–132, 1999.

[24] HERBRICH, R.; GRAEPEL, T.; OBERMAYER, K. **Large margin rank boundaries for ordinal regression**. MIT Press, Cambridge, MA, 2000.

[25] JONES, K. S.; WALKER, S.; ROBERTSON, S. E. **A probabilistic model of information retrieval: Development and comparative experiments**. *Inf. Process. Manage.*, 36(6):779–808, Nov. 2000.

[26] KOHAVI, R.; JOHN, G. H. **Wrappers for feature subset selection**. *Artif. Intell.*, 97(1-2):273–324, Dec. 1997.

[27] LI, P.; BURGES, C. J.; WU, Q.; PLATT, J.; KOLLER, D.; SINGER, Y.; ROWEIS, S. **Mcrank: Learning to rank using multiple classification and gradient boosting.** In: *NIPS*, volume 7, p. 845–852, 2007.

[28] LIU, T.-Y. **Learning to rank for information retrieval**. *Found. Trends Inf. Retr.*, 3(3):225–331, Mar. 2009.

[29] LIU, T.-Y. **Learning to rank for information retrieval**. springer, 2011.

[30] LUHN, H. P. **A statistical approach to mechanized encoding and searching of literary information**. *IBM J. Res. Dev.*, 1(4):309–317, Oct. 1957.

[31] LUHN, H. P. **The automatic creation of literature abstracts**. *IBM J. Res. Dev.*, 2(2):159–165, Apr. 1958.

[32] MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to Information Retrieval**. Cambridge University Press, New York, NY, USA, 2008.

[33] MARON, M. E.; KUHNS, J. L. **On relevance, probabilistic indexing and information retrieval**. *J. ACM*, 7(3):216–244, July 1960.

[34] MARON, M. E.; KUHNS, J. L.; RAY, L. C. **Probabilistic indexing: A statistical approach to the library problem**. In: *Preprints of Papers Presented at the 14th National Meeting of the Association for Computing Machinery*, ACM '59, p. 1–2, New York, NY, USA, 1959. ACM.

[35] MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. MIT Press, 2012.

[36] MOOERS, C. N. **Zatocoding applied to mechanical organization of knowledge**. *American documentation*, 2(1):20–32, 1951.

[37] QIN, T.; LIU, T.-Y.; XU, J.; LI, H. **Letor: A benchmark collection for research on learning to rank for information retrieval**. *Inf. Retr.*, 13(4):346–374, Aug. 2010.

[38] QIN, T.; LIU, T.-Y.; ZHANG, X.-D.; WANG, D.-S.; XIONG, W.-Y.; LI, H. **Learning to rank relational objects and its application to web search**. In: *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, p. 407–416, New York, NY, USA, 2008. ACM.

[39] QIN, T.; ZHANG, X.-D.; TSAI, M.-F.; WANG, D.-S.; LIU, T.-Y.; LI, H. **Query-level loss functions for information retrieval**. *Inf. Process. Manage.*, 44(2):838–855, Mar. 2008.

[40] RIJSBERGEN, C. J. V. **Information Retrieval**. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[41] RIJSBERGEN, C. J. V. **Information Retrieval**. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[42] ROBERTSON, S. E.; SPARCK JONES, K. **Document retrieval systems**. chapter Relevance Weighting of Search Terms, p. 143–160. Taylor Graham Publishing, London, UK, UK, 1988.

[43] SALTON, G. **The SMART Retrieval System - Experiments in Automatic Document Processing**. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[44] SHASHUA, A.; LEVIN, A. **Ranking with large margin principle: Two approaches**. *Advances in neural information processing systems*, p. 961–968, 2003.

[45] SONG, Y.; LEUNG, K.; FANG, Q.; NG, W. **Fp-rank: An effective ranking approach based on frequent pattern analysis**. In: *Database Systems for Advanced Applications*, p. 354–369. Springer, 2013.

[46] SPARCK JONES, K. **Document retrieval systems**. chapter A Statistical Interpretation of Term Specificity and Its Application in Retrieval, p. 132–142. Taylor Graham Publishing, London, UK, UK, 1988.

[47] SPINK, A.; WOLFRAM, D.; JANSEN, M. B. J.; SARACEVIC, T. **Searching the web: The public and their queries**. *J. Am. Soc. Inf. Sci. Technol.*, 52(3):226–234, Feb. 2001.

[48] VELOSO, A. A.; ALMEIDA, H. M.; GONÇALVES, M. A.; MEIRA JR., W. **Learning to rank at query-time using association rules**. In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, p. 267–274, New York, NY, USA, 2008. ACM.

[49] WU, Q.; BURGES, C. J.; SVORE, K. M.; GAO, J. **Ranking, boosting, and model adaptation**. *Tecnical Report, MSR-TR-2008-109*, 2008.

[50] WU, Q.; BURGES, C. J.; SVORE, K. M.; GAO, J. **Adapting boosting for information retrieval measures**. *Information Retrieval*, 13(3):254–270, 2010.

[51] YUE, Y.; FINLEY, T.; RADLINSKI, F.; JOACHIMS, T. **A support vector method for optimizing average precision**. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, p. 271–278, New York, NY, USA, 2007. ACM.

[52] ZHENG, Z.; ZHA, H.; ZHANG, T.; CHAPELLE, O.; CHEN, K.; SUN, G. **A general boosting method and its application to learning ranking functions for web search.** In: *NIPS*, volume 20, p. 1697–1704, 2007.