# ALGORITMOS DE APRENDIZADO ASSOCIATIVO PARA ORDENAÇÃO DE DOCUMENTOS

Proposta submetida ao CNPq
Edital MCT/CNPq No. 014/2010 - Universal.

**Coordenador: Adriano Alonso Veloso**
**UNIVERSIDADE FEDERAL DE MINAS GERAIS**
**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**
**Julho/2010**

## 1. INTRODUÇÃO

Diversas aplicações precisam apresentar resultados na forma de listas ordenadas. Este é o caso de várias aplicações ligadas a tarefas de Recuperação de Informação, nas quais os documentos precisam ser retornados ao usuário de forma ordenada (i.e., máquinas de busca, como o *Google* apresentam os documentos mais relevantes antes daqueles menos relevantes aos desejos do usuário).

A relevância de um documento, porém, não é um atributo que possa ser bem definido − documentos altamente relevantes para um usuário podem não ser tão relevantes para outros usuários. Analisando propriedades associadas a documentos que geralmente são considerados relevantes pela maioria dos usuários, e confrontando-as com as propriedades observadas em documentos considerados pouco relevantes pelos mesmos, constata-se que algumas dessas propriedades poderiam ser utilizadas para discriminar ou diferenciar documentos relevantes de documentos irrelevantes. Dentre tais propriedades pode-se destacar *PageRank* [4], *HITS* [9], BM25 [12], TF-IDF [13], entre várias outras [3]. O próximo passo foi utilizar tais propriedades para estimar da relevância de documentos em geral, e de posse dessas estimativas, os documentos poderiam finalmente ser ordenados pelos respectivos graus de relevância. Evidentemente, devido à impossibilidade de uma definição formal para a relevância de documentos, erros de estimativa são esperados.

Com o objetivo de atingir taxas de efetividade cada vez maiores, tem-se observado uma tendência crescente em métodos de ordenação de documentos que combinam diferentes propriedades, ao invés de métodos de ordenação que utilizam propriedades de forma isolada. Mais especificamente, tais métodos produzem uma função de ordenação, que recebe um documento como entrada e retorna uma estimativa para sua relevância. Contudo, é inviável criar tal função de ordenação de forma manual, pois não se sabe de antemão os pesos a serem associados a cada uma das propriedades. Dessa forma, existe um interesse crescente em "treinar" as máquinas de busca usando-se aprendizado de máquina, de forma que elas possam combinar automaticamente as propriedades dos documentos, e produzir funções de ordenação cada vez mais precisas [14]. O estado-da-arte em termos de máquina de busca vem utilizando cada vez mais frequentemente essa estratégia, chamada de *Ordenação de Documentos baseada em Aprendizado de Máquina*.

Devido ao enorme interesse na área, a ordenação de documentos emergiu como uma nova classe de problemas de aprendizado de máquina [10]. No entanto, o problema de ordenação possui particularidades que o tornam bem distinto dos problemas clássicos, tipicamente enquadrados como problemas de classificação ou de regressão. Uma dessas particularidades é a noção intrínsica de contexto, que é naturalmente decorrente da consulta especificada pelo usuário. Mais especificamente, a relevância de um documento depende não apenas de suas características e propriedades, mas também da consulta que foi realizada (i.e., o mesmo documento pode ter graus de relevância diferentes para duas consultas diferentes). Como consequência, os algoritmos de ordenação baseados em aprendizado de máquina devem levar em conta informação sobre a consulta [15], de forma a produzir funções de ordenação mais precisas [16, 19]. Outra nuance do problema de ordenação de documentos, é a vasta gama de tipos diferentes de documentos: texto, imagens, vídeos etc. Documentos de tipos diferentes geralmente estão associados a domínios e aplicações diferentes, e consequentemente, as propriedades usadas para estimar a relevância dos documentos são diferentes também (i.e., apropriedade "TF-IDF" não

se aplica a uma imagem [6], enquanto a propriedade “Palavra Visual” não se aplica a um documento textual). Por fim, outra particularidade do problema de ordenação de documentos é a maior dificuldade em se combinar resultados obtidos por diferentes funções de ordenação. Mais especificamente, enquanto em problemas típicos de classificação a saída de uma função é a classe do documento (e as classes previstas podem ser combinadas de forma eficaz, por exemplo, com uma simples votação majoritária [8]), no problema de ordenação a saída é uma ordenação parcial dos documentos, o que impõe a necessidade por novos métodos de combinação [17].

Além disso, frequentemente as tarefas associadas à uma máquina de busca precisam ser executadas de forma rápida e interativa. Sendo assim, os algoritmos de aprendizado de máquina devem ser computacionalmente eficientes e capazes de prover tempos de execução aceitáveis. Na prática (i.e., fora do ambiente de laboratório), diversos algoritmos de aprendizado de máquina, como vários baseados em SVMs [20] e em redes neuronais [5], têm dificuldade em antender tais requisitos computacionais. Recentemente introduzimos um algoritmo de aprendizado de máquina comprovadamente eficiente, baseado em regras de associação [18], e instanciamos tal algoritmo para produzir funções de ordenação em [15]. Nosso algoritmo mapeia propriedades do documento a graus de relevância por meio de regras, e combina essas regras [1] de forma a produzir funções de mapeamento altamente precisas.

Na direção de avançar o estado-da-arte, este projeto tem como principal objetivo prover algoritmos mais eficazes, que sejam capazes de:

- processar o contexto dos documentos de forma a melhor diferenciá-los dependendo da consulta realizada.
- agregar diferentes funções de ordenação de forma a prover uma função mais precisa.

Além disso, os algoritmos a serem desenvolvidos deverão ser capazes de ordenar diferentes tipos de objetos, sejam eles documentos Web [15], imagens [6] ou *tags* [11] (em certas aplicações *tags* são vistas como sendo *links* para documentos e por isso a ordenação de *tags* é uma ordenação indireta de documentos). Os algoritmos a serem desenvolvidos serão baseados em regras de associação, de forma a manter a praticidade de uso e eficiência computacional.

## 2. Objetivos e Metas

O objetivo geral deste projeto é desenvolver novos algoritmos de ordenação de documentos baseados em aprendizado de máquina. Para isso, listamos os seguintes objetivos específicos:

- explorar informação implícita na consulta do usuário, de forma a produzir funções de ordenação mais precisas, capazes de detectar e diferenciar o contexto no qual o documento está inserido.
- avaliar a efetividade os algoritmos desenvolvidos em cenários de aplicação diferentes, e mediante diferentes tipos de documentos.
- combinar resultados fornecidos por diferentes funções de ordenação, de forma a se obter uma função híbrida ou agregada, que ofereça uma precisão superior.

## 3. Descrição

A seguir definimos o problema de interesse neste projeto: ordenação de documentos baseada em aprendizado de máquina. Em seguida discutiremos nossa abordagem para o problema.

A ordenação de documentos baseada em aprendizado de máquina é definida da seguinte forma: temos como entrada o conjunto de treino (que será referido por $\mathcal{D}$), que consiste de um conjunto de registros da forma $< q,d,r >$, onde $q$ é a consulta (representada como uma lista de termos $\{t_1,t_2,\ldots,t_n\}$, $d$ é um documento (representado por uma lista de propriedades $\{f_1,f_2,\ldots,f_m\}$ (i.e., valores de PageRank, BM25, TF-IDF e muitos outros atributos), e $r$ é a relevância do documento $d$ para a consulta $q$. A relevância do documento pode assumir valores discretos (muito relevante, relevante, pouco relevante etc.). O conjunto de treino é usado para construir uma função que relaciona propriedades do documento ao seu respectivo valor de relevância. O conjunto de teste (que será referido por $\mathcal{T}$) consiste de registros na forma $< q,d,? >$ para os quais apenas a consulta e o documento a ser retornado são conhecidos, mas o valor de relevância de $d$ a $q$ é desconhecido. A função aprendida utilizando-se o conjunto de treino é usada para produzir uma estimativa das relevâncias desses documentos a suas respectivas consultas, de forma que tais documentos possam ser ordenados de acordo com a estimativa fornecida.

### 3.1. Produzindo Funções de Ordenação com Regras de Associação

Existem incontáveis formas de se modelar um algoritmo de ordenação de documentos baseado em aprendizado de máquina. Os algoritmos existentes geralmente utilizam conceitos derivados de técnicas como redes neuronais artificiais [5], programação genética [2], e máquinas de suporte [20]. Recentemente elaboramos uma outra abordagem que consiste em explorar diretamente as associações que existem entre as propriedades do documento e sua relevância. Essas associações estão geralmente implícitas nos exemplos fornecidos no conjunto de treino, mas quando descobertas, elas podem revelar aspectos importantes sobre o fenômeno por trás dos exemplos. Tais aspectos pode ser explorados para a produção de funções de ordenação de documentos. Esta abordagem deu origem aos chamados *algoritmos de ordenação associativos* [15].

Os algoritmos de ordenação associativos exploram a relação entre propriedades do documento e sua relevância. Tal relação é representada por regras de associação [1] da forma $\mathcal{X} \rightarrow r_i$, onde cada regra indica uma associação entre o conjunto de propriedades $\mathcal{X}$=$\{f_j \wedge \ldots \wedge f_l\}$, e o grau de relevância $r_i$. Denotaremos por $\mathcal{R}$ o conjunto de regras extraídas do conjunto de treino $\mathcal{D}$. Essas regras podem misturar propriedades diferentes no antecedente, mas sempre contêm um grau de relevância no consequente.

Uma medida estatística, chamada de confiança e denotada por $\theta(\mathcal{X} \rightarrow r_i)$, quantifica a força da associação entre $\mathcal{X}$ e $r_i$, e é usada para estimar a relevância dos documentos no conjunto de teste.

O desafio principal que limita a utilização de tais algoritmos é a complexidade da tarefa de extração das regras (a complexidade pode crescer exponencialmente com a quantidade de propriedades). Recentemente introduzimos o conceito de *extração de regras sob demanda*, e demonstramos que esse tipo de extração tem complexidade polinomial [11].

**Extração de Regras sob Demanda:** O espaço de busca por regras é imenso, e dessa forma, restrições de custo devem ser impostas durante o processo de extração de regras. Geralmente, um limiar de frequência, denotado por $\sigma_{min}$, é usado com o objetivo de filtrar regras que contenham propriedades que apareçam em poucos documentos no conjunto de treino. Essa abordagem, embora seja simples, tem alguns problemas. Se o valor de $\sigma_{min}$ é muito baixo, um número grande de regras serão extraídas de $\mathcal{D}$, e na prática a maior parte dessas regras são inúteis por não trazerem informação que possa ser usada para estimar a relevância do documento $d$ (i.e., um regra $\mathcal{X} \rightarrow r_i$ só é considerada útil para um documento $d \in \mathcal{T}$ caso o conjunto de propriedades $\mathcal{X} \subseteq d$, caso contrário informação contida na regra não tem sentido para o documento $d$). De outra forma, se o valor de $\sigma_{min}$ é muito alto, algumas regras importantes não serão extrídas de $\mathcal{D}$, e não serão incluídas em $\mathcal{R}$. Isso causará problemas caso documentos em $\mathcal{T}$ sejam compostos por propriedades que raramente apareçam nos documentos em $\mathcal{D}$. Não existe um valor ótimo para $\sigma_{min}$, ou seja, não existe um valor único capaz de assegurar que somente regras úteis sejam extraídas de $\mathcal{D}$, e também que garanta que regras importantes não sejam extraídas.

A extração sob demanda soluciona os problemas mencionados acima. O processo de extração é postergado até que um registro $< q, d, ? >$ no conjunto de teste seja informado. Esse documento, $d$, é usado como um filtro que remove propriedades sem informação pertinente ao documento $d$, produzindo uma projeção do conjunto de treino para cada documento $d \in \mathcal{T}$ (denotado por $\mathcal{D}_d$), que é obtida através da remoção de toda a informação não pertinente ao documento $d$. Então, um conjunto de regras, denotado por $\mathcal{R}_d$ é extraído de $\mathcal{D}_d$, e é finalmente usado para estimar a relevância do documento $d \in \mathcal{T}$.

### 3.2. Funções Globais

Para que se possa estimar a relevância de um documento $d \in \mathcal{T}$, é preciso combinar todas a regras em $\mathcal{R}_d$. Nossa abordagem interpreta $\mathcal{R}_d$ como um conjunto de votos, onde cada regra $\{\mathcal{X} \xrightarrow{\theta} r_i\} \in \mathcal{R}_d$ é um voto dado por $\mathcal{X}$ para a relevância $r_i$. Votos possuem pesos diferentes, dependendo da confiança de cada regra. Os votos são somados e normalizados, formando finalmente a pontuação associada a cada grau de relevância $r_i$ para o documento $d$, como mostrado na Equação 1 (onde $\theta(\mathcal{X} \rightarrow r_i)$ é o valor da confiança para a regra $\{\mathcal{X} \rightarrow r_i\}$):

$$s(d, r_i) = \frac{\sum \theta(\mathcal{X} \rightarrow r_i)}{\mid \mathcal{R}_d \mid}, \text{ onde } \mathcal{X} \subseteq d \tag{1}$$

Consequentemente, para cada documento $d$, a pontuação associada com a relevância $r_i$ é dada pela confiança média das regras em $\mathcal{R}_d$ que predizem $r_i$. A chance de $d$ ter relevância $r_i$ é obtida pela normalização das pontuações, expressada por $\hat{p}(r_i|d)$ e mostrada na Equação 2:

$$\hat{p}(r_i|d) = \frac{s(d, r_i)}{\sum_{j=0}^{k} s(d, r_j)} \tag{2}$$

Finalmente, a relevância do documento $d$ é estimada por uma combinação linear

das chances associadas a cada grau de relevância, resultando na função de ordenação $rank(d)$, mostrada na Equação 3:

$$rank(d) = \sum_{i=0}^{k} \big(r_i \times \hat{p}(r_i|d)\big) \tag{3}$$

O valor de $rank(d)$ é uma estimativa da relevância do documento $d \in \mathcal{T}$ usando $\hat{p}(r_i|d)$. Essas estimativas são usadas para produzir listas ordenadas de documentos.

**Exemplo:** A Tabela 1 mostra um exemplo onde $\mathcal{D}$ contém três consultas. Para cada consulta três documentos são retornados, e cada documento é representado por três propriedades – PageRank, BM25 and $tf$ (cada propriedade assume valores dentro dos intervalos mostrados, e os intervalos são calculados usando o algoritmo de discretização descrito em [7]). Suponha que queiramos estimar a relevância do documento $d_{10}$. Nesse caso, o conjunto de treino original é projetado utilizando como filtro o documento $d_{10}$, o que resulta na projeção $\mathcal{D}_{d_{10}}$, que é mostrada na Tabela 2.

| | | Documentos Retornados | | | | |
|---|---|---|---|---|---|---|
| | Consulta | id | PageRank | BM25 | *tf* | Relevância |
| $\mathcal{D}$ | $q_1$ | $d_1$ | [0.85-0.92] | [0.36-0.55] | [0.23-0.27] | 1 |
| | | $d_2$ | [0.74-0.84] | [0.36-0.55] | [0.23-0.27] | 1 |
| | | $d_3$ | [0.74-0.84] | [0.56-0.70] | [0.46-0.61] | 0 |
| | $q_2$ | $d_4$ | [0.93-1.00] | [0.36-0.55] | [0.46-0.61] | 0 |
| | | $d_5$ | [0.85-0.92] | [0.56-0.70] | [0.62-0.76] | 1 |
| | | $d_6$ | [0.74-0.84] | [0.36-0.55] | [0.28-0.45] | 0 |
| | $q_3$ | $d_7$ | [0.74-0.84] | [0.22-0.35] | [0.12-0.22] | 0 |
| | | $d_8$ | [0.65-0.73] | [0.56-0.70] | [0.46-0.61] | 0 |
| | | $d_9$ | [0.85-0.92] | [0.71-0.80] | [0.46-0.61] | 1 |
| $\mathcal{T}$ | $q_4$ | $d_{10}$ | [0.51-0.64] | [0.36-0.55] | [0.28-0.45] | 0 |
| | | $d_{11}$ | [0.85-0.92] | [0.00-0.21] | [0.46-0.61] | 1 |
| | | $d_{12}$ | [0.74-0.84] | [0.56-0.70] | [0.46-0.61] | 0 |

**Tabela 1. Conjuntos de treino e de teste.**

| | id | PageRank | BM25 | *tf* | Relevância |
|---|---|---|---|---|---|
| $\mathcal{D}_{d_{10}}$ | $d_1$ | – | [0.36-0.55] | – | 1 |
| | $d_2$ | – | [0.36-0.55] | – | 1 |
| | $d_4$ | – | [0.36-0.55] | – | 0 |
| | $d_6$ | – | [0.36-0.55] | [0.28-0.45] | 0 |

**Tabela 2. Projeção para o documento $d_{10}$.**

A quatro regras a seguir são estraídas de $\mathcal{D}_{d_{10}}$:

1. $tf$=[0.28-0.45] $\rightarrow$ $r$=0 ($\theta = 1.00$)
2. BM25=[0.36-0.55] $\rightarrow$ $r$=0 ($\theta = 0.50$)
3. BM25=[0.36-0.55] $\rightarrow$ $r$=1 ($\theta = 0.50$)

4. BM25=[0.36-0.55] $\wedge$ $tf$=[0.28-0.45] $\rightarrow$ $r$=0 ($\theta = 1.00$)

As previsões realizadas por essas regras são combinadas de acordo com as Equações 1 e 2, de forma a produzir $\hat{p}(r_i|d_{10})$. Finalmente, de acordo com a Equação 3, $rank(d_{10})$=0.37. Segunindo o mesmo processo, obtemos $rank(d_{11})$=0.54 e $rank(d_{12})$=0.24.

### 3.3. Funções Estáveis

Documentos associados a consultas diferentes são geralmente interpretados de formas difernetes, dependendo do contexto imposto pela consulta. Um documento considera muito relevante para uma certa consulta, pode ser considerado pouco relevante para outras consultas. Nesta seção nós apresnetamos propostas para explorar a informação referente à consulta, com o objetivo de melhorar a efetividade das funções de ordenação a serem produzidas. Para que a informação referente à consulta possa ser processada, as regras extraídas (que são chamadas regras locais) tem a forma $\{q \wedge \mathcal{X} \rightarrow r_i\}$, onde $q$ é a consulta. A seguir, discutimos como explorar essas regras para produzir funções estáveis.

A regra $\{\mathcal{X} \rightarrow r_i\}$ é dita ser estável se a força da associação entre $\mathcal{X}$ e $r_i$ não varia muito entre diferentes consultas.

**Estabilidade da regra:** a regra $\{\mathcal{X} \rightarrow r_i\}$ é estável se:

$$\forall q_j, |\theta(\mathcal{X} \rightarrow r_i) - \theta(q_j \wedge \mathcal{X} \rightarrow r_i)| \leq \phi_{min}$$

Quanto menor o valor de $\phi_{min}$ é, o mais estável é a regra. Regras estáveis são particularmente importante porque a previsões fornecidas por elas tendem a ser muito confiáveis. Denotamos por $\mathcal{R}_d^{\phi}$ o conjunto de regras estáveis extraídas de $\mathcal{D}_d$. De forma a se estimar a relevância do documento $d$, regras $\phi$-estáveis são combinadas de acordo com a Equação 4. Por fim, as Equações 1 e 2 são usadas para estimar a relevância de $d$.

$$s(d, r_i) = \frac{\sum \theta(\mathcal{X} \rightarrow r_i)}{\mid \mathcal{R}_d^{\phi} \mid}, \text{ onde } \mathcal{X} \subseteq d \tag{4}$$

### 3.4. Funções Locais

Não se espera que uma única função de ordenação seja capaz de refletir a verdadeira relação entre propriedades e a relevância de documentos. Isso ocorre porque a relevância dos documentos não obedece uma única distribuição, mas sim várias distribuições diferentes, dependendo do contexto da consulta. Nesta seção propomos o uso de regras locais para produzir múltiplas funções de ordenação, que são chamadas de funções locais. Tais funções levam em consideração o contexto de cada consulta, como mostrado nas Equações 5, 6 e 7:

$$s(q, d, r_i) = \frac{\sum \theta(q \wedge \mathcal{X} \rightarrow r_i)}{\mid \mathcal{R}_d \mid}, \text{ onde } \mathcal{X} \subseteq d \tag{5}$$

$$\hat{p}(r_i|d,q) = \frac{s(q,d,r_i)}{\sum_{j=0}^{k} s(q,d,r_j)} \tag{6}$$

$$rank(q,d) = \sum_{i=0}^{k} (r_i \times \hat{p}(r_i|d,q)) \tag{7}$$

Funções locais diferentes podem fornecer estimativas diferentes para o mesmo documento. Por exemplo, $rank(q_1, d_{12})$=0.35, $rank(q_2, d_{12})$=0.50, e $rank(q_3, d_{12})$=0.36. Isso sugere que funções locais diferentes são apenas capazes de estimar de forma procesa as relevâncias de certos documentos. O mapeamento ótimo entre funções locais e documentos é informação valiosa. A seguir propomos uma abordagem para se aproximar esse mapeamento. Começamos pela definição de *competência de ordenação*, e depois discutimos como separar documentos que são ordenados de forma competente por uma função, daqueles documentos que não são.

**Competência de Ordenação:** a competência de uma função local, que é denotada por $\Delta(q,d)$, é definida como:

$$\Delta(q,d) = |rank(q,d) - r^d| \tag{8}$$

A competência de uma função em relação a um documento $d$, é essencialmente a discreância entre a estimativa e a relevância real. Uma função local $rank(q_a, d)$ é mais competente que outra função local $rank(q_b, d)$ se $\Delta(q_a, d) < \Delta(q_b, d)$.

A competência de uma função local é informação nova, derivada apenas do conjunto de treino, que pode ser utilizada para fortalecer o conjunto de treino original, $\mathcal{D}$. Mais especificamente, para cada documento $d \in \mathcal{D}$, é informado de qual contexto foi produzida a função mais precisa (ou mais competente). Essa informação é obtida estimando-se a relevância de cada documento em $\mathcal{D}$. Esse processo resulta em um conjunto de treino "fortalecido", que é denotado por $\mathcal{D}^*$. Inicialmente, $\mathcal{D}^*$ está vazio. A cada iteração, um documento $d \in \mathcal{D}$ juntamente com a consulta associada à função local mais competente em relação a $d$ são incluídos em $\mathcal{D}^*$. Esse processo continua até que todos os documentos em $\mathcal{D}$ sejam incluídos em $\mathcal{D}^*$, como mostrado na Tabela 3.

**Mapeando Documentos a Funções:** O conjunto de treino "fortalecido", $\mathcal{D}^*$, pode ser usado para aproximar o mapeamento entre documentos e funções locais. Mais especificamente, ao invés de extrair regras da forma $\{\mathcal{X} \rightarrow r_i\}$, primeiro extraí-se regras da forma $\{\mathcal{X} \rightarrow q_i\}$ (i.e., o antecedente é um conjunto de propriedades do documento e o consequente é a consulta associada à funcção local mais competente). Essas regras são usadas para aproximar o casamento entre documentos e funções, de acordo com a Equações 9 e 10 (onde $n$ é o número de consultas em $\mathcal{D}$). Quanto maior o valor de $\hat{p}(q_i|d)$ é, maior é a chance de que $\Delta(q_i, d)$ seja baixo.

| | Documentos Retornados | | | | Função local mais competente |
|---|---|---|---|---|---|
| | id | PageRank | BM25 | *tf* | |
| $\mathcal{D}^*$ | $d_1$ | [0.85-0.92] | [0.36-0.55] | [0.23-0.27] | $q_3$ |
| | $d_2$ | [0.74-0.84] | [0.36-0.55] | [0.23-0.27] | $q_1$ |
| | $d_3$ | [0.74-0.84] | [0.56-0.70] | [0.46-0.61] | $q_3$ |
| | $d_4$ | [0.93-1.00] | [0.36-0.55] | [0.46-0.61] | $q_2$ |
| | $d_5$ | [0.85-0.92] | [0.56-0.70] | [0.62-0.76] | $q_2$ |
| | $d_6$ | [0.74-0.84] | [0.36-0.55] | [0.28-0.45] | $q_3$ |
| | $d_7$ | [0.74-0.84] | [0.22-0.35] | [0.12-0.22] | $q_2$ |
| | $d_8$ | [0.65-0.73] | [0.56-0.70] | [0.46-0.61] | $q_1$ |
| | $d_9$ | [0.85-0.92] | [0.71-0.80] | [0.46-0.61] | $q_3$ |

**Tabela 3. Conjunto de treino "fortalecido". A última coluna denota a função local mais competente para o documento.**

$$s(d, q_i) = \frac{\sum \theta(\mathcal{X} \rightarrow q_i)}{\mid \mathcal{R}_d \mid}, \text{ onde } \mathcal{X} \subseteq d \tag{9}$$

$$\hat{p}(q_i|d) = \frac{s(d, q_i)}{\sum_{j=0}^{n} s(d, q_j)} \tag{10}$$

Agora podemos combinar diferentes funções locais. Para que tal combinação possa ser feita, precisamos encontrar os parâmetros apropriados de forma que a combinação ofereça a melhor estimativa possível. O casamento entre documentos e funções locais pode ser usado para encontrar esses parâmetros, como mostrado na Equação 11.

$$rank(d) = \sum_{i=0}^{k} \left( r_i \times \sum_{j=0}^{n} \left( \hat{p}(r_i|d, q_j) \times \hat{p}(q_j|d) \right) \right) \tag{11}$$

A idéia básica é ponderar as estimativas fornecidas por cada função local usando-se a estimativa de competência de cada função. Intuivamente, se uma função local provavelmente fornece estimativas precisas para um documento, então essas estimativas serão ponderadas mais fortemente.

### 3.5. Funções Híbridas

Também podemos combinar funções globais produzidas por diferentes algoritmos de ordenação de documentos. Novamente, a idéia de competência pode ser explorada, mas ao invés da competência ser associada à uma consulta, agora a competência deve ser associada a um algoritmo. Por fim, a relevância de alguns documentos será estimada por certos algoritmos (que são mais competentes para esses documentos), enquanto a relevância de outros documentos serão estimadas por outros algoritmos (i.e., aproxima-se o mapeamento entre algoritmos e documentos).

A Figura 1 nos permite viualizar áreas de competência para 7 algoritmos diferentes. Regiões mais claras mostram conjuntos de documentos que são ordenados de forma

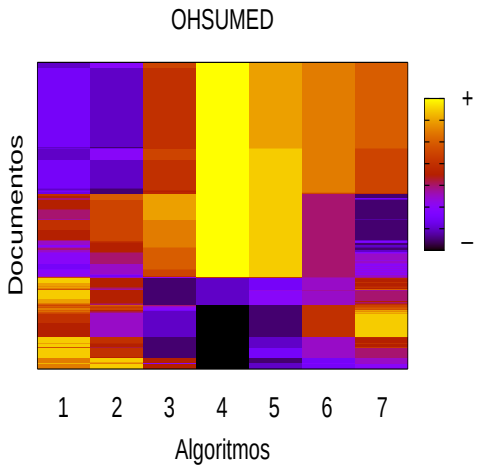


**Figura 1. Competência de cada algoritmo.**

precisa pelo algoritmo correspondente. Um desafio nesse caso, é mapear documentos aos algoritmos apropriados.

## 4. Metodologia de Execução

A metodologia a ser utilizada para o desenvolvimento deste projeto pode ser definida da seguinte forme:

1. Levantamento bibliográfico: buscar artigos acadêmicos que tratem do problema de ordenação de diferentes tipos de documentos, e algoritmos que representam o estado-da-arte em ordenação de documentos baseada em aprendizado de máquina.
2. Preparação dos dados: aquisição e pré-processamento das bases de dados a serem utilizadas. Experimentos que envolvam a ordenação de documentos Web serão realizados utilizando-se as coleções disponibilizadas pela Microsoft Research Asia [*] e pelo Yahoo[†]. Experimentos envolvendo a ordenação de imagens serão realizados utilizando-se as coleções disponibilizadas pela ImageNet [‡]. Ordenação indireta de vídeos e músicas através de *tags* será realizada utilizando-se dados coletados do YouTube [§], Delicious [¶]e LastFM [‖].
3. Projeto do algoritmo: definição detalhada de cada um dos componentes dos algoritmos.
4. Implementação: implementação dos algoritmos definidos em detalhes na fase de projeto.
5. Execução e configuração dos parâmetros: execução dos algoritmos e testes com diferentes variações de parâmetros.
6. Avaliação: os algoritmos implementados serão comparados com representantes do estado-da-arte.

[*] http://research.microsoft.com/users/LETOR/
[†] http://learningtorankchallenge.yahoo.com/
[‡] http://www.image-net.org/challenges/LSVRC/2010/index
[§] http://www.youtube.com/
[¶] http://www.delicious.com/
[‖] http://www.lastfm.com

7. preparação e escrita de artigos cietíficos.

### 4.1. Cronograma de Execução

A Tabela 4 mostra o cronograma de execução deste projeto, baseado nas atividades descritas na Seção 4.

| Atividade | Bimestre | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **Levantamento bibliográfico** | ● | ● | ● | | | | | | | | | |
| **Preparação dos dados** | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | |
| Documentos Web | ● | ● | ● | | | | | | | | | |
| Imagens | | ● | ● | ● | ● | | | | | | | |
| Vídeos e músicas Web | | | | ● | ● | ● | ● | ● | ● | | | |
| **Modelagem dos algoritmos** | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | |
| Baseados em regras locais | ● | ● | ● | | | | | | | | | |
| Baseados em estabilidade | | | | ● | ● | ● | | | | | | |
| Baseados em competência | | | | | | | ● | ● | ● | | | |
| **Configuração dos parâmetros** | | | ● | | | ● | | | ● | | | |
| **Avaliação** | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |
| **Preparação de artigos** | | | | | ● | ● | | | ● | ● | ● | ● |

**Tabela 4. Cronograma de atividades do projeto.**

## 5. Contribuições e Resultados Esperados

A ordenação de documentos é parte fundamental de diversos problemas ligados à Recuperação de Informação. O desenvolvimento de novos algoritmos de ordenação baseados em aprendizado de máquina possibilitará uma melhoria na qualidade de recuperação da informação. Sendo assim, várias tecnologias relacionadas a tarefas de Recuperação de Informação serão potencialmente afetadas pelos algoritmos a serem desenvolvidos neste projeto.

Vale mencionar que este projeto será desenvolvido dentro do contexto do Instituto Nacional de Ciência e Tecnologia para Web (InWeb), do qual o coordenador é membro. O InWeb apresenta vários problemas desafiadores onde os algoritmos desenvolvidos neste projeto poderão ser avaliados em cenários mais realísticos.

Como resultados esperados deste projeto, destacamos:

- o desenvolvimento de algoritmos associativos para ordenação de documentos.
- a avaliação dos algoritmos desenvolvidos, e a comparação com o estado-da-arte.
- a publicação de pelos menos um artigo em conferência nacional, um artigo em conferência internacional, e um artigo de periódico internacional.
- contribuição para a formação de um aluno de iniciação científica.
- contribuição para a formação de um aluno de mestrado.

## 6. Orçamento

Os experimentos que serão realizados para validação dos algoritmos propostos neste projeto são de alto custo computacional, dado que as coleções a serem utilizadas contêm

alguns milhões de documentos. Além disso, de forma a respeitar a técnica experimental adequada, os algoritmos deverão ser executados várias vezes, para garantir a validade estatística dos experimentos.

Portanto, faz-se necessária a aquisição de uma máquina com alta capacidade de processamento, como a DELL PowerEdge, equipada com processador Intel Xeon Quad-Core 2x6MB de cache, 8GB de memória RAM, e 2 discos rígidos de 500 GB. Também está prevista a compra de uma impressora, utilizada para imprimir e documentar os resultados parciais deste projeto.

Além disso, estão sendo requisitados recursos para compra de alguns periféricos (e.g., discos rígidos, memória etc.), que poderão ser necessários para repor peças da máquina requisitada neste projeto, bem como equipamentos já existentes nos laboratórios dos pesquisadores participantes (e que entrarão como contrapartida) deste projeto. Também estão sendo requisitados recursos para a compra de material bibliográfico, essencial para um levantamento completo do estado da arte na área de pesquisa. Em relação a material de consumo, prevemos a compra de mídias para backup, como *pen-drives* e DVDs, e toner e itens de papelaria para a impressão dos resultados.

Finalmente, pretendemos submeter os resultados obtidos neste projeto em conferências nacionais e internacionais, para divulgação dos trabalhos e uma maior inserção dos pesquisadores com a comunidade internacional. Os trabalhos também poderão ser apresentados durante visitas técnicas a pesquisadores internacionais. Por isso, solicitamos uma passagem internacional e 5 diárias para apresentação destes trabalhos. Um resumo do orçamento está listado na Tabela 5.

Já a Tabela 6 descreve o cronograma físico-financeiro. Os equipamentos necessários para realização dos experimentos serão comprados no primeiro semestre do projeto. Aproximadamente metade do material bibliográfico será adquirida nesse mesmo período, mas conservaremos a segunda parte para atualizações durante os 24 meses de duração do projeto. O material de consumo foi igualmente dividido entre os quatro semestres, já que sua utilização pode ser necessária durante todo o período. Já a previsão para uso das passagens e diárias é para o terceiro semestre deste projeto. Porém, ela vai depender da aceitação de artigos em conferências internacionais. Portanto, esse último item pode ser alterado futuramente.

## 7. Equipe

Além do coordenador, a equipe deste projeto conta com mais 4 membros, sendo um deles um aluno de iniciação científica, e o outro um aluno de mestrado, ambos a serem selecionados. São eles:

- Gisele Pappa, professora adjunta da UFMG e especialista em computação natural. Apoiará a utilização de métodos bio-inspirados para realizar o mapeamento entre documentos e funções locais, a partir do conceito de competência.
- Wagner Meira Jr., professor associado da UFMG e especialista em mineração de dados e recuperação de informação. Apoiará a utilização de métodos de mineração de dados para encontrar regras estáveis, bem como para propor técnicas de ponderação para a produção de funções híbridas.

Os alunos participarão de todas as etapas do projeto, auxiliando a modelagem, implementação e avaliação dos algoritmos propostos.

| Custeio | |
|---|---|
| **Material de Consumo** | |
| Periféricos | R$500,00 |
| Toner para impressora e itens de papelaria | R$700,00 |
| **Subtotal** | R$1.200,00 |
| **Passagens e diárias** | |
| 1 passagem aérea internacional | R$2.800,00 |
| 5 diárias região C (U$220,00 cotados a R$2,00) | R$2.200,00 |
| **Subtotal** | R$5.000,00 |
| Capital | |
| **Material Bibliográfico** | R$1.200,00 |
| **Equipamento** | |
| 2 máquinas para execução de experimentos | R$12.000,00 |
| 1 impressora | R$600,00 |
| **Subtotal** | R$13.800,00 |
| **Total** | R$20.000,00 |

**Tabela 5. Resumo dos recursos solicitados.**

| **Despesas** | **Semestre** | | | |
|---|---|---|---|---|
| | **1** | **2** | **3** | **4** |
| **Material de Consumo** | R$300,00 | R$300,00 | R$300,00 | R$300,00 |
| **Passagens e diárias** | | | R$5.000,00 | |
| **Material Bibliográfico** | R$600,00 | R$300,00 | | R$300,00 |
| **Equipamento** | R$12.600,00 | | | |

**Tabela 6. Cronograma físico-financeiro.**

## 8. Colaborações

Apesar de não fazerem parte da equipe deste projeto, a equipe está em constante colaboração com o grupo do pesquisador Mohammed Zaki, da RPI (Rensselaer Polytechnic Institute), e com o grupo do pesquisador Ricardo da Silva Torres, da UNICAMP (Universidade Estadual de Campinas). Os grupos irão cooperar no desenvolvimento dos algoritmos propostos neste projeto.

## 9. Contrapartida

Atualmente, o coordenador deste projeto está vinculado ao Laboratório de Processamento de Alto Desempenho (SPEED), coordenado pelo Prof. Virgílio Almeida. Ele têm ótima infra- estrutura computacional que, juntamente com os equipamentos que estão sendo solicitados neste projeto, darão suporte ás atividades previstas. O laboratório possui cerca de 25 servidores de alto desempenho, organizados em dois *clusters* conectados por Gigabit Ethernet, e mais de 25 estações de desenvolvimento. O laboratório está conectado à rede da UFMG através de uma conexão Ethernet (1 GBPS) e à Rede Nacional de Ensino e Pesquisa. Isso equivale a uma contrapartida de aproximadamente R$ 200.000,00.

Embora o SPEED conte com uma ótima infra-estrutura, o número de alunos utilizando os servidores disponíveis também é grande. No futuro, a intenção do proponente deste projeto é criar um novo laboratório de Aprendizado de Máqeuina, onde serão desenvolvidos trabalhos nessa área de pesquisa. Porém, faz-se necessária uma infra-estrutura mínima para requisição de espaço físico para tal laboratório. Portanto, até que o laboratório não seja criado, os equipamentos solicitados ficarão no laboratóorio SPEED.

## 10. Recursos de outras fontes

O coordenador do projeto e os 2 outros pesquisadores são pesquisadores participantes do Instituto Nacional de Ciência e Tecnologia para a Web (InWeb).

## Referências


[1] R. Agrawal, T. Imielinski, and A. Swami:. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[2] H. Almeida, M. Gonçalves, M. Cristo, and P. Calado. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 399–406, 2007.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1999.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine Learning*, pages 89–96, 2005.

[6] F. Faria, A. Veloso, H. Almeida, E. Valle, R. da Silva Torres, M. Gonçalves, and W. Meira Jr. Learning to rank for content-based image retrieval. In *Proceedings of the ACM Multimedia Information Retrieval*, pages 285–294, 2010.

[7] U. Fayyad and K. Irani. Multi interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[8] S. Goldman and M. Warmuth. Learning binary relations using weighted majority voting. *Machine Learning*, 20(3):245–271, 1995.

[9] J. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[10] T. Liu. *Learning to Rank for Information Retrieval*. Now Publishers, Inc., New York, USA, 2009.

[11] G. Menezes, J. Almeida, F. Belém, M. Gonçalves, A. Lacerda, E. Moura, G. Pappa, A. Veloso, and N. Ziviani. Demand-driven tag recommendation. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, 2010 (aceito para publicação).

[12] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC*, 1994.

[13] G. Salton, E. Fox, and H. Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, 1983.

[14] A. Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.

[15] A. Veloso, H. Almeida, M. Gonçalves, and W. Meira Jr. Learning to rank at query-time using association rules. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–274, Singapore, 2008.

[16] A. Veloso, W. Meira Jr., M. Gonçalves, and H. Almeida. Learning to rank using query-level rules. *Journal of Information and Data Management*, 2010 (aceito para publicação).

[17] A. Veloso, W. Meira Jr., M. Gonçalves, and H. Almeida. Competence-conscious associative rank aggregation by meta learning. In *Proceedings of the International Conference on Data Mining*, 2010 (submetido).

[18] A. Veloso, W. Meira Jr., and M. Zaki. Lazy associative classification. In *Proceedings of the International Conference on Data Mining*, pages 645–654, 2006.

[19] A. Veloso, W. Meira Jr., M. Zaki, M. Gonçalves, and H. Almeida. Calibrated lazy associative classification. *Information Sciences*, 2009. `http://dx.doi.org/10.1016/j.ins.2010.03.007`.

[20] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278, 2007.