Identificação de Comunidades na *Web* Usando Evidências e Algoritmos de Agrupamento Baseados em Grafos

Álvaro R. Pereira Jr José Augusto Nacif
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Belo Horizonte, Brasil
{alvaro, jnacif}@dcc.ufmg.br

Resumo

Este trabalho apresenta um modelo para a identificação de comunidades na Web. O modelo se baseia na construção de um grafo rotulado não direcionado $G(V, A)$ onde os vértices V representam documentos Web e as arestas A representam relações de similaridades entre as documentos Web. O rótulo das arestas representam pesos entre vértices. O cálculo do peso do grafo G é feito com base em evidências diferentes, compreendidas entre evidências de links, evidências sintáticas e evidências de localização. Quanto maior o peso das arestas, maior a similaridade entre os documentos Web. Após a construção do grafo são criados agrupamentos de documentos com maiores pesos em suas arestas, buscando identificar comunidades, através de um algoritmo baseado na árvore geradora mínima (AGM) do grafo G. Experimentos estão sendo realizados e serão apresentados em breve, utilizando uma coleção de páginas sob o domínio www.dcc.ufmg.br.

1. Introdução

O mundo real é dividido em comunidades onde cada indivíduo, uma entidade do mundo, é membro de diversas comunidades. As pessoas têm necessidade de se inserirem em novas comunidades, mas muitas vezes não fazem por não saberem onde ela está ou como chegar até ela. A Internet e a *Web* podem funcionar como facilitadores para a inserção de membros em comunidades, o que releva o estudo de identificação de agrupamentos de documentos, buscando identifcar as comunidades.

Este trabalho apresenta um modelo para identificação de comunidades na *Web*, realizando agrupamento de documentos de assuntos relacionados. O modelo é composto de dois estágios: combinação de evidências para geração de grafo rotulado e agrupamento usando árvore geradora mínima para particionamento do grafo.

No primeiro estágio, um grafo de evidências é construído a partir do grafo da *Web* (no caso desse trabalho, de uma coleção de documentos HTML). O grafo da *Web* considera cada documento como sendo um vértice e cada *link* como sendo uma aresta. Três tipos de evidências são consideradas: evidências de *link*, evidências sintáticas e evidências de localização. As evidências de *links*, quando identificadas, podem introduzir novas arestas no grafo, além das arestas já existentes que identificam um *link* entre documentos. As evidências sintáticas e de localização são somente complementares, podendo contribuir com o crescimento do peso das arestas, fortalecendo indícios de similaridade entre documentos. Detalhes sobre o estágio de combinação de evidências são apresentados na seção 2.

No segundo estágio, com o grafo já construído, deve-se realizar o agrupamento das páginas similares. Agrupamento de dados não é uma tarefa trivial. Não existe nenhum algoritmo de agrupamento que gere bons resultados para qualquer tipo de massa de dados e características desejadas de agrupamento. Pelo contrário, a maioria dos algoritmos funciona bem para um aplicação específica. Neste trabalho, foi escolhido um algoritmo baseado na teoria de grafos. Primeiramente a árvore geradora máxima (AGMáx) é criada e em seguida arestas que seja inferiores a um limite L são removidas do grafo criando assim uma série de agrupamentos. Detalhes de como o agrupamento das páginas é feito podem ser encontrados na Seção 3.

2. Combinação de Evidências

O trabalho em questão apresenta um mecanismo para agrupamento de documentos *Web*, utilizando tipos diferentes de evidências. As evidências consideradas são classificadas em três grupos: evidências de *links*, evidências sintáticas e evidências de localização. As evidências de *links* se baseiam na perspectiva da *Web* como um grafo direcionado [15, 17, 8], onde os documentos representam os vértices e os *links* representam arestas direcionadas. As evidências sintáticas utilizam informações textuais dos documentos para inferir relações entre os mesmos. Evidências de localização estão relacionadas à localização geográfica do documento e, principalmente, ao domínio no qual ele se encontra hospedado. A cada evidência verificada é atribuído um peso entre zero e um para a aresta, valor que é somado ao peso já atribuído para aquela aresta, caso exista.

O modelo proposto para a fase de combinação de evidências é apresentado na figura 1 e funciona da seguinte forma: na etapa 1 da figura é feito o *parsing* da coleção de documentos HTML, de forma a retirar os textos, os títulos e os *links* de cada documento. Na etapa 2 a estrutura de *links* entre os documentos da coleção é usada para gerar um grafo direcionado. As evidências de *links* são processadas de acordo com as direções das arestas, retornando um grafo não direcionado rotulado, com novas arestas, o que ocorre na etapa 3. Na etapa 4, as evidências sintáticas e de localização são processadas comparando somente os documentos que estão conectados por arestas, ou seja, que até este ponto possuem algum indício de tratarem de assuntos relacionados. Um grafo não direcionado rotulado é gerado, e deve ser particionado na fase de agrupamento, apresentada na seção 3. As próximas seções apresentam as evidências consideradas neste trabalho.

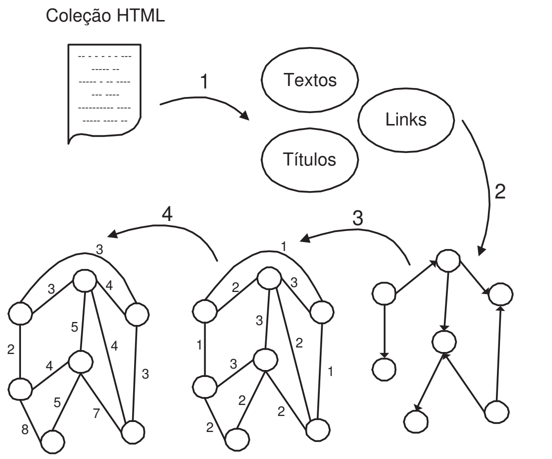

Figura 1. Modelo proposto para a fase de combinação de evidências.

2.1. Evidências de *Links*

Foram consideradas cinco evidências diferentes de *links*, a saber: estrutura de *links* entre os documentos; relação de co-citação; relação de acoplamento bibliográfico; Amsler, algoritmo que combina as relações de co-citação e acoplamento bibliográfico; e Companion, que também combina as duas relações. A figura 2-a) apresenta um exemplo de grafo composto de cinco documentos e suas estruturas de *links*. A estrutura de *links* é a primeira evidência de *link* a ser considerada e seu peso é sempre 1, quando existir um *link*. Durante a fase de coleta de evidências de *links* novas arestas serão inseridas, a medida em que várias evidências são combinadas. As seções seguintes apresentam os demais tipos de evidências considerados. conforme apresentado em [6].

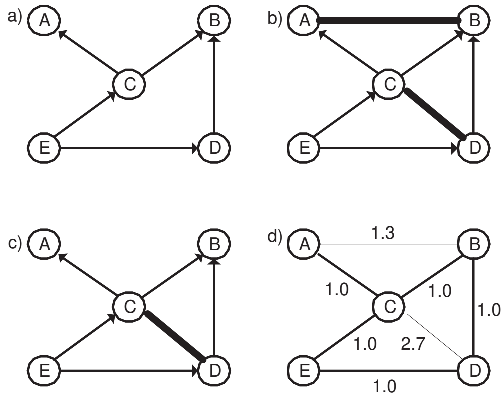

Figura 2. Grafos da *Web*. $a)$ Grafo de documentos (vértices) e *links* (arestas). $b)$ Co-citação. $c)$ Acoplamento bibliográfico. $d)$ Grafo rotulado não direcionado.

2.1.1. Co-Citação

A co-citação foi proposta por Small [18] como uma medida de similaridade entre artigos científicos. Dois artigos são co-citados se um terceiro artigo tem citações para ambos. Esta afirmação supõe que um autor de artigos científicos citará apenas artigos relacionados com seu próprio trabalho. Apesar de *links* da *Web* serem bastante diferentes de citações, é possível assumir que muitos deles têm o mesmo significado, ou seja, um autor de uma página *Web* irá inserir *links* para páginas relacionadas à sua própria página. Assim, é possível aplicar a co-citação em documentos *Web* tratando *links* como citações.

Para refinar melhor esta idéia, considere d uma página *Web* e considere P_d o conjunto de páginas que tenham *links* para d, chamados de pais de d. A similaridade de co-citação entre duas páginas d_1 e d_2 é definida como:

$$co - citacao(d_1, d_2) = \frac{P_{d_1} \bigcap P_{d_2}}{|P_{d_1} \bigcup P_{d_2}|}$$

A Equação acima mostra que quanto mais pais d_1 e d_2 tiverem em comum, mais relacionados eles são. Este valor é normalizado pelo conjunto total de pais, de forma que a similaridade de co-citação varie entre zero e um. Se ambos P_{d_1} e P_{d_2} estão vazios a similaridade de co-citação é zero. Como exemplo, considere o grafo da figura 2-a) . Como as páginas A e B são apontadas pela página C, elas possuem uma relação de co-citação. Desta forma, uma nova aresta deve ser inserida no grafo, o que representa um indício de similaridade entre A e B. O mesmo ocorre com as arestas C e D, que são apontadas por E. A figura 2-b) destaca as novas arestas inseridas.

2.1.2. Acoplamento Bibliográfico

Também com o objetivo de determinar a similaridade entre páginas, kessler [14] introduziu a medida de acoplamento bibliográfico. Dois documentos dividem uma unidade de acoplamento bibliográfico se ambos citam o mesmo artigo. Esta idéia é baseada na noção de que autores de artigos que trabalham no mesmo assunto tendem a citar os mesmos artigos. Assim como na co-citação pode-se aplicar este princípio para a *Web*. Assume-se que dois autores de páginas *Web* sobre o mesmo assunto tendem a inserir *links* para as mesmas páginas. Pode-se dizer então que duas páginas tem uma unidade de acoplamento bibliográfico entre si se ambas possuem um *link* para a mesma página. Mais formalmente, considere d uma página *Web*. Define-se C_d como o conjunto de páginas que d possui *link*, também chamados filhos de d. O acoplamento bibliográfico entre duas páginas d_1 e d_2 é definido como:

$$acoplamentobibliografico(d_1, d_2) = \frac{C_{d_1} \bigcap C_{d_2}}{|C_{d_1} \bigcup C_{d_2}|}$$

De acordo com a equação acima, quanto mais filhos em comum a página d_1 tem com a página d_2, mais relacionadas elas são. Este valor é normalizado pelo conjunto total de filhos entre zero e um. Se ambos C_{d_1} e C_{d_2} estão vazios, define-se o acoplamento bibliográfico como zero. Como exemplo, considere novamente o grafo da figura 2-a) . Como a página B é apontada pelas páginas C e D, elas possuem uma relação de acoplamento bibliográfico. Desta forma, uma nova aresta deve ser inserida no grafo, o que representa um indício de similaridade entre C e D. A figura 2-c) destaca a nova aresta inserida. Ao final da etapa de evidências de *links*, o grafo estará com um formato similar ao apresentado na figura 2-c) , rotulado mas sem conexões.

2.1.3. Amsler

Em uma tentativa de tirar maior vantagem da informação disponível entre artigos, Amsler [1] propôs uma medida de similaridade que combina co-citação e acoplamento bibliográfico. De acordo com Amsler, dois artigos A e B são relacionados se (1) A e B são citados pelo mesmo artigo, (2) A e B citam o mesmo artigo, ou (3) A cita um terceiro artigo C que cita B. Como para as medidas anteriores, pode-se aplicar a similaridade de Amsler para medir páginas *Web*, substituindo citações por *links*.

Considere d uma página *Web*, P_d o conjunto de pais de d, C_d o conjunto de filhos de d. A similaridade de Amster entre duas páginas d_1 e d_2 é definida como:

$$amsler(d_1, d_2) = \frac{(P_{d_1} \bigcup C_{d_1}) \bigcap (P_{d_2} \bigcup C_{d_2})}{|(P_{d_1} \bigcup C_{d_1}) \bigcup (P_{d_2} \bigcup C_{d_2})|}$$

A equação acima mostra que quanto mais *links* (tanto pais como filhos) d_1 e d_2 têm em comum mais eles são relacionados. A medida é normalizada pelo número total de *links*. Se nem d_1 nem d_2 têm filhos ou pais a similaridade é definida como zero. A similaridade de Amsler também foi usada como uma evidência de similaridade entre as páginas *Web*.

2.1.4. Companion

O algoritmo companion foi proposto por Dean e Henzinger [7]. Dada uma página d, o algoritmo encontra um conjunto de páginas relacionadas a d examinando a sua estrutura de *links*. Companion é capaz de retornar o quão cada página é relacionada a d. Este valor pode ser usado como medida de similaridade entre d e outras páginas.

Para achar um conjunto de páginas relacionados à página d, o algoritmo companion possui dois passos: (1) Construir um grafo de vizinhança de d e (2) calcular o grau de similaridade. No passo 1, páginas que estão conectadas a d são recuperadas. Constrói-se então o conjunto V, de vizinhos de d, que contém os pais de d, os filhos dos pais de d e os pais dos filhos de d. Este conjunto de páginas é relacionado a d.

No passo 2 é calculado o grau com que as páginas em V são relacionadas a d. Para tanto, considera-se as páginas em V e os *links* entre eles como um grafo chamado grafo de vizinhança de d. Este grafo é então processado pelo algoritmo HITS [15]. O algoritmo HITS retorna o grau de autoridade e centralização de cada página V. Intuitivamente, uma boa autoridade é uma página com informações importantes sobre um dado assunto. Um bom *hub* é a página que aponta para páginas que são boas autoridades. Na prática os graus de autoridade e *hub* são calculados recursivamente: a página é um bom *hub* se aponta para boas autoridades e é uma boa autoridade se é apontada por muitos outros bons *hubs*. Uma vez que o algoritmo HITS é aplicado, usa-se uma combinação do grau de autoridade e *hub* como uma medida de similaridade entre d e qualquer página que não está em V como zero.

2.2. Evidências Sintáticas

Três evidências sintáticas diferentes foram consideradas, a saber: evidência de texto do documento; evidência de título e evidência de texto da URL, que serão detalhadas nas seções seguintes.

2.2.1. Evidência de Texto do Documento

Uma evidência de texto é identificada se o conteúdo texto de dois documentos ligados por arestas apresentam alguma similaridade. A similaridade é computada utilizando o conceito de *shingles*, proposto por Broder [3]. O conjunto de *shingles* de tamanho k de um documento é o conjunto de todas as sequências diferentes de k termos do mesmo, considerando interseção de termos. Como exemplo: o conjunto de *shingles* do texto: "uma rosa é uma rosa é uma rosa" é: $\{$(uma, rosa, é, uma), (rosa, é, uma, rosa), (é, uma, rosa, é)$\}$, resultando em três *shingles* diferentes de tamanho 4.

Para identificar uma evidência de texto, todos os *shingles* de um determinado documento do grafo são inseridos em uma tabela *hash* [21] e alguns *shingles* de cada documento que possui aresta com o documento em questão são pesquisados na tabela. Quanto mais *shingles* encontrados maior é a evidência de que os textos sejam similares, ou ao menos tratem do mesmo assunto. Se todos os *shingles* são encontrados, a evidência de texto fica em 1.

2.2.2. Demais Evidências Sintáticas

Os títulos das páginas HTML foram identificados e agora são usados como um indício de similaridade entre documentos. A verificação é simples. Os termos do título de um documento são comparados com os termos dos títulos de todos os documentos que possuem aresta com o mesmo. Quanto maior o número de termos em comum, maior é o valor da evidência de título. Como exemplo, considere o grafo da figura 2-d). Se o título de A possui três termos, e dois termos do título de B são encontrados entre os três termos de A, o peso da aresta A–B fica acrescido de $0,66$. Se nenhum termo de C casa com algum termo de A, o peso da aresta A–C não é acrescido pela evidência de título. Nenhum outro vértice é comparado ao vértice A, pois não existem mais arestas em A.

Os termos contidos no endereço de URL das páginas também foram identificados e usados como indício de similaridade entre documentos. É considerado um termo todo o trecho de caracteres que estiverem entre barras. Por exemplo, para a URL:
http://www.dcc.ufmg.br/usuario/teste/index.html
temos três diferentes termos: usuario, teste e index.html. Quanto maior o número de termos na URL de um documento B que são encontrados na URL de um documento A, maior a evidência de texto da URL dos documentos, contribuindo com o peso da aresta A–B.

2.3. Evidência de Localização

A única evidência de localização considerada foi o domínio da Internet. Páginas hospedadas sob o mesmo domínio podem abordar assuntos semelhantes. Se as páginas A e B estão sob o mesmo domínio da Internet, a evidência de localização contribui para o peso da aresta A–B em 1. Caso contrário, não existe contribuição. Como exemplo, considere as páginas:
A – http://www.dcc.ufmg.br/usuario/teste/index.html
B – http://www.dcc.ufmg.br/usuario/teste/index.html
C – http://www.ufmg.br/novo/test/index2.html
Consideramos que as páginas A e B estão sob o mesmo domínio www.dcc.ufmg.br. No entanto, o domínio da página C não é o mesmo de A e B: www.ufmg.br.

3. Agrupamento de Dados

Uma vez construído o grafo G, deve-se agora dividi-lo em agrupamentos que representem conjuntos de páginas similares. Diferentes abordagens podem ser usadas neste processo de agrupamento de dados (do inglês *data clustering*). Jain, Murty e Flynn [13] apresentam e comparam diversas técnicas de agrupamento. A maioria dos algoritmos descritos não podem ser usados em grandes conjuntos de dados pois realizam operações computacionalmente caras. Na Tabela 1 tem-se as complexidades de tempo e espaço de algumas técnicas usadas em grandes conjuntos de dados, onde n é número de dados a serem agrupados, k o número de grupos e l o número de iterações. O método mais popular é o *k-means* pois normalmente k e l são fixos e o algoritmo tem tempo de execução linear. Apesar da complexidade de tempo linear, os agrupamentos gerados pelo *k-means* são bastante sensíveis à escolha do primeiro particionamento e ao valor de k. O algoritmo *ISODATA* é uma variação do *k-means* onde agrupamentos são criados e divididos com base em variâncias limite pré-definidas. Os algoritmos aglomerativos são mais flexíveis mas possuem complexidades de tempo $O(n^2 logn)$ e de espaço quadrática. Maiores detalhes sobre estes algoritmos e suas aplicações podem ser encontrados em [13].

Algoritmo	Complexidade de tempo	Complexidade de espaço
k-means	$O(nkl)$	$O(k)$
ISODATA	$O(nkl)$	$O(k)$
Aglomerativos	$O(n^2 logn)$	$O(n^2)$

Tabela 1. Complexidade de algoritmos de agrupamento

Além dos algoritmos citados na Tabela 1 existem as técnicas baseadas em teoria de grafos. Como vantagens destes algoritmos podem-se citar facilidade de implementação, independência da forma geométrica dos agrupamentos e melhor desempenho [19].

3.1. Agrupamento em Grafos Baseado na Árvore Geradora Mínima

Em [20] é proposto um algoritmo para agrupamento usando teoria de grafos. Primeiramente deve-se calcular a árvore geradora mínima (AGM) do grafo e depois remover arestas com valores maiores que um determinado limite, gerando assim uma série de agrupamentos. Apesar desta idéia ser antiga, algoritmos de agrupamento baseados na AGM têm apresentado bons desempenhos em trabalhos recentes [2, 5]. Um aspecto sensível deste algoritmo é como o valor limite é calculado. A Figura 3 ilustra o funcionamento deste algoritmo. A AGM do grafo é composta por 9 pontos. Através da remoção da aresta de maior peso CD, são formados dois agrupamentos, $\{$A,B,C$\}$ e $\{$D,E,F,G,H,I$\}$. Estes dois grupos representam os conjuntos que possuem similaridade.

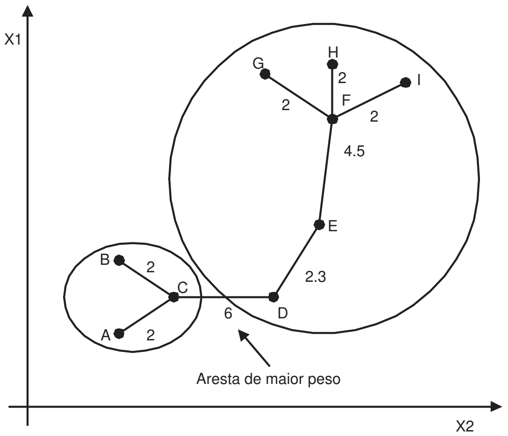

Figura 3. Geração de agrupamentos através da AGM

3.2. Algoritmo Implementado

A Figura 4 apresenta o algoritmo implementado. Este algoritmo é descrito em [10]. Primeiramente deve-se calcular a árvore geradora máxima (AGMáx) T do grafo G, pois no modelo adotado quanto maior é o peso da aresta, maior a similaridade entre as páginas. O próximo passo é calcular o limite L de peso para as arestas através da equação:

$$L = Media(T) + \alpha.\frac{Media(G) - Media(T)}{(Max(G) - Max(T)) + 1}$$

onde α é um fator de ajuste. Esta equação é sugerida em [10]. Outras equações podem refletir melhor a distribuição estatística do grafo G. Finalmente deve-se remover todas as arestas do grafo T com peso menor que L. Como resultado tem-se vários subgrafos que representam páginas similares.

```
Entrada: Grafo G
Saída: Partições P1, ..., Pn

Início
  Calcular a AGMáx T do Grafo G
  Calcular o limite L de peso das arestas
  Remover as arestas A onde peso(A) < L
  do Grafo T
  Usar cada conjunto conectado de
  componentes para formar as
  partições P1, ..., Pn
Fim
```

Figura 4. Algoritmo de agrupamento implementado

3.3. Análise de complexidade do algoritmo implementado

Para a geração da AGMáx sobre o grafo G usou-se o algoritmo de Prim disponibilizado em [21]. A complexidade de tempo do algoritmo de Prim é $O(|A|log|V|)$, onde A é o número de arestas e V o número de vértices do grafo G. O cálculo do limite L pode ser feito em $O(|A|)$. A geração das partições é feita em $O(|V| + |A|)$, mas como o grafo T é uma AGMáx, pode-se afirmar que $|A| = |V| - 1$ sendo portanto a sua complexidade de tempo $O(2|V| - 1)$ = $O(|V|)$. A complexidade de tempo total do algoritmo é $O(|A|log|V| + |A| + |V|)$ = $O(|A|log|V|)$.

A comparação desta complexidade de tempo com as complexidades de tempo dos algoritmos da Tabela 1 não é trivial pois as complexidades não estão em função dos mesmos parâmetros. Pode-se afirmar que o número de vértices V é igual ao número de dados a serem agrupados n. As complexidades algoritmos $k - means$ e $ISODATA$ também estão em função do número de iterações l e do número de agrupamentos k. Já a complexidade de tempo do algoritmo implementado depende também do número de arestas A.

4. Trabalhos Relacionados

De acordo com E. H. Han, G. Karypis, V. Kumar and B. Mobasher [12], agrupamento em Mineração de dados é um processo de descoberta que agrupa um conjunto de dados tal que a similaridade intra-grupo é maximizada e a similaridade inter-grupo é minimizada. Foi proposto um método para agrupamento de dados em um espaço de alta dimensão baseado no modelo de Hipergrafo. No modelo de hipergrafo, cada item de dado é representado como um vértice e itens de dados relacionados são conectados com hiperarestas rotuladas. Uma hiperaresta representa um relacionamento (afinidade) entre sub-conjuntos de dados e o rótulo (peso) da hiperaresta reflete a intensidade dessa relação. Um algoritmo de particionamento de hipergrafo é usado para encontrar um particionamento dos vértices. Um hipergrafo $H = (V, E)$ consiste de um conjunto de vértices V e um conjunto de hiperarestas E. Um hipergrafo é uma estensão de um grafo, no sentido de que cada hiperaresta pode conectar mais do que dois vértices.

No trabalho de J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan and A. S. Tomkins [15], as páginas *Web* e os *links* são vistos como vértices e arestas em um grafo direcionado, respectivamente. Vários outros trabalhos também utilizam esse modelo [9, 8, 16, 4, 11]. Kleinberg *et. all.* [15] observaram o seguinte fenômeno recorrente na *Web*: para qualquer tópico particular, há a tendência de existir um conjunto de páginas *autoridades* focadas naquele tópico, e um conjunto de páginas *hub* que contêm *links* para páginas úteis e relevantes naquele tópico. Essa observação motivou o desenvolvimento de dois algoritmos: o primeiro usa técnicas de mineração de dados para filtrar páginas de alta qualidade para um tópico em uma consulta e o segundo algoritmo para enumerar todos os tópicos que são representados na *Web*, em termos de conjuntos densos de tais páginas *hubs/autoridades*.

Em uma versão estendida de [15], R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins and E. Upfal apresentaram [17], na qual um algoritmo de classificação é utilizado. O problema de classificação supervisionado é: dado um conjunto de categorias predefinidas, construir um sistema que aprende com um conjunto exemplo e atribui um dado documento a uma dessas categorias. O algoritmo de classificação usado é chamado de HyperClass. A intuição é que páginas no mesmo tópico ou em tópicos relacionados tendem a serem apontadas mais frequentemente do que aquelas em tópicos não relacionados. O uso de *links* em conjunção com o texto melhora a precisão da categorização.

5. Conclusões e Trabalhos Futuros

Este trabalho apresentou um modelo para identificação de comunidades na *Web*, realizando agrupamento de documentos de assuntos relacionados. O modelo é composto de dois estágios: combinação de evidências para geração de grafo rotulado e agrupamento usando árvore geradora mínima para particionamento do grafo.

Como continuação do trabalho deve-se refinar os algoritmos de geração de pesos das arestas através do melhor ponderamento de cada evidência e/ou inclusão de novas evidências. Deve-se também investigar a melhor forma de gerar o limite de peso L para a remoção das arestas do grafo da AGMáx.

Agradecimentos

Agradecemos ao Professor Dr. Ricardo Baeza-Yates, da Universidade do Chile, pela sugestão e orientação do trabalho e ao aluno de doutorado Thierson Couto Rosa, pelas ajudas com relação à análise de *links* e disponibilização de seus programas.

Referências

[1] R. Amsler. Application of citation-based automatic classification. Technical report, University of Texas at Austin, Linguistics Research Center, Austin, Texas, USA, 1972.

[2] Ulrik Brandes, Marco Gaertler, and Dorothea Wagner. Experiments on graph clustering algorithms. In *Proceedings of the 12th Annual European Symposium on Algorithms, Lecture Notes in Computer Science 2832*, pages 568–579. Springer Verlag, 2003.

[3] A. Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29. IEEE Computer Society, 1998.

[4] Colin Cooper and Alan M. Frieze. A general model of undirected Web graphs. *Lecture Notes in Computer Science*, 2161:500–514, 2001.

[5] Steven R. Corman, Marco Gaertler, and Dorothea Wagner. Geometric mst clustering of text network collections. Presented at SUNBELT 2004, 2004.

[6] Marco Cristo, Pável Calado, Edleno Silva de Moura, Nivio Ziviani, and Berthier Ribeiro-Neto. Link information as a similarity measure in web classification. In *Proceedings of the String Processing and Information Retrieval SPIRE, Lecture Notes in Computer Science 2857*, pages 43–55. Springer Verlag, 2003.

[7] Jeffrey Dean and Monika R. Henzinger. Finding related pages in the world wide web. In *Proceeding of the eighth international conference on World Wide Web*, pages 1467–1479. Elsevier North-Holland, Inc., 1999.

[8] E. Drinea, M. Enachescu, and M. Mitzenmacher. Variations on random graph models for the web, 2000.

[9] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.

[10] Marco Gaertler. Clustering with spectral methods. Master's thesis, Universität Konstanz, 2002.

[11] Jean-Loup Guillaume and Matthieu Latapy. The web graph: an overview, 2002.

[12] Eui-Hong Han, George Karypis, Vipin Kumar, and Bamshad Mobasher. Hypergraph based clustering in high-dimensional data sets: A summary of results. *Data Engineering Bulletin*, 21(1):15–22, 1998.

[13] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[14] M.M. Kessler. Bibliographic coupling between scientific papers. American Documentation, 1963.

[15] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins. The Web as a graph: Measurements, models and methods. *Lecture Notes in Computer Science*, 1627:1–18, 1999.

[16] Kumar, Raghavan, Rajagopalan, Sivakumar, Tomkins, and Upfal. Stochastic models for the web graph. In *41th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 57–65, November 2000.

[17] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. The Web as a graph. In *Proc. 19th ACM SIGACT-SIGMOD-AIGART Symp. Principles of Database Systems, PODS*, pages 1–10. ACM Press, 15–17 2000.

[18] H.G. Small. Co-citation in the scientific literature: A new measure of relationship between two documents. *Journal of the American Society for Information Science*, 24:265–269, 1973.

[19] Nora Speer, Peter Merz, Christian Spieth, and Andreas Zell. Clustering gene expression data with memetic algorithms based on minimum spanning trees. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, pages 1848–1855. IEEE Press, 2003.

[20] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 20(1):68–86, 1971.

[21] Nivio Ziviani. *Projeto de Algoritmos com Implementações PASCAL e C*. Pioneira Thomson Learning, 2004.