Iuliana G. S. Rodrigues,
Pedro Shiguihara-Juárez,
Jorge Valverde-Rebaza

Mineração de Links

– Seminário da disciplina de Redes Complexas –

ICMC - Universidade de São Paulo

02 de Junho, 2011

Sumário

1

Introdução

A descoberta de *Data Mining* assim como o *Link Mining* surgiu como muitos outros desafios na tentativa de recuperar informações de uma vasta quantidade de dados. À medida que a população aumenta, a estimativa é de que haja em torno de 150 milhões de páginas web com a duplicação em menos de um ano (Page e Brin, 2005), tornando assim, mais complicado a tarefa de recuperação de informação.

Os conjuntos de dados são descritos como coleções ligadas de objetos inter-relacionados que podem ser simples redes homogêneas (redes sociais ou a própria web) ou ricas redes heterogêneas (domínio médico - pacientes, paciente - doenças, doença - tratamentos e paciente - contatos) com múltiplos tipos de objetos ligados. *Link Mining* é a técnica de *data mining* que considera explicitamente esses *links* quando se constroem modelos preditivos ou descritivos desses dados inter-relacionados (Gettor et al, 2003).

Existem várias tarefas desse modelo de definição de dados, como ranking de objetos detecção de grupos, classificação de coleções , predição de *links* e descoberta de subgrafos. Levando em consideração as ligações com *links*, padrões mais complexos podem surgir. Isto leva a outros estudos focados em descoberta de subestruturas tais como grupos, comunidades ou subgrafos comuns.

Um dos principais desafios para a mineração de dados é combater o problema de dados estruturados em conjuntos heterogêneos. Esses tipos de conjuntos de dados são melhor descritos como redes ou gráficos. Os domínios geralmente consistem em uma variedade de objetos. Os objetos podem ser ligados em uma variedade de maneiras. Assim, o gráfico pode ter diferente nós e diferentes tipos de aresta. Podem ser aplicados em procedimentos tradicionais de inferência estatística, que assume que as instâncias são independentes, podendo levar a conclusões inadequadas sobre os dados (Jensen, 1999). Contudo, pode ser utilizado para tentar predizer se o *link* vai existir no futuro baseado o *link* previamente observado.

O trabalho da mineração de *link* consite em análise de *links*, redes semanticâs, lógica de programação indutiva, teoria de grafos, análise de redes sociais e descoberta de conhecimento (Jensen et al, 1998).

No intuito de desenvolver algumas dessas tarefas, tem-se como intuito pesquisar dentro de relevantes comunidades dados de representação para a mineração de *links* na tentativa de definir algoritmos de mineração visando encontrar padrões em mineração de *links*. Algumas dessas tarefas itens serão descritos em subsecções posteriores. Na Secção 2 são amostradas las tarefas relacionadas com objetos. Na Seção 3 são amostradas las tarefas relacionadas com links. Na Secção 4 são amostradas as tarefas relacionadas com grafos. Na Seção 5 são amostradas nossas conclusões.

2

Tarefas com Objetos

O problema da mineração em conjuntos de dados, onde os objetos estão ligados de alguma forma, é um novo desafio para o aprendizado de máquina. Em muitos casos, estes dados podem ser descritos por um gráfico, por links e as arestas entre os objetos.Pode ser útil para muitas tarefas de aprendizagem e são geralmente difíceis de capturar usando modelos estatísticos tradicionais. Os objetos podem ser rotulados ou não-rotulados e sua classificação deve explorar a estrutura e a ligação dos dados entre ambos os rotulados e não rotulados objetos.

2.1 Link baseado no Ranking de objetos

2.1.1 Definição

É uma tarefa do *link mining* utilizada para explorar a estrutura de um grafo para ordenar ou priorizar um conjunto de objetos dentro de um grafo. A maioria desses objetos é utilizada considerando grafos com tipos simples de objetos e tipos simples de links. No contexto da recuperação de informação na web, vários algoritmos são utilizados para utilização de ranking. *PageRank* e HITS são as abordagens mais notáveis.

Exemplo

Análise das relações e trocas de informações entre pessoas, grupos, organizações, computadores ou quaisquer entidades capazes de processar informações e/ou conhecimento Intimamente relacionada com Link Mining. Os Sites de redes sociais, Figura 2.1, também oferecem ricas fontes de dados comportamentais. Perfil e interligação dos dados da SNSs - *Social Network Sites* pode ser obtida através da utilização de técnicas de coleta mecanizada ou através de conjuntos de dados fornecidos diretamente da empresa, permitindo aos investigadores de análise de rede para explorar os padrões de grande escala

de amizade, extendendo o uso, e continuando uma tendência que começou com a análise de exames de blogs e outros sites. Por exemplo, Golder, Wilkinson e Huberman (2007) [50] analisou um conjunto de dados anônimos constituído de 362 milhões de mensagens trocadas por mais de quatro milhões de usuários do Facebook 2.2 para compreensão em redes de amizades e atividades de troca de mensagens. Lampe, Ellison e Steinfield (2007) [51] explorou a relação entre elementos do perfil e do número de amigos do Facebook, considerando que os campos de perfil que reduzem os custos de transação e são mais difíceis de falsificar é mais provável ser associado com maior número de laços de amizade.

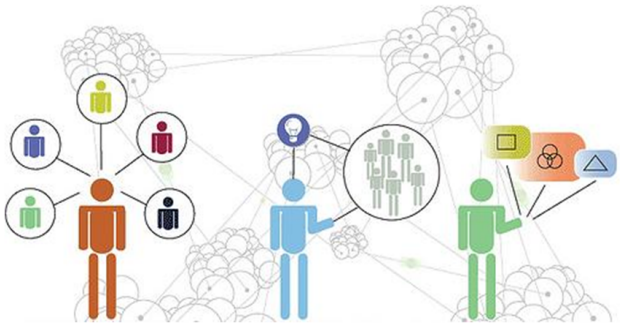


Figura 2.1. Essa figura representa a generalização de Redes Sociais

Figura 2.2. Figura simbólica representando a rede do Facebook

2.2 Link baseado em classificação de objetos

Recentemente um grande volume de estudos têm sido realizados para processamento de dados (por exemplo, dados da rede social e hiperlinks da web), em parte devido à grande quantidade de dados que estão sendo disponibilizados.Uma tarefa popular na mineração é baseada em classificação de link, ou seja, classificação das amostras utilizando as relações ou links que estão presentes entre eles. Para [13], várias abordagens essa propostas tem sido classificada.

2.2.1 Definição

O objetivo da classificação é detectar os membros de um conjunto composto de objetos conectados em um conjunto finito de valores categóricos [1]. O procedimento de classificação dos objetos vai depender do algoritmo que se está usando, em síntese e de modo mais simples, a classificação se dará com a indicação de segmentos para determinadas classes, sendo esses usados como amostras.

O que faz a Classificação de objetos ser diferente da classificação tradicional é que em muitos casos, o rótulo dos objetos relacionados tendem a ser correlacionados. O desafio é construir algoritmos para classificação coletiva que exploram tais correlações e conjuntamente e infere os valores categóricos associados com os objetos no gráfico [1].

Exemplo

A classificação pode ser muito mais apurada quando se faz um estudo das estatísticas dos grafos para identificação dos grafos pretendidos. Geralmente, nesses algoritmos, usam-se os classificadores de pertinência para associar os objetos as classes, esses classificadores inibem a noção determinista de sim ou não, deixando com que as estatísticas de cada segmento definam o grau de preenchimento a uma determinada classe. O principio da segmentação é, que partindo de uma imagem digital, possamos através de algoritmos estatísticos, reduzir as informações da mesma, em regiões homogêneas nesse grafo (objetos) 2.3, as quais são funções diretas do problema considerado, ajudando em uma análise mais adequar da imagem.

Desafios

1. Dinamismo das necessidades do usuário;
2. Grafos em constante mudança;
3. Combinar técnicas;
4. Análise de grafos gigantescos;
   E muitos outros.

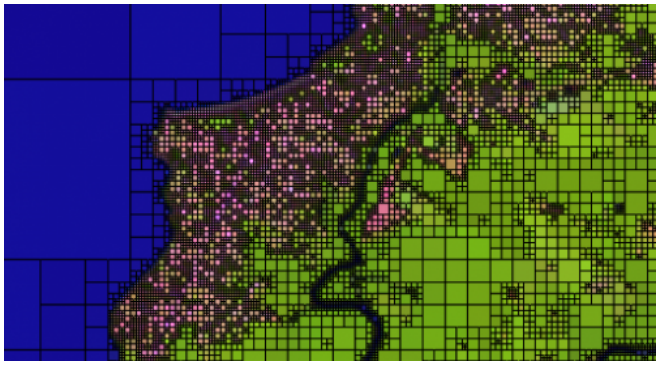


Figura 2.3. Figura de segmentação de regiões utilizando algoritmos estatísticos para a classificação da mesma.

2.3 Detecção de grupo

O objetivo é extrair conhecimento valioso que podem apresentar padrões ocultos [26]. O sucesso das aplicações esta em descobrir estruturas ocultas de organizações, a identificação de conduta fraudulenta, e extrair as atividades de um grupo [26][28][29][31]. A descoberta de membros de grupo não conhecidos e identificação completa de grupos não conhecidos é um gap nesta area [31]. Getoor [**?**] diz que a deteção de grupo tem como objetivo fazer uma *clusterização* dos nós no grafo em grupos que compartilham características comuns.

Segundo Wang et. al [27], a tarefa de detecção de grupo tem duas fases:

1. Criar grupos por meio de uma semente
2. Expandir os grupo existente por teste de potenciais membros

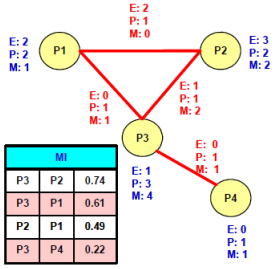
Segundo Wasserman et. al. [30], a detecção de grupo tenta detectar subgrupos coesos sobre os quais há uma relação forte, direta, intensa e frequente. As aplicações da detecção de grupos é variada:

- Mineração de dados
- Análise de Redes Sociais
- Teoria de grafos

2.3.1 Análise de detecção de grupo em Redes Sociais

A detecção de grupo esta muito relacionada a Análise de Redes Sociais (*Social Network Analysis - SNA*), onde a detecção do grupo não só tem que ver com os atributos comuns dos membros do grupo, senão que também com as interaçãoes entre eles. Assim, a comunicação de membros de grupo, a freqüência da comunicação, as transações comerciais entre eles e as relações familiares entre outras coisas, são avaliadas pela detecção de grupo. Estritamente, a avaliação tem que ver tanto com os atributos das entidades como as propriedades

dos links. Trabalhos neste sentido, tem sido publicados durante os últimos anos, como Jennifer J. Xu and Hsinchun Chen [29] que estabelecem um *framework* para a análise de redes criminosas. Ozgul [28] analisa um caso de detecção de um grupo criminoso em Turquia, na cidade de Bursa também.



Time	Action	P1	P2	P3	P4
1	P1 Email P2	E	E	Φ	Φ
2	P3 Phone Call P4	Φ	Φ	P	P
3	P2 Meeting P3	Φ	M	M	Φ
4	P1 Email P2	E	E	Φ	Φ
5	P2 Email P3	Φ	E	E	Φ
6	P1 Phone Call P3	P	Φ	P	Φ
7	P1 Phone Call P2	P	P	Φ	Φ
8	P2 Meeting P3	Φ	M	M	Φ
9	P3 Meeting P4	Φ	Φ	M	M
10	P1 Meeting P3	M	Φ	M	Φ
11	P2 Phone Call P3	Φ	P	P	Φ

Figura 2.4. Obtenção de MI entre duas variáveis. Onde E=email, M=messages, P=phone

A característica principal das últimas pesquisas é que, os métodos desenvolvidos tem que ser escaláveis, desta forma ter eficiência e possam explotar incrementalmente grafos complexos para a obtenção e descoberta de conhecimneto. [26][28]. Por exemplo Adibi et. al.[31] propuseram um localizador de grupo chamado: *KOJAK Group Finder*, em que um posicionamento dos grupos era fixada e logo era expandida utilizando técnicas baseadas em conhecimento para acrescentar mais candidatos, segundo das interações que mostram possíveis associações. Para medir a dependência entre duas variáveis, Adabi et. al. [31] utilizavam a medida de *Mutual Information (MI)*. O trabalho de Adabo et. al.[31] se mostra na figura 2.3.1.

2.4 Resolução de entidade

Uma entidade poder ser cualquer objeto do mundo real: pessoas, lugares, coisas entre outras. Logo, a resolução de entidade é o processo de determinar se dois referências do mundo real estão se referendo ao mesmo o diferentes objetos [32]. No contexto das bases de dados, a resolução de entidade é o processo de identificar quais registros numa base de dados refere-se à mesma entidade do mundo real, o que envolve medir a similaridade entre cada par de registros [33], que pode ser muito caro para grandes conjuntos de dados. Assim, as técnicas de *blocking* são muito utilizadas para melhorar o desenvolvimento da resolução de entidade, dividindo os registros em blocos de múltiplas formas e comparando só os registros no mesmo bloco [33]. O problema de isso, é que, não se tem em conta a informação dos demais blocos, para o que mais técnicas são implementada para tratar esse novo problema dos blocos.

A resolução de entidade trata os problemas como a duplicação e integração de dados [1]. Algumas aplicações de resolução de entidade são citadas por Euijong [33] e Chen [36] e as listamos aqui:

Mailing list: listas de emails que podem ter o mesmo endereço físico, mas cada registro pode variar ligeiramente, como a ortografia ou falta de informação
Empresas: duas empresas que querem misturar seus dados de clientes. Mas, utilizar uma resolução de entidade exaustiva envolve a comparação de todos os pares de registros, o que pode ser muito caro.
Citações: Se existe uma pessoa *John Smith* e *Jane Smith*, e utilizam a descrição: *J. Smith*, aconlleva à ambigüidade.

2.4.1 Definição

Uma definição da resolução de entidade [36] é: seja D um conjunto de dados, que tem um conjunto de entidades $E = \{e_1, e_2, \ldots, e_m\}$ e as relações em que participam. Entidades podem ser de diferentes tipos e as entidades em D são representadas por um conjunto de instâncias $R = \{r_1, r_2, \ldots, r_n\}$ referidas como representações de entidades ou referências. O objetivo é agrupar corretamente as representações em R co-relacionadas, isto é, que se refere à mesma entidade.

2.4.2 Considerações da resolução de entidade

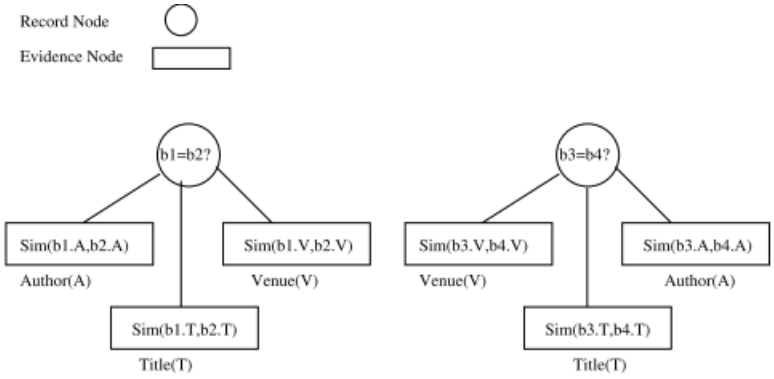
O abordagem poder ser visto segundo a quantidade de registros que são comparados. No princípio, era comum utilizar dois registros para fazer a comparação, uma e outra vez até ter resolvido o problema da resolução de entidade. Neste caso cada comparação era resolvida de forma independente das outras [33]. Em outros casos, os registros tem sido resolvidos coletivamente, o que tem obtido melhores resultados [34]. Neste sentido, os modelos *coletivos* fazem uma *clusterização* dos nós [34], então, não só é considerada a distância entre dois nós, senão também, a distância entre os nós e seus vizinhos em seus *clusters*. A figura 2.4.2 mostra uma comparação das duas abordagens feitas por Culotta et. al [34].

Então, como se explicou ao início desta seção, a resolução de entidade pode ter duas abordagens desde o ponto de vista da análise: por *clusterização* ou independente para cada par de registros analizados. Domingos [35] diz, que a diferença entre o modelo independente e o coletivo, é que o modelo coletivo não faz decisões independentemente senão que haz decisões coletivas para todos os pares de candidatos, propagando a informação através de atributos que compartilham os pares, pelo que é uma decisão mais informada sobre asociações potenciais. Para isto, ele utilizara *Conditional Random Fields* para seu modelo coletivo. Além, as técnicas de resolução de entidade independente são métodos de similaridade baseados em características (*feature-based-similarity-FBS*) [35].

		Paper		Venue	
		indep	**joint**	**indep**	**joint**
Citeseer	**constraint**	88.9	**91.0**	79.4	**94.1**
	reinforce	92.2	92.2	56.5	**60.1**
	face	88.2	**93.7**	80.9	**82.8**
	reason	**97.4**	97.0	75.6	**79.5**
	Micro Avg.	91.7	**93.4**	73.1	**79.1**
Cora	**kibl**	92.9	**93.3**	93.6	**99.3**
	fahl	**95.5**	95.0	87.3	**99.7**
	utgo	79.9	**84.0**	51.7	**60.4**
	Micro Avg.	89.4	**90.8**	77.5	**84.5**

Figura 2.5. Tabela que mostra os experimentos de Culotta et. al [34] utilizado a resolução de entidades por *clusters* e de forma independente

A forma em que trabalha o modelo independente e o modelo coletivo descrita por Domingo [35], se mostra na figura 2.4.2.



(a) Each pairwise decision considered independently

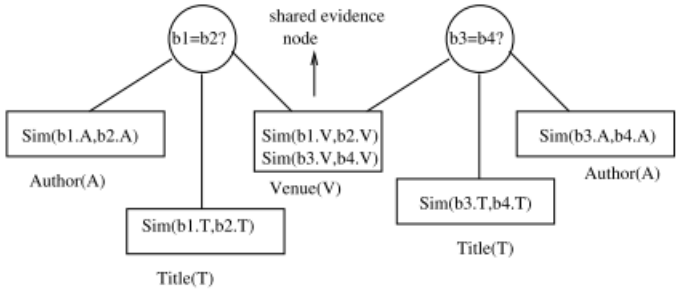


Figura 2.6. Forma em que trabalha o modelo independente e o modelo coletivo [35]

2.4.3 Abordagens da Resolução de entidade

A proposta de Bhattacharya e Getoor [37] dos abordagans da resolução de entidade são: baseados em atributo, naive relacional e resolução de entidade coletiva.

Resolução de entidade baseado em atributo

É a abordagem tradicional, onde a similaridade $sim_A(r_i, r_j)$ é calculado para cada par de referências r_i e r_j baseados em seus atributos. Só os pares que tem similaridade sobre um limiar, são considerados co-referentes. Muitas medidas tem sido desenvolvidas como: Jaro, Levenstein, Jaro-Winkler, e TF-IDF.

Resolução de entidade por Naive relacional

A forma mais simples para usar relações para a resolução de entidades é tratar referências relacionadas como atributos adicionais para a associação. Por exemplo, para determinar se dois referências de autores são co-referentes, se faz uma comparação dos nomes de seus coautores.

Resolução de entidade coletivamente

Dada uma medida de similaridade entre pares de referências, o objetivo é fazer uma *clusterização* das referências, de modo que só as referências que pertencem à mesma entidade sejam atribuídas ao mesmo *cluster*.

3

Tarefas com Links

Getoor [1] estabelece uma única tarefa com links chamada: *predição de links*. A *predição de links* tem tido um desenvolvimento cada vez maior nos últimos anos, e isto é, pelo reconhecimento de sua importância nas interações nos objetos ou instâncias em redes complexas de diferentes tipos. De esta forma, neste capítulo apresentamos a *predição de links* como principal e único tema de estudo. Portanto, tudo o conteúdo mostrado aqui pertence estritamente a este assunto. Na primeira seção se mostra a definição do problema de predição para links em uma rede complexa, em seguida, na seção dois se encontra os abordagens comuns utilizados para poder fazer a predição de links, e finalmente se apresentam alguns exemplos e comentários finais acerca deste problema.

3.1 Predição de links

Dado um grafo, a predição de links está focada em relações de arestas, onde o objetivo é a predição da existência de um link entre dois entidades ou vértices, tendo em conta seus atributos e outros links observados [1]. As aplicações de este tipo de tarefa é diversa: predições de amigos entre pessoas que ainda não tem sido ligadas em redes sociais, ou predições de seus participações em eventos como em redes de co-autoria, e-mail ou ligações telefônicas.

3.1.1 Definição

A predição de links está baseado no espaço temporal. Assim, novos links serão estabelecidos ou serão excluídos após a passagem de un tempo t, no tempo seguinte: $t+\alpha$, onde $\alpha \in Z$. O problema pode ser definido como um problema binário [1], onde se deseja obter informação de uma possível ligação no espaço temporal entre as entidades: O_i e O_j.
Seja G = (V,E), onde $O_i, O_j \in G$, e seja $\ell_{i,j}$ uma aresta entre os objetos O_i e O_j, e dado um *snapshot* da rede e seus links no tempo t, logo, fazer uma predição dos links no tempo $t+1$.

$$\ell_{i,j} = \begin{cases} 1, \ O_i \ ligado \ a \ O_j \\ 0, \ outro \ caso \end{cases} \tag{3.1}$$

3.2 Considerações na classificação da predição de links

A predição de links pode-se classificar segundo vários critérios. O problema pode ser classificado segundo se os links existem ou não existem [43]. De esta forma, é possível fazer a predição de links que existem mas não conhecemos de seu existência, como webs de refeições, redes de interações de proteína-proteína e redes metabólicas; por outro lado, se pode fazer predições de links que não existem mas no futuro na evolução das redes podem existir. Também pode ser classificado segundo as informações que se utilizam para fazer a predição, como se apresentará na seção 3.3. Com respeito à classificação baseada nas informações estruturais da rede, ésta pode ser subdividida segundo as medidas baseadas em: os vizinhos ou nas rotas.

3.3 Abordagens da predição de links

Existe uma quantidade diversa de abordagens para fazer a predição sobre os links de um grafo. Uma abordagem pode ser a predição de um link baseado somente nas propriedades estruturais do grafo [1], como as medidas de proximidade. Outros abordagens fazem uso da informação dos atributos para a predição de um link, como as características relacionais definidas numa base de dados. É possível fazer uma abordagem utilizando tanto as propriedades estruturais como os atributos dos nós. Além, pode considerar-se a toda a rede como um só modelo probabilístico, desta forma, se pode calcular a distribuição conjunta ($P(E)$) sobre o conjunto de arestas E, ou uma distribuição condicionada sobre os atributos dos nós $P(E|X)$ [1]; assim, o nível de incerteza para a predição de links pode ser utilizado também. Em este sentido, consideramos o dito anteriormente e apresentamos uma tabela com tipos de abordagens segundo o tipo de informação utilizada do grafo 3.1:

Destas abordagens, existem ainda muitas limitações, tais como menciona Getoor [1]:

- A ligação de predições é muito dificil porque os dados ligados estão esparsos
- A construção de modelos probabilísticos é muito pequena, pelo que é difícil a avaliação do modelo e a quantificação do nível de confiança nas predições

Para isto, Rattigan e Jensen [1] estabelecem uma discussão sobre os novos desafios baseados nestos problemas, em que a qualidade das predições pode melhorar ao fazer predições coletivamente. No caso dos modelos probabilísticos, a utilização de todo o grafo dentro do modelo probabilístico, permite capturar as correlações sobre os links, pelo que permite predições adequadas

Abordagem	Autor	Tipo especifico
Estruturais	Liben-Nowell e Kleinberg	Medidas de proximidade em diferentes grafos
Atributos	Popescul et. al	Modelo de regressão logística para fazer uso de características relacionais definidas por consultas de bases de dados
Estruturais e de atributos	O'Madadhain et. al.	Construção de modelos de probabilidade condicional
Probabilísticas	Zhu et. al. [39]	Modelos de Markov

Tabela 3.1. Algumas abordagens utilizadas para a predição de links descritas por Getoor [1]

mas tem um custo computacional alta quando se utiliza tuda a rede; por isto, é preciso utilizar técnicas de inferência aproximadas [1].

3.3.1 Abordagem estrutural

A abordagem estrutural esta baseado nas informações que se podem obter da rede, utilizando para isto, medidas em função dos vizinhos de um nó ou em rotas. Na seção 3.3.1 mostramos algumas medidas utilizadas para quantificar a configuração da estrutura da rede; na próxima seção (3.3.1) fazemos a definição do problema da abordagem estrutural e seguidamente alguns considerações sobre esta abordagem e um exemplo que mostra como é utilizada.

Medidas de similaridade

As medidas podem ser classificadas baseadas em seus vizinhos (A) ou nas rotas (B) [45][44], como se apresenta a continuação:

A1. Common Neighbors: o número de vizinhos comuns de x e y ($|\Gamma(x) \cap \Gamma(y)|$) pode representar a similaridade de nós.

A2. Jaccard's Coefficient: mede o número de vizinhos de x e y comparado ao número de nós que são ou vizinhos com x ou vizino com y: $\frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$

A3. Adamic/Adar: mede as características que os objetos compartilham e é definida por: $\sum_{z:caract.dex,y} \frac{1}{log(frequency(z))}$.

B1. Distância do grafo: está definida como a distância do caminho mais curto entre pares de nós no grafo.

B2. $Katz_\beta$*:* está definida com a suma dos pesos de todos os caminhos entre dois nós: $\sum_{\ell=1}^{\infty} \beta^\ell . \left|paths_{x,y}^{<\ell>}\right|$, onde $paths_{x,y}^{<\ell>}$ é o conjunto de distâncias dos caminhos de x até y.

Definição

Tendo a definição da predição de links mostrada na equação 3.1, consideramos algumas novas coisas na definição para focar a abordagen estrutural nesta

seção. Lembramos que o grafo G pode ter uma evolução se seus interações mudam com o tempo [38], assim temos novamente a definição da predição de links além considerando *clusters*.

Seja $G = (V, E)$ um grafo que muda com o tempo; onde V são entidades únicas e E são as interações totais que existem sobre esses entidades. Um *snapshot* do grafo G é: $S_i = (V_i, E_i)$ que é um grafo representando só entidades e interações ativas num intervalo de tempo particular $[T_{s_i}, T_{e_i}]$. Na figura 3.3.1 se observa a evolução de um grafo em dois intervalos. No primeiro intervalo existe uma interação entre A e C e entre A e D. No segundo intervalo esses interações já não existem mais.

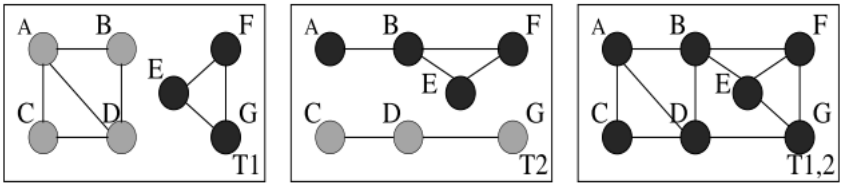


Figura 3.1. Evolução de um grafo G, onde os dois primeiros grafos são *snapshots* em tempo t_1 e t_2 respectivamente, e a seguinte *snapshot* é uma junção dos dois grafos anteriores [38]

Clusters no grafo

Para estudar a evolução do grafo é necessária uma representação de seu estrutura em diferentes *snapshots*. Para alcançar isto, Asur [38] considera *clusters* para cada *snpashot* do grafo, portanto, cada S_i é particionado em k_i comunidades ou *clusters*: $C_i = \left\{C_i^1, C_i^2, \ldots, C_i^{k_i}\right\}$. Na figura 3.3.1 se observa dois clusters marcados com dois cores diferentes utilizando o algoritmo *MCL* [38].

Medida estrutural $\kappa - Merge$

Utilização de uma medida estrutural proposta por Asur [38], chamada $\kappa - Merge$. Sejam dois *clusters* diferentes: C_i^k e C_i^l a serem mesclados só se existe um *cluster* na seguinte *snapshot* que contem como mínimo $\kappa\%$ dos nós pertencentes aos dois *clusters*.

$$Merge(C_i^k, C_i^l, \kappa) = 1, \iff \exists C_{i+1}^j, talque, \frac{\left|\left(V_i^k \cup V_i^l\right) \cap V_{i+1}^j\right|}{Max\left(\left|V_i^k \cup V_i^l\right|, \left|V_{i+1}^j\right|\right)} > \kappa\% \tag{3.2}$$

Na figura 3.2 se mostra um exemplo em que a medida $\kappa - Merge$ (definida na ecuação 3.2) é aplicada no quadro t)3. Para este caso, a porcentagem de nós foi de $\kappa = 100$, porque todos os nós formam um novo *cluster*.

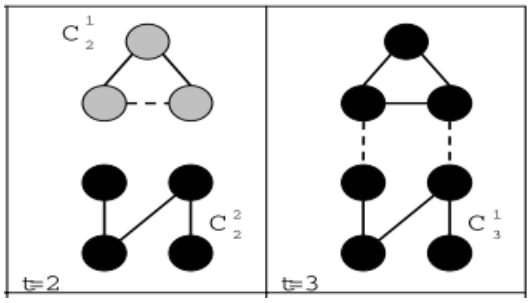


Figura 3.2. Junção de dois *clusters* no próximo quadro [38]

Exemplo: Co-autoria

Asur [38], utiliza uma rede de co-autoria DBLP. Artigos publicados em 28 conferências importantes durante 10 anos (1997-2006). Considera-se um *snapshot* em um ano.

Ano 2005	
4*Cluster 1	(AAAI 2005) Niels Landwehr, Kristian Kersting e Luc De Raedt: nFOIL: Integrating Naïve Bayes and FOIL (AAAI 2005) Luc De Raedt, Kristian Kersting e Sunna Torge: Towards Learning Stochastic Programs from Proof-Banks
4*Cluster 2	(ICML 2005) Sauro Menchetti, Fabrizio Costa e Paolo Frasconi: Weighted Decomposition Kernels (IJCAI 2005) Andrea Passerini e Paolo Frasconi: P. Kernels on Prolog Ground Terms
Ano 2006	
Merged cluster	(ILP 2006): Niels Landwehr, Andrea Passerini, Luc De Raedt, Paolo Frasconi: kFOIL: Learning Simple Relational Kernels

Tabela 3.2. *Snapshot 2005 e 2006 de uma rede de co-autoria*

Um resultado obtido por Asur [38] na rede de co-autoria foi em dois *snapshots* anos: 2005 e 2006, apresentados na tabela 3.2. A conclusão dos resultados foi que a proximidade dos autores e a similaridade dos tópicos de pesquisa podem determinar uma nova relação entre os autores.

No caso da rede de colaboração, dois nós são *clusterizados* numa junção se eles trabalham em artigos relacionados ou pertencem ao mesmo grupo de trabalho.

3.3.2 Abordagem probabilístico

No contexto dos modelos probabilísticos para a predição de links, os modelos de Markov tem sido muito utilizados [42][41][40][39]. Um dos temas muito abordados para a utilização deste abordagen é a predição de links para recomendação aos usuários na web para visitar rotas que usuários passados também tem visitado.

$l=1$		$l=2$		$l=3$	
x_i	w_i	$x_i \rightarrow x_j$	w_{ij}	$x_i \rightarrow x_j$	w_{ij}
a	4	$a \rightarrow b$	2	$a \rightarrow b \rightarrow c$	1
b	5	$a \rightarrow c$	1	$a \rightarrow b \rightarrow e$	1
c	4	$b \rightarrow c$	3	$a \rightarrow c \rightarrow d$	1
d	3	$b \rightarrow e$	1	$b \rightarrow c \rightarrow b$	1
e	1	$b \rightarrow g$	1	$b \rightarrow c \rightarrow d$	1
f	2	$c \rightarrow b$	1	$b \rightarrow c \rightarrow f$	1
g	1	$c \rightarrow d$	1	$b \rightarrow e \rightarrow d$	1
		$c \rightarrow f$	1	$c \rightarrow b \rightarrow g$	1
		$d \rightarrow f$	1	$c \rightarrow d \rightarrow f$	1
		$e \rightarrow d$	1	$c \rightarrow f \rightarrow a$	1
		$f \rightarrow a$	1		

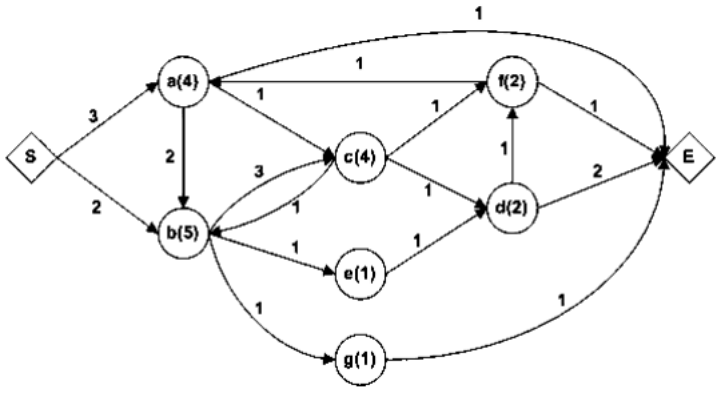


Figura 3.3. Modelo de Markov para a navegação dos usuários na web [40]

Em Zhu et. al [39], se modela o comportamento da navegação de usuários na web. Se utiliza *clusters* para estabelecer relações entre sites web e construir uma hierarquia da web. O modelo de Markov é utilizado para fazer predições na hierarquia para ajudar à navegação dos usuários na web. Em Eirinaki et.

al. [40] o modelo de Markov é utilizado em conjunto com o algoritmo de Page Rank para recomendação personalizada para cada usuário na navegação na web. Na figura 3.3.2 se pode observar um Modelo de Markov, onde os números em parênteses nos nós são a quantidade de visitas à site web, além, os pesos das arestas são o número de vezes que o link foi visitado. Os nós S e E são os nós de início e fim respectivamente.

4

Tarefas relacionadas com Grafos

Nos último anos, foi desenvolvida uma grande variedade de técnicas de aprendizado de máquina, as quais têm sua própria representação do conhecimento, sendo a representação através de redes uma das quais tem uma grande atenção em tarefas de mineração de dados. Como é sabido, uma rede pode ser definida como um conjunto de itens conectados por relações existentes entre eles, podendo ser representada por grafos, nos quais os itens são os vértices e suas conexões são as arestas.

A representação por grafos é possível devido a que os grafos são estruturas de dados de caracter universal que permitem o modelagem de diferentes entidades e suas relações. Assim, os algoritmos de mineração de grafos permitem o incremento ou entendimento das informações representadas pelos conjuntos de dados representados em grafos, os quais podem formar estruturas muito complexas.

As informações que os grafos têm nas arestas conectando os vértices e muita, e é possível fazer tarefas de mineração nelas. Na mineração de *links*, as tarefas relacionadas com grafos tem que ver com o trabalho utilizando a estrutura de grafos com base nas informações de suas arestas, além de seus vértices. Em [1] as tarefas de mineração de *links* relacionadas com grafos e que apresentamos neste capítulo são: a descoberta de sub-estruturas (Seção 4.1), a classificação de grafos (Seção 4.2), e a geração de modelos para grafos (Seção 4.3).

4.1 Descoberta de Sub-estruturas

Um sistema de descoberta de sub-estruturas representa dados estruturados na forma de um grafo rotulado. Os objetos do conjunto de dados são os vértices do grafo e as relações entre eles são as arestas do grafo, as quais podem ser dirigidas o não dirigidas. Um sub-grafo é a parte mínima de um grafo, ou seja, só um vértice. Uma sub-estrutura é uma conexão de sub-grafos [2].

A descoberta de sub-estruturas é o processo de identificar conceitos que representam interessantes e repetitivas sub-estruturas de um grafo. Uma vez feita a descoberta, a sub-estrutura pode ser usada para simplificar o conjunto de dados substituindo as instâncias da sub-estrutura com um ponteiro para a uma nova e única instância da sub-estrutura. Além disso, a descoberta de sub-estruturas permite a abstração da estrutura detalhada do conjunto de dados original e fornece atributos novos e relevantes para a interpretação dos dados [2].

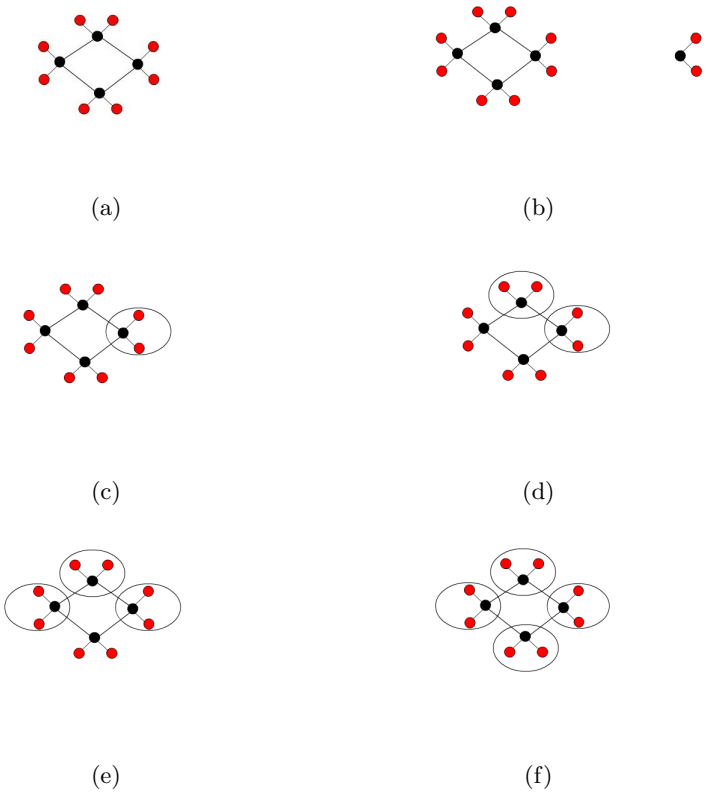


Figura 4.1. O processo de descoberta de sub-estruturas num grafo. Em (a), amostra-se um dado grafo. Em (b) é descoberta uma sub-estrutura padrão que satisfaz a especificação de ser a maior sub-estrutura do grafo. Em (c), a sub-estrutura padrão é encontrada na parte direita do grafo. Em (d), a sub-estrutura padrão é encontrada na parte superior do grafo. Em (e), a sub-estrutura padrão é encontrada na parte esquerda do grafo. E, em (f), a sub-estrutura padrão é encontrada na parte inferior do grafo.

Em geral, o processo de descoberta de sub-estruturas começa com a descoberta de alguma sub-estrutura padrão que poda atender alguma especificação precisa, isso é o principal trabalho nesta tarefa da mineração de *links*. Tendo a sub-estrutura padrão, o seguinte é fazer seu busca no resto do grafo. Na figura 4.1 amostra-se o processo da descoberta de uma sub-estrutura padrão e seu subseqüente busca no resto do grafo.

4.1.1 Abordagens

Muitas abordagens foram feitas para a descoberta de sub-estruturas em grafos. O programa *ARCH* [3] faz a descoberta de sub-estruturas com o objetivo de aprofundar descrição hierárquica de um cenário e para um grupo de objetos dentro de conceitos mais geral. O programa *ARCH* busca dois tipos de sub-estruturas num domínio de blocos de palavras. O primeiro tipo envolve uma seqüência de objetos ligados por uma cadeia de ligações similares. O segundo tipo envolve um conjunto de objetos dos quais, cada um tem uma relação similar para algum agrupamento de objetos.

No [4] desenvolve-se um sistema para armazenar grafos rotulados, sendo que cada grafo é representado por um conjunto de vértices num grafo universal. No [5] descreve-se um sistema para armazenar grafos utilizando um modelo de grafos probabilístico para a representação de um grafo.

Em [6] descreve-se o *AGM*, um algoritmo que aproveita a probabilidade a priori dos nós do grafo para encontrar todas as sub-estruturas do grafo satisfazendo o suporte mínimo. Em [7] o algoritmo *AGM* é melhorado com o uso da representação de adjacência do grafo e com a descrição de novas otimizações para a geração da estrutura candidata, embora, é só aplicável para grafos estáticas.

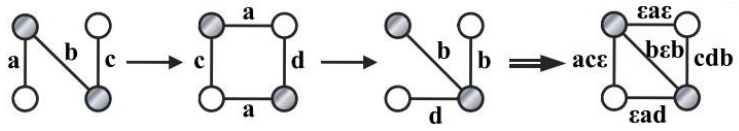


Figura 4.2. Transformação de um grafo no tempo em um grafo dinâmico. Os três grafos da esquerda são a representação de inserções e remoções de *links* no tempo. O grafo da direita é um grafo dinâmico que sumariza toda as informações que ocorreram ao longo do tempo.

Em [8], o autor baseia-se na idéia de árvores de sufixos para encontrar as sub-estruturas mais freqüentes mediante a união de grafos numa série de tempo. Este trabalho aproveita tanto as informações dos vértices e das arestas de grafos dinâmicos, considerando em particular operações de inserção e remoção de arestas no tempo. A figura 4.2 amostra a representação de um grafo dinâmico numa série de tempo.

Em [9], as técnicas de programação lógica indutiva são usadas para encontrar padrões freqüentes em grafos num domínio da toxicologia.

Uma proposta feita em [10] para a geração e compressão eficiente das sub-estruturas mais freqüentes é mediante o uso de busca com heurística greedy local, sendo o *Subdue*[1] e *GBI* as aplicações mais notável. O *Subdue* [2] está baseado na heurística *MDL* (*Minimum Description Lenght*) para encontrar a estrutura padrão que permite uma melhor compressão do grafo. O *GBI* - Indução Baseada em Grafos [11], faz a compressão do grafo pela fragmentação de paires de vértices que aparecem com mais freqüência. A figura 4.3 amostra o processo de compressão feito pelo *Subdue*.

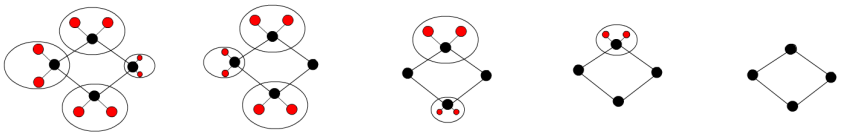


Figura 4.3. O processo de compressão do grafo depois da descoberta das sub-estruturas feito pelo *Subdue*. Em (a), (b), (c) e (d) amostra-se que as sub-estruturas encontradas na figura 4.1 são reduzidas ao vértice central que tem a ligação com o resto do grafo. Isso permite a otimização do tamanho do grafo.

4.2 Classificação de grafos

A classificação de grafos não é similar à classificação de objetos baseada em *links*, a qual tem por objetivo o rotulado dos vértices num grafo. A classificação de grafos é um problema de aprendizado supervisionado que tem o objetivo de categorizar um grafo inteiro como uma instância positiva ou negativa de um conceito. A classificação de grafos é uma das primeiras tarefas que foi direcionada no contexto da aplicação das técnicas de aprendizado de máquina e de mineração de dados para dados com estrutura de grafos [1].

Outra diferença da classificação de grafos com a classificação de objetos baseada em *links* é que a classificação de grafos não precisa (sempre) da inferência coletiva, devido que os grafos são geralmente gerados de forma independente. A maior dificuldade na classificação de grafos é lidar com a estrutura complexa dos grafos é obter um vetor para sua representação.

[1] `http://ailab.uta.edu/old_site/subdue/`

4.2.1 Abordagens

A classificação de grafos tem muitas abordagens, sendo três delas as mais importantes: mineração de padrões de grafos (*Feature Mining on Graphs*), programação lógica indutiva (*Inductive Logic Programming*), e Kernels de grafos (Graphs kernels).

Mineração de Padrões de Grafos

É uma tarefa muito relacionada com a descoberta de sub-estruturas (ver Seção 4.1) nas instâncias de um grafo. O processo de mineração de padrões de grafos começa com a descoberta de todas as sub-estruturas mais freqüentes ou informativas do grafo. Essas sub-estruturas descobertas são utilizadas para fazer a transformação dos dados do grafo em dados representados por uma tabelas simples. Este processo é feito para todos os grafos de nosso conjunto de grafos.

O objetivo de fazer a transformação dos dados dos grafos em tabelas simples é permitir o uso dos classificadores tradicionais nas instâncias da tabela gerada.

Em [14] usa-se um algoritmo de mineração de sub-estruturas freqüentes para construir descritores nessas sub-estruturas e aplicar o principio da máxima entropia para converter os padrões locais em um modelo de classificação global para dados com estrutura de grafos.

Em [15] é considerada a distribuição espacial das características da sub-estrutura no grafo para escolher só aquelas que tem localização espacial consistente.

A presença de grafos previamente rotulados e de outros ainda não rotulados é um problema da classificação de grafos, [12] propor uma seleção de padrões semi-supervisionada e o uso de um algoritmo de ramificação e acotamento (*branch and bound*) para a busca eficiente das sub-estruturas ótimas do grafo e seu posterior classificação.

Muitas abordagens da classificação de grafos tem por objetivo a colocação de um único rótulo para um grafo do conjunto de grafos, [13] propor a colocação de muitos rótulos simultaneamente para um mesmo grafo. o autor baseia-se na extração de bons padrões a partir do critério de independência de Hilbert-Schmidt e o uso de um algoritmo de ramificação e acotamento (*branch and bound*) para a busca eficiente de sub-estruturas ótimas em um grafo do conjunto de grafos.

Programação Lógica Indutiva

A Programação Lógica Indutiva (*Inductive Logic Programming*) é uma abordagem que utiliza sistemas de programação lógica indutiva para criar uma hipótese que permita fazer a classificação de grafos.

Em [16] constrói-se um mapa dos dados do grafo que descreve a mutagênese na representação relacional dos dados. Seu representação lógica utiliza relações dos vértices com as arestas para depois utilizar um sistema de programação lógica indutiva para a busca de uma hipótese em seu espaço.

Kernels de grafos

Encontrar todas as sub-estruturas freqüentes de todos os grafos num conjunto de grafos é uma tarefa computacionalmente proibida. O uso de kernels é uma alternativa para isso.

Os trabalhos de [17] e [18] descrevem kernels de grafos baseados nas medidas de caminhos nos grafos. O [17] propor um kernel que quantifica os caminhos que têm rótulos iguais no começo e no final. O [18] propor um kernel que obtém a probabilidade dos caminhos aleatórios com iguais seqüências dos rótulos.

Em [19] apresenta-se um kernel baseado num modelo probabilístico. O kernel é aplicado na classificação de grafos que representam estruturas de sequências de proteínas.

4.3 Geração de Modelos para grafos

A geração de modelos para grafos é a tarefa de mineração de *links* relacionada com grafos que tenta desenvolver métodos de construção de modelos sobre conjuntos de dados com estruturas de grafos, ou seja, gerar a partir de um conjunto de grafos gerar novos grafos que fazem parte da distribuição do conjunto de grafos original [20].

Os modelo de geração para uma grande gama de tipos de grafos têm sido estudadas extensivamente na comunidade de análise de redes sociais [1]. Os modelos de geração de grafos admitem estruturas de dependência que são mais gerais que as introduzidas pelos grafos de Markov, e juntamente com modelos para muitos objetos, tipos de *links* e redes dinâmicas com variadas estruturas de *links* e quantidade de objetos [22].

4.3.1 Abordagens

Muitas abordagens foram feitas para esta tarefa e ainda tem muitas propostas. Para grafos dirigidos com um objeto simples e um tipo de link as melhores propostas foram feitas utilizando distribuições aleatórias de grafos. Os grafos de Bernoulli [21] (também conhecido como modelo Erdos-Rényi ou grafos aleatórios) assume que as arestas dirigidas que pertencem a os objetos origem e destino têm uma distribuição idêntica e independente (IID) e dessa maneira estabelece que quando a probabilidade da existência de um link é 0.5 se tem um grafo com distribuição aleatória uniforme.

Em [23] é apresentado um modelo de geração para *links* observados entre os indivíduos dada sua participação na rede que pertencem. Em [24] apresenta um modelo de geração de *links* para uma analise de *links* e consultas de colaboração que permitem tipos diferentes de *links* e informação temporal.

Em [25] apresenta-se um modelo relacional probabilístico que fornece um modelo de geração unificado para objetos e *links*.

5

Conclusões

Depois de apresentar um panorama da mineração de links, sua importância e suas principais tarefas, nossas conclusões são:

1. A tarefa com link tem como ponto central à predição de links, onde o problema é determinar quais links estarão presentes o ausentes depois de um tempo dado.
2. O problema de predição de links tem dois subtipos de problema, os links a serem preditos podem não existir realmente na rede ou os links se existem mas não se conhece sua existência.
3. Existem muitas medidas para a medição topológica da estrutura da rede, mas uma das mais eficazes é: Adami.
4. As redes sociais são muito grandes na maioria de vezes, pelo que é necessário estabelecer técnicas incrementais que possam estabelecer grupo pequenos que vai inserindo novos membros à rede até definir a rede completamente.
5. A resolução de entidades tenta solucionar problema relativos à duplicação de dados o integração de dados em caso de bases de dados, mas é possível encontrar casos semelhantes em outras áreas de trabalho.
6. A resolução de entidades tem como uma técnica importante a *clusterização* para poder representar melhor os dados e poder identificar com sucesso as referências analisadas.
7. As tarefas realizadas nos grafos baseadas em seus links, trabalham numa coleção de grafos é consideram um grafo como uma entidade relacionada cujos atributos são gerados pelos padrões existentes no mesmo grafo (relações entre vértices e atributos dos vértices relacionados).
8. A tarefa de descoberta de sub-grafos é importante porque encontra sub-estruturas freqüentes, que por sua vez, são importantes no grafo, estas sub-estruturas permitem a compressão do grafo o qual é necessário para redes muito grandes e que apresentam muitas sub-estruturas como por exemplo nas redes de biologia ou química.

9. A tarefa de classificação de grafos permite rotular um grafo numa coleção de grafos, a capacidade de atribuir vários rótulos a um único grafo é umas dos desafios que tem esta tarefa.
10. A tarefa de geração de grafos permite gerar um novo grafo a partir de um conjunto de grafos, sendo que o novo grafo pode apresentar as melhores estruturas do conjunto de grafos original. O uso das informações dos links é muito importante nesta tarefa porque permite a geração de um modelo com estruturas de dependência mais gerais.

Referências

1. Getoor, Lise and Diehl, Christopher (2005) Link Mining: A Survey. SigKDD Explorations Special Issue on Link Mining 2(7).
2. D. J. Cook and L. B. Holder (1994) Substructure discovery using minimum description length and background knowledge. Journal of Artificial Intelligence Research, 1:231255.
3. Winston, P.H. (1975). Learning structural description from examples. In Winston, P.H. (Ed.), The Psycology of Computer Vision, pp. 157-210, McGraw-Hill.
4. Levinson, R. (1984). A self-organization retrieval system for graphs. In Proceedings of the Second National Conference on Artificial Intelligence, pp. 203-206.
5. Segen, J. (1990). Graph clustering and model learning by data compression. In Proceedings of the Seventh International Machine Learning Workshop, pp. 93-101.
6. Inokuchi A., T. Washio, and H. Motoda. (2000) An Apriori based algorithm for mining frequent substructures from graph data. In European Conference on Principles and Practice of Knowledge Discovery and Data Mining, pages 13-23.
7. M. Kuromachi and G. Karypis (2001). Frequent subgraph discovery. In IEEE International Conference on Data Mining, pp. 313-320.
8. B. Wackersreuther, P. Wackersreuther, A. Oswald, C. Bóhm and K. Borgwardt (2010). Frequent Subgraph Discovery in Dynamic Networks, MLG's 10, Washington pp. 155-162.
9. L. Dehaspe, H. Toivonen and R. King (1998). Finding Frequent Substructures in Chemical Compounds, Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD).
10. T. Matsuda, T. Horiuchi, H. Motoda, and T. Washio (2000). Extension of graph-based induction for general graph structured data. In PAKDD, pages 420-431.
11. K. Yoshida, H. Motoda, and N. Indurkhya (1994). Graph based induction as a unified learning framework. Journal of Applied Intelligence, 4(3):297-316.
12. Kong, Xiangnan and Yu, P. S. (2010). Semi-supervised feature selection for graph classification. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'10, 793-802.
13. Kong, Xiangnan and Yu, Philip S. (2010). Multi-label Feature Selection for Graph Classification. Data Mining (ICDM), 2010 IEEE 10th International Conference on 2010.

14. H.D.K. Moonesinghe, H. Valizadegan, S. Fodeh, P.N. Tan (2007). A Probabilistic Substructure-Based Approach for Graph Classification. In 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), pp. 346-349.
15. H. Fei and J. Huan (2008). Structure Feature Selection for Graph Classification. In Proceedings of the 17th ACM Conference on Information and Knowledge Mining - CIKM'08.
16. R. D. King, S. H. Muggleton, A. Srinivasan, and M. J. E. Sternberg (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. National Academy of Sciences, 93(1):438-442.
17. T. Gartner (2002). Exponential and geometric kernels for graphs. In NIPS Workshop on Unreal Data: Principles of Modeling Nonvectorial Data.
18. H. Kashima and A. Inokuchi (2002). Kernels for graph classification. In ICDM Workshop on Active Mining.
19. T. Jaakkola, M. Diekhans, and D. Haussler (2000). A discriminative framework for detecting remote protein homologies. Journal of Computational Biology, 7(1,2):95-114.
20. D. White and R. C. Wilson (2010). Generative Models for Chemical Structures. In J. Chem. Inf. Model., 50(7), pp 1257-1274.
21. O. Frank and K. Nowicki (1993). Exploratory statistical analysis of networks. Annals of Discrete Mathematics, 55:349-366.
22. M. Huisman and T. A. B. Snijders (2003). Statistical analysis of longitudinal network data with changing composi- tion. Sociological Methods and Research, 32:253-287.
23. J. Kubica, A. Moore, J. Schneider, and Y. Yang (2002). Stochastic link and group detection. In Eighteenth National Conference on Artificial Intelligence, pp. 798-804. American Association for Artificial Intelligence.
24. J. Kubica, A. Moore, D. Cohn, and J. Schneider (2003). cGraph: A fast graph-based method for link analy- sis and queries. In IJCAI 2003 Text-Mining and Link- Analysis Workshop.
25. L. Getoor, N. Friedman, D. Koller, and B. Taskar (2003). Learning probabilistic models of link structure. Journal of Machine Learning Research, 3:679-707.
26. Jafar Adibi, Hans Chalupsky, Marko Grobelnik, Dunja Mladenic, and Natasa Milic-Frayling. 2004. KDD-2004 workshop report link analysis and group detection (LinkKDD-2004). SIGKDD Explor. Newsl. 6, 2 (December 2004), 136-139.
27. Wei Wang and Thomas E. Daniels. 2008. A Graph Based Approach Toward Network Forensics Analysis. ACM Trans. Inf. Syst. Secur. 12, 1, Article 4 (October 2008), 33 pages.
28. Fatih Ozgul, Julian Bondy, and Hakan Aksoy. 2007. Mining for offender group detection and story of a police operation. In Proceedings of the sixth Australasian conference on Data mining and analytics - Volume 70 (AusDM '07), Peter Christen, Paul Kennedy, Jiuyong Li, Inna Kolyshkina, and Graham Williams (Eds.), Vol. 70. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 189-193.
29. Jennifer J. Xu and Hsinchun Chen. 2005. CrimeNet explorer: a framework for criminal network knowledge discovery. ACM Trans. Inf. Syst. 23, 2 (April 2005), 201-226.

30. S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications (Structural Anaylysis in the Social Sciences), p.249. Cambridge Univesity Press, thirteenth edition, 1994.
31. J. Adibi, H. Chalupsky, E. Melz, A. Valente, and Others. The KOJAK Group Finder: Connecting the Dots via Integrated Knowledge-Based and Statistical Reasoning. Innovative Applications of Artificial Intelligence Conference, 2004.
32. Yinle Zhou and John Talburt. 2011. Staging a realistic entity resolution challenge for students. J. Comput. Small Coll. 26, 5 (May 2011), 88-95.
33. Steven Euijong Whang, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina. 2009. Entity resolution with iterative blocking. In Proceedings of the 35th SIGMOD international conference on Management of data (SIGMOD '09), Carsten Binnig and Benoit Dageville (Eds.). ACM, New York, NY, USA, 219-232
34. Aron Culotta and Andrew McCallum. 2005. Joint deduplication of multiple record types in relational data. In Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM '05). ACM, New York, NY, USA, 257-258.
35. Pedro Domingos. 2004. Multi-relational record linkage. In Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining. 31–48.
36. Zhaoqi Chen, Dmitri V. Kalashnikov, and Sharad Mehrotra. 2007. Adaptive graphical approach to entity resolution. In Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries (JCDL '07). ACM, New York, NY, USA, 204-213.
37. Indrajit Bhattacharya and Lise Getoor. 2007. Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data 1, 1, Article 5 (March 2007)
38. S. Asur, S. Parthasarathy and Duygu Ucar. An Event-based Framework for Characterizing the Evolutionary Behavior of Interaction Graphs.
39. Jianhan Zhu, Jun Hong, and John G. Hughes. 2002. Using Markov models for web site link prediction. In Proceedings of the thirteenth ACM conference on Hypertext and hypermedia (HYPERTEXT '02), James Blustein (Ed.). ACM, New York, NY, USA, 169-170.
40. Magdalini Eirinaki and Michalis Vazirgiannis. 2007. Web site personalization based on link analysis and navigational patterns. ACM Trans. Internet Technol. 7, 4, Article 21 (October 2007).
41. Magdalini Eirinaki, Michalis Vazirgiannis, and Dimitris Kapogiannis. 2005. Web path recommendations based on page ranking and Markov models. In Proceedings of the 7th annual ACM international workshop on Web information and data management (WIDM '05). ACM, New York, NY, USA, 2-9.
42. Faten Khalil, Jiuyong Li, and Hua Wang. 2008. Integrating recommendation models for improved web page prediction accuracy. In Proceedings of the thirty-first Australasian conference on Computer science - Volume 74 (ACSC '08), Gillian Dobbie and Bernard Mans (Eds.), Vol. 74. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 91-100.
43. Linyuan Lü and Tao Zhou. 2009. Role of weak ties in link prediction of complex networks. In Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management (CNIKM '09). ACM, New York, NY, USA, 55-58.
44. Zan Huang, Xin Li, and Hsinchun Chen. 2005. Link prediction approach to collaborative filtering. In Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries (JCDL '05). ACM, New York, NY, USA, 141-142.

45. David Liben-Nowell and Jon Kleinberg. 2003. The link prediction problem for social networks. In Proceedings of the twelfth international conference on Information and knowledge management (CIKM '03). ACM, New York, NY, USA, 556-559.
46. pagebrin Page L., Brin S., Motwani R., and Winograd T. (1998) The PageRank citation ranking: Bringing order to the web.Technical report, Stanford University
47. jensen1999 Jensen D. (1999)Statistical challenges to inductive inference in linked data. In Seventh International Workshop on Artificial Intelligence and Statistics
48. jensenJensen D. and Goldberg H.(1998) AAAI Fall Symposium on AI and Link Analysis. AAAI Press
49. Boyd, D.M., and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), article 11.
50. Golder, S. A., Wilkinson, D., and Huberman, B. A. (2007, June). Rhythms of social interaction: Messaging within a massive online network. In C. Steinfield, B. Pentland, M. Ackerman, and N. Contractor (Eds.), Proceedings of Third International Conference on Communities and Technologies (pp. 41-66). London: Springer.
51. Lampe, C., Ellison, N., and Steinfield, C., (2006). A Face(book) in the crowd: Social searching vs. social browsing. Proceedings of CSCW-2006 (pp. 167-170). New York: ACM Press.