Descoberta de Sub-Estruturas utilizando o Comprimento de Descrição Mínima

Jorge C. Valverde Rebaza, Pedro N. Shiguihara Juárez, and Iuliana G. S. Rodrigues

Instituto de Ciências Matemáticas e de Computação
Universdidade de São Paulo

Resumo Neste trabalho apresenta-se o funcionamento do algoritmo que é utilizado pelo *Subdue* para a descoberta das sub-estruturas mais freqüentes num grafo de entrada procurando obter a maior compressão dele. A escolha da sub-estrutura mais freqüente que permita uma maior compressão do grafo de entrada é feita com base no principio do Comprimento de Descrição Mínima - *Minimum Description Length Principle* - (MDLP), é a busca dessas sub-estruturas é feita com base na estratégia de busca *beam search*. Tendo em conta a distorção das instâncias da sub-estrutura no grafo é considerado um casamento inexato ao momento de substituir essa instância por um só vértice simples que represente a sub-estrutura inteira. A extração de sub-estruturas representativas em 10,000 compostos químicos foi feita em 5531.86 segundos. Os resultados também mostraram que é possível extrair múltiplas patrões de múltiplas arestas entre apenas um par de vértices.

1 Introdução

A habilidade de identificar subestruturas de interesse é um componente essencial para a descoberta do conhecimento em dados estruturais. A grande quantidade de dados coletados hoje é utilizada por pesquisadores para buscar e interpretar dados na tentativa de descobrir padrões de interesse nesses dados.

Os algoritmos para descoberta de padrões frequentemente são utilizados em várias áreas de aplicação. Contudo essas técnicas são aplicadas a conjuntos não tradicionais, havendo a necessidade de padrões de algoritmos que seja capaz de capturar a força espacial, topológica, geométrica e a natureza relacional de conjuntos que caracterizam esses domínios.

Durante anos, o estudo grafos rotulados surgem como uma abstração promissora para capturar características nesses conjuntos de dados. Nesta abordagem, a representação onde os vértices são os objetos e as arestas são as relações entre eles, à descoberta de sub-grafos ocorre frequentemente dentro do conjunto inteiro de grafos [4].

A habilidade para modelar grafos a partir de um conjunto de dados complexos tem sido reconhecida por diversos pesquisadores como:

– O programa Arch para descoberta de sub-estruturas para aprofundar a descrição hierárquica de uma cena e agrupar objetos dentro de conceitos mais gerais [13].

– Descobriu um sistema para armazenar grafos usando o modelo de probabilidade para representar classes de grafos [11].
– O software Labirinth [12], extendeu o conceito de clusterização incremental do Cobweb para formar conceitos hierárquicos de representação de subestruturas comuns para a entrada de objetos.
– O software Clip [14], para grafos baseados na indução, interativamente descobre padrões (sub-estruturas) em grafos para expandir e combinar descoberta de padrões em iterações prévias.

A regra para produção de sub-grafos pode fornecer uma representação adequada para um conjunto de dados de conhecimento durante o processo de descoberta de sub-estruturas.

1.1 Subdue

Subdue[1] é um sistema baseado em descoberta de conhecimento, que encontra padrões estruturais de relacionamento de dados que representem entidades e relações. *Subdue* representa os dados através de um grafo rotulado, dirigido, na qual as entidades são representadas por vértices rotulados ou sub-grafos e relacionamentos são representados por arestas rotuladas entre as entidades. *Subdue* utiliza o princípio de comprimento mínimo de descrição (MDL) para identificar padrões que permitam minimizar o número de bits necessários para descrever o grafo de entrada depois de ser comprimido por padrão. Pode executar várias tarefas de aprendizagem, incluindo a aprendizagem não supervisionada, aprendizagem supervisionada, o agrupamento e a gramática do gráfico de aprendizagem. A utilização do *Subdue* pode ser comprovada pois foi aplicado com sucesso em várias áreas, incluindo a bioinformática, mineração web estrutura, combate ao terrorismo, análise de redes sociais, a aviação e geologia.

1.2 Comprimento de Descrição Mínima

Em [1], propor em seu trabalho a utilização da métrica do Comprimento de Descrição Mínima (*Minimum Description Length* - MDL), cuja origem é fundamentada na Teoria de Codificação como uma medida de qualidade para a escolha da estrutura de rede. O princípio básico consiste em reduzir ao máximo o número de elementos necessários para representar uma mensagem, baseando-se em sua probabilidade de ocorrência. Assim, mensagens mais frequentes são representadas por códigos menores e as mensagens menos freqüentes, por códigos maiores. No caso do aprendizado estrutural de redes, como por exemplo - Redes Bayesianas, a ideia básica é encontrar a estrutura de rede que melhor descreva o conjunto de dados, utilizando o mínimo de elementos possíveis para calcular a probabilidade conjunta da rede de crença, reduzindo dessa maneira o esforço computacional necessário no cálculo das inferências [8]. Nesse contexto, métrica

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

de pontuação com parâmetros mais restritivos, ou seja que selecionam estruturas de redes mais simples, apresentam resultados superiores àqueles menos restritivos.

Experimentos em uma variedade de domínios demonstram a habilidade do Subdue para encontrar subestruturas capazes de comprimir os dados originais e descobrir conceitos estruturais importantes para o domínio. Algumas dessas informações serão descritos em subsecções posteriores. Na Seção 2 são amostradas as informações relacionadas a fundamentação do trabalho. Na Seção 3 são fornecidas informações relacionadas à avaliação experimental. Na Secção 4 são discutidas os resultados. Na Secção 5 são fornecidas nossas conclusões.

2 Fundamentação

Nesta seção são apresentados os principais conceitos relacionados com a descoberta de sub-estruturas com base no principio do Comprimento de Descrição Mínima para a escolha da melhor sub-estrutura para conseguir a melhor compressão de um dado grafo.

2.1 Descoberta de Sub-estruturas

Um sistema de descoberta de sub-estruturas representa dados estruturados como um grafo rotulado. Em [4], os objetos do conjunto de dados são os vértices do grafo e as relações entre eles são as arestas do mesmo grafo, as quais podem ser dirigidas ou não dirigidas. Um sub-grafo é a parte mínima de um grafo, isso poder ser, por exemplo, um só vértice. Uma sub-estrutura é a conexão de sub-grafos. Na figura 1, apresenta-se na parte (a), um exemplo de um grafo formado com formas geométricas, e na parte (b) uma de suas sub-estruturas. Os objetos no grafo são vértices rotulados (por exemplo, T1, S1, R1), e suas relações são as arestas rotuladas (por exemplo, on(T1, S1), shape(T1, triangle)).

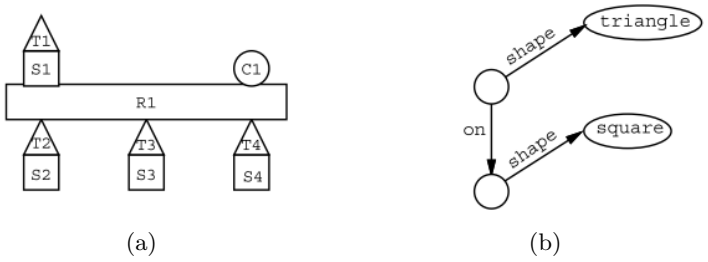


Figura 1. Em (a), apresenta-se um exemplo de um grafo. Em (b), apresenta-se uma sub-estrutura do grafo de (a), [4].

Una instância de uma sub-estrutura num grafo de entrada é um conjunto de vértices e arestas obtidas do grafo de entrada, os quais fazem casamento com uma sub-estrutura. Por exemplo, as instâncias da sub-estrutura da figura 1(b) no grafo da figura 1(a), são apresentadas na figura 2.

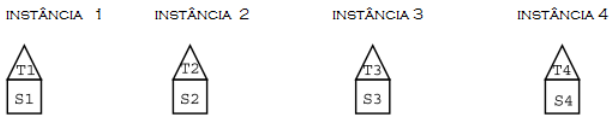


Figura 2. Instâncias da sub-estrutura da figura 1(b) no grafo da figura 1(a), [4].

O algoritmo de descoberta de sub-estruturas usado pelo *Subdue* faz uso da estratégia de busca *beam search*. O algoritmo começa fazendo o casamento com uma sub-estrutura composta por um só vértice do grafo. Para cada iteração, o algoritmo faz a escolha da melhor sub-estrutura e amplia a busca das instâncias da sub-estrutura por algum de suas arestas vizinhas em todas as formas possíveis. O algoritmo procura a melhor sub-estrutura até que todas as possíveis sub-estruturas sejam consideradas ou a quantidade total de computação ultrapassa um determinado limite. A avaliação de cada sub-estrutura é feita pelo principio de comprimento de descrição mínima (MDL).

Tipicamente, alguns comprimentos de descrições de uma sub-estrutura começa a ter uma ampliação que, logicamente, não irá a produzir um menor comprimento da descrição. Nesse caso, o algoritmo utiliza um mecanismo de poda para excluir as sub-estruturas cuja ampliação aumenta.

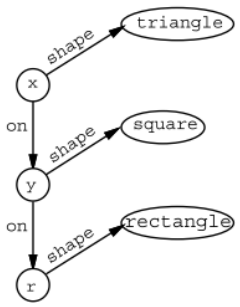
2.2 O Comprimento de Descrição Mínima para a codificação de Grafos

O principio do Comprimento de Descrição Mínima - *Minimum Description Length Principle* - (MDLP) foi introduzida pelo [10], diz que a melhor teoria para a descrição de um conjunto de dados é a teoria que minimiza o comprimento da descrição do conjunto de dados inteiro. O MDLP tem sido utilizado em diferentes áreas como: indução de árvores de decisão [9], processamento de imagens [7], aprendizado de conceitos desde dados relacionais [5], entre outros.

Em [4] é apresentado o uso do principio do Comprimento de Descrição Mínima (MDLP) para a descoberta de sub-estruturas em dados de redes complexas. Em particular, a avaliação de uma sub-estrutura é baseada em quão bem ele pode comprimir o conjunto de dados inteiro utilizando o Comprimento de Descrição Mínima (*Minimum Description Length* - MDL). Assim, a definição do Comprimento de Descrição Mínima de um grafo é o número de bits necessários para descrever completamente o grafo.

De acordo ao principio do Comprimento de Descrição Mínima (MDLP), a teoria que melhor representa uma coleção de dados é uma que minimiza $I(S) + I(G|S)$, donde S é a sub-estrutura descoberta, G é o grafo de entrada, $I(S)$ é o número de bits requeridos para fazer a codificação da sub-estrutura descoberta, e $I(G|S)$ é o número de bits requeridos para a codificação do grafo de entrada G com relação a S.

A conectividade do grafo pode ser representada por uma matriz de adjacência. Assim, considerando um grafo que tem n vértices, os quais são enumeradas $0, 1, \ldots, n-1$. Uma matriz de adjacência A de tamanho $n \times n$ pode ser formada com um ítem $A[i, j]$ com valor 0 ou 1. Se $A[i, j] = 0$, não existe conexão desde o vértice i ao vértice j. Se $A[i, j] = 1$, existe uma conexão desde o vértice i ao vértice j. Na figura 3(a), apresenta-se um exemplo de um grafo e na figura 3(b), seu respectiva matriz de adjacência.



(a)

$$\begin{array}{r} x \\ triangle \\ y \\ square \\ r \\ rectangle \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b)

Figura 3. Em (a), apresenta-se um exemplo de grafo. Em (b), apresenta-se sua matriz de adjacência, [4].

A codificação de um grafo é feita assumindo que o decodificador tem uma tabela dos l_u rótulos únicos do grafo de entrada G. Então, os pasos para fazer a codificação de um grafo são:

1. Determinar o número de bits *vbits* necessários para codificar os vértices rotulados do grafo. Primeiro, precisa-se $(\log v)$ bits para codificar os v vértices de um grafo. Então, codificar os rótulos de todos os v vértices precisa de $(v \log l_u)$ bits. Assumindo que os vértices são especificados no mesmo ordem o qual aparecem na matriz de adjacência, então, o número total de bits para codificar os rótulos dos vértices é:

$$vbits = \log v + v \log l_u \tag{1}$$

Por exemplo, no grafo da figura 3(a), o número de vértices é $v = 6$, e o número de rótulos únicos do grafo é $l_u = 8$, então o número de bits ne-

cessários para codificar esses vértices é $\log 6 + 6 \log 8 = 20.58$ bits.

2. Determinar o número de bits *rbits* necessários para codificar as linhas da matriz de adjacência A. Tipicamente, em grafos muito grandes, um só vértice pode ter arestas para só um pequeno percentagem dos vértices do grafo inteiro. Por esse motivo, uma linha típica na matriz de adjacência terá muito menos que v 1's, onde v é o número total de vértices no grafo. Então, é possível representar a linha i ($1 \leq i \leq v$) como uma cadeia de bits de tamanho v contendo k_i 1's. Se, nos temos que $b = \max k_i$, então a $i^{ésima}$ linha da matriz de adjacência pode ser codificada da seguinte maneira:
   (a) Codificar o valor de k_i precisa de $\log(b+1)$ bits.
   (b) Dado que apenas k_i 1's só ocorrem na linha (cadeia de bits) de tamanho v, só $\binom{v}{i}$ cadeias de 0's e 1's são possíveis. Uma vez que todas as cadeias têm a mesma probabilidade de ocorrência, $\log \binom{v}{i}$ bits são necessários para codificar as posições dos 1's na linha i. O valor de v, já é conhecido.

   Finalmente, é preciso uma quantidade $\log(b+1)$ bits para codificar o número de bits necessários para especificar o valor de k_i para cada linha. Então o número de bits para codificar as linhas da matriz de adjacência é:

$$rbits = \log(b+1) + \sum_{i=1}^{v} \log(b+1) + \log \binom{v}{i}$$

$$rbits = (v+1)\log(b+1) \sum_{i=1}^{v} \log \binom{v}{i} \tag{2}$$

   Por exemplo, no grafo da figura 3(a), $b = 2$, e o número de bits necessários para codificar a matriz de adjacência é $7 \log 3 + \log \binom{6}{2} + \log \binom{6}{0} + \log \binom{6}{2} + \log \binom{6}{0} + \log \binom{6}{1} + \log \binom{6}{0} = 21.49$ bits.

3. Determinar o número de bits *ebits* necessários para codificar as arestas representadas pelos ítens $A[i,j] = 1$ da matriz de adjacência A. O número de bits necessários para codificar o ítem $A[i,j]$ é $(\log m) + e(i,j)[1 + \log l_u]$, onde $e(i,j)$ é o número atual de arestas entre os vértices i e j no grafo, e $m = max_{i,j} e(i,j)$. É preciso $(\log m)$ bits para codificar o número de arestas entre os vértices i e j, e $[1 + \log l_u]$ bits por cada aresta para codificar os rótulos de cada aresta e se a aresta é dirigida ou não. Além disso, para a codificação das arestas, precisa-se codificar o número de bits $(\log m)$ necessários para especificar o número de arestas por ítem. Então, o número total de bits na codificação de arestas é:

$$ebits = \log m + \sum_{i=1}^{v} \sum_{j=1}^{v} \log m + e(i,j)[1 + \log l_u]$$

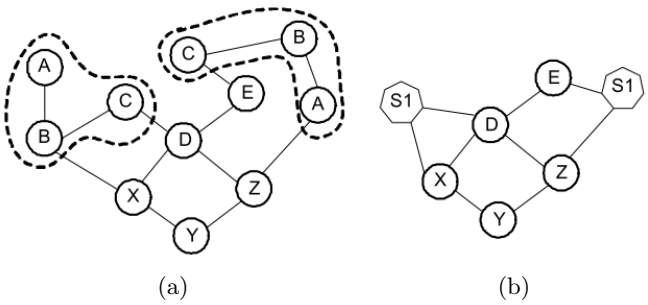
$$ebits = \log m + e(1 + \log l_u) + \sum_{i=1}^{v}\sum_{j=1}^{v} A[i,j] \log m$$

$$ebits = e(1 + \log l_u) + (K + 1) \log m \tag{3}$$

onde e é o número de arestas no grafo, e K é o número de 1's na matriz de adjacência A. Por exemplo, no grafo da figura 3(a), $e = 5$, $K = 5$, $m = 1$, $l_u = 8$, e o número de bits necessários para codificar as arestas é $5(1 + \log 8) + 6 \log 1 = 20$ bits.

Então, para codificar um grafo inteiro precisa-se de ($vbits$ + $rbits$ + $ebits$) bits. Por exemplo, no grafo da figura 3(a), esse valor é 62.07 bits.

Assim, o grafo de entrada e a sub-estrutura descoberta podem ser codificados utilizando o esquema apresentado. Depois de que a sub-estrutura é descoberta, cada instância da sub-estrutura no grafo de entrada é substituída por um vértice simples que representa à sub-estrutura inteira. Na figura 4 apresenta-se o processo de substituição das instâncias da sub-estrutura descoberta num grafo.



(a) (b)

Figura 4. Em (a), apresenta-se um exemplo de grafo e uma sub-estrutura comum que foi descoberta. Em (b), apresenta-se o grafo depois da substituição de todas as instâncias da sub-estrutura descoberta por um só vértice simples, [3].

A sub-estrutura descoberta é representada em $I(S)$ bits, e o grafo depois da substituição da sub-estrutura é representado em $I(G|S)$ bits. Então, para ter uma melhor compressão do grafo é preciso a busca da sub-estrutura S no grafo G que poda minimizar $I(S) + I(G|S)$.

É importante notar que, o processo de busca de uma sub-estrutura S no grafo G é baseado no trabalho de [2] o qual faz o casamento inexato de grafos tendo em conta a distorção que as diferentes instâncias da sub-estrutura podem ter em todo o grafo.

3 Avaliação Experimental

Para efetuar a avaliação, utilizamos uma das distâncias descritas em [6] para medir a similaridade entre o sub-grafo e o grafo original chamada *grau de distribuição*. Esta medida é uma propriedade do grafo muito conhecida, em que, o grau de um nó é a quantidade de arestas ligadas a esse nó. Para este experimento, se utilizou conjuntos de dados no domínio de compostos químicos. Cada composto químico está formado normalmente por um conjunto de átomos, ligados entre si por conexões distintas. De esta forma, cada átomo pode estar ligado mais de uma vez com outro átomo pelo que podem ter múltiplas arestas entre eles. Na figura 5 se observa as interações de vários átomos em um só composto químico, de esta forma, podemos ver que toda a estrutura de um composto pode ser representado como um grafo não dirigido.

Assim, os conjuntos de dados utilizados foram de compostos químicos obtidos de dois bases de dados públicas. O primeiro conjunto de dados é chamado *PTE*, porque originalmente foi utilizado para a avaliação de previsão toxicológica, cujo acrônimo em inglês vem do termo: *Predictive Toxicology Evaluation*. O segundo conjunto de dados pertence a *The National Cancer Institute* e é parte do programa de desenvolvimento terapêutico com o acrônimo em inglês: *DTP*.

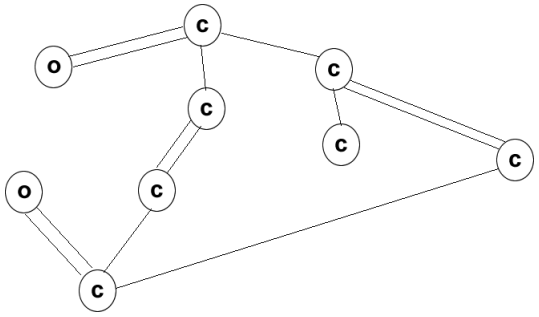


Figura 5. Estrutura de um composto químico da base de dados DTP, que esta conformada por 9 átomos.

3.1 Conjunto de dados PTE

O primeiro conjunto de dados[2] contém 340 compostos químicos. Cada composto químico contém átomos que estão ligados a través de uma conexão. Assim, nós pre-processamos os dados para que possam ser transformados a grafos, onde cada grafo representa um composto químico, os vértices são os átomos e as arestas

[2] `ftp://ftp.comlab.ox.ac.uk/pub/Packages/ILP/Datasets/carcinogenesis/progol/carcinogenesis.tar.Z`

são os tipos de conexões entre átomos. Na base de dados original, as informações de vértices e arestas estão descritas como fatos ou base de conhecimento dum programa em Prolog, onde a informação dos átomos está descrita por o nome do composto ao que pertence, o ordem em que se encontra dentro do composto, a tipo de átomo que é, a carga que tem e informação adicional; isto pode ser descrito da seguinte forma: *atm(nom_composto,ord_atom, tipo_atomo, carga, info_adic)*. Assim também, os conexões (arestas) estão descritas pelo nome do composto, os nomes dos átomos que conformam a conexão e o tipo de conexão que há entre eles; como na seguinte forma: *bond(nom_composto, nom_atomo1, nom_atomo2, nom_conexão)*. Ao ter 340 compostos químicos, geramos 340 grafos onde se encontram as interações de seus átomos com diferentes tipos de conexões entre si. A média de arestas para cada grafo é de 27. Para este conjunto de dados, avaliamos todo o conjunto inteiro.

3.2 Conjunto de dados DTP

O segundo conjunto de dados [3] foi extraído de *The National Cancer Institute*, do programa de desenvolvimento terapéutico, cujo acrônimo em inglés é: *DTP*. Nós obtivemos o conjunto de dados atualizados até o ano 2010. Neste caso a base de dados contém 266151 compostos químicos, com um tamanho de arquivo de 648 MB. Os compostos químicos estão com o formato de *arquivo de dados estruturais* (*SDF*, acrônimo em inglês). Para este conjunto de dados, utilizamos uma seleção de diversas quantidades de compostos químicos (Q) acrescentando-se gradualmente: $Q_1 = \{10, 20, \ldots, 100\}$, $Q_2 = \{100, 200, \ldots, 1000\}$ e $Q_3 = \{2000, 3000, 4000, 5000, 10000\}$ extraídas da base de dados *DTP*. Todos esses conjuntos de compostos químicos foram convertidos a grafos de uma forma parecida à utilizada para o conjunto de dados *PTE* na seção 3.1. Neste caso, a média de arestas para cada grafo obtido é de 22.

Na figura 6 se mostra como se acrescenta gradualmente o conjunto de dados de teste para ser analisado e extrair os sub-grafos representativos de aqueles grafos ou compostos químicoss.

[3] `http://dtp.nci.nih.gov/docs/3d\_database/structural\_information/structural\_data.html`

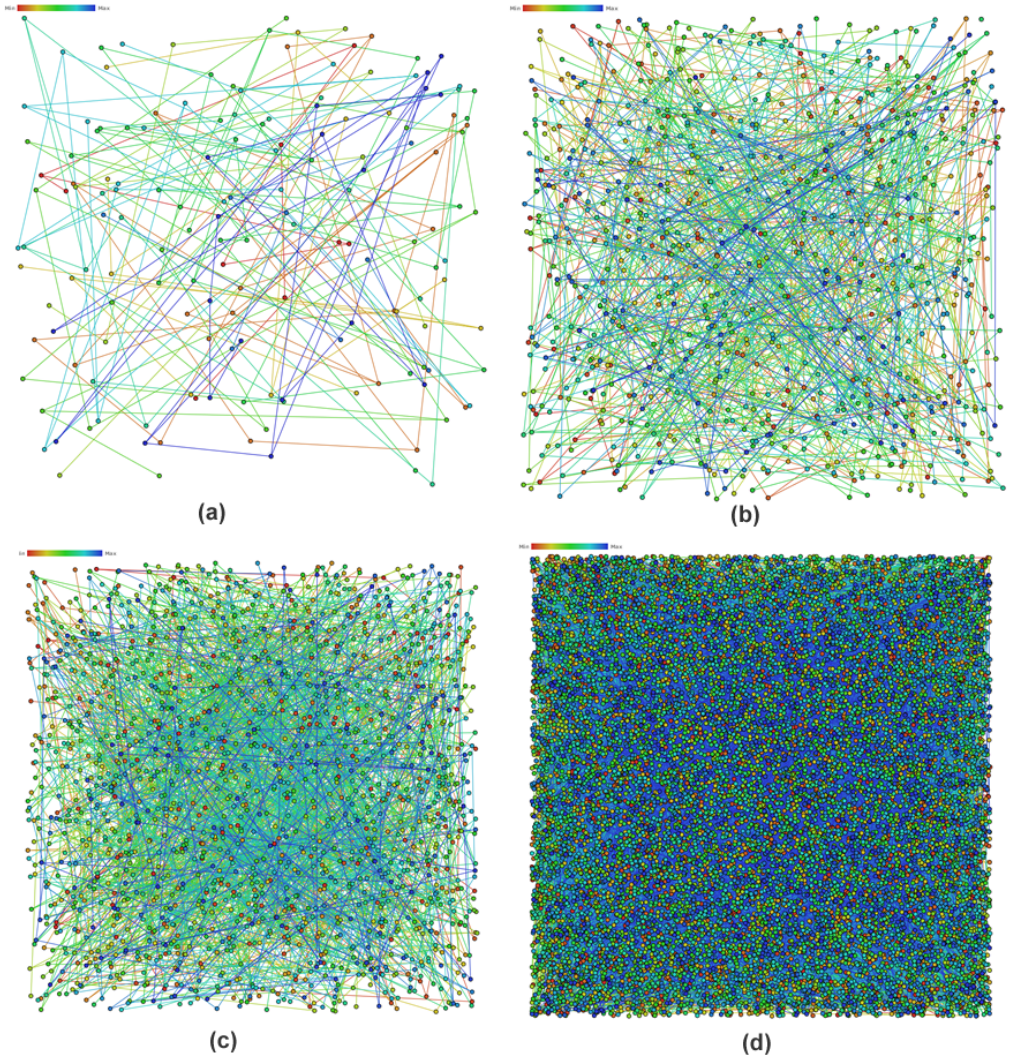


Figura 6. (a) 10 compostos químicos, (b) 50 compostos químicos, (c) 100 compostos químicos e (d) 1,000 compostos químicos. Cada composto esta conformado por átomos que interagem ente si mediante múltiplas tipos de conexões.

4 Resultados e Discussão

4.1 Conjunto de dados PTE

O conjunto de dados PTE contém 340 compostos químicos, o que representa 340 grafos onde seus vértices são os átomos e seus arestas são as interções entre esses átomos. Sobre o conjunto de dados PTE se efetuaram dois análises utilizando a medida MDL e uma medida estrutural baseada no tamanho do grafo sobre o tamanho da possível sub-estrutura chamada *Size* na ferramenta *Subdue*.

Tabela 1. Obtenção das três melhores sub-estruturas obtidas pela medida *MDL* e *Size* onde se observa cada sub-estrura como um grafo $G_s < V, E >$, onde V é o conjunto de vértices e E é o conjunto de arestas.

Obtenção de sub-grafos $G_s(V, E)$		
Melhor sub-estrutura	MDL $[t = 12.55seg]$	Size $[t = 11.79seg]$
1	(3, 2)	(3, 2)
2	(8, 8)	(7, 7)
3	(9, 9)	(8, 8)

Como se mostra na tabela 1, a medida MDL tem um consumo de tempo muito maior à medida Size, mas as melhoes sub-estruturas de MDL são mais compactas que da medida Size, já que a quantidade de vértices e arestas são maiores para MDL, para este conjunto de dados de 340 grafos.

4.2 Conjunto de dados DTP

A primeira avaliação neste conjunto de dados, foi feita sobre um único composto químico. Este composto é descrito na figura 5. Na figura 7 se pode observar que foram obtidos três sub-estruturas representativas do composto. Observe-se que a análise inclui a interação de dois átomos de carbono de dois formas distintas, como se mostra na parte *(a)* e *(b)* da figura 7.

Como se mostra na tabela 2, o algoritmo de descoberta de subgrafos patrões utilizando a medida *MDL* obteve compressões iniciais (ver $|F|$) muito grandes; como no caso da utilização de 5,000 compostos, onde se obtiveram 32 subgrafos representativos desses compostos (grafos), e para 10,000 compostos foram 34 subgrafos. Além, cada vez que a quantidade de compostos aumentava, então, a quantidade de segundos se acrescentava grandemente. Por exemplo, com 1,000 compostos se obteve 51.72 segundos, mas ao dobrar a quantidade de compostos a 2,000, o tempo aumentou 3 vezes mais (165.49 segundos) e assim por diante.

Na figura 8 mostra os resultados da tabela 2, com relação à quantidade de compostos com o tempo utilizado para calcular seus sub-grafos patrões.

Os mesmos conjuntos de dados foram empregados utilizado a medida estrutural *Size* que consiste em que o valor da sub-estrutura é a divisão do tamanho

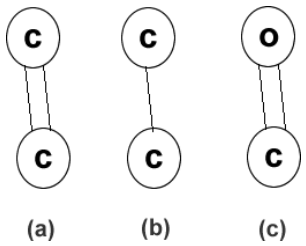


Figura 7. Três sub-estruturas patrões obtidas do cálculo de um único composto químico. O composto é da figura 5. A interação dos átomos do carbono e oxigênio foram extraídas.

Tabela 2. Tabela de resultados para o conjunto de dados DTP. Onde: t[sec] é o tempo em segundos, n é a quantidade de compostos químicos utilizados, $|V|$ é a quantidade de vértices utilizados, $|E|$ é a quantidade de arestas utilizadas e $|F|$ é a quantidade de subestruturas iniciais encontradas. $|\alpha|$ é a medida de avaliação das sub-estruturas.

DTP $\|\alpha\| = MDL$				
t[sec]	n	$\|V\|$	$\|E\|$	$\|F\|$
0.27	10	159	171	6
0.20	20	310	329	6
0.33	30	459	481	7
0.86	40	616	645	7
1.14	50	785	821	7
1.92	60	954	1003	8
2.25	70	1156	1217	8
2.59	80	1317	1381	8
2.47	90	1475	1540	8
1.94	100	1604	1669	8
5.01	200	3137	3209	13
9.53	300	4678	4783	16
16.07	400	6414	6567	16
18.36	500	7866	8037	17
27.96	600	9308	9479	17
38.66	700	10701	10897	17
47.18	800	12371	12559	17
46.11	900	13900	14074	18
51.72	1000	15330	15468	18
165.49	2000	31791	32253	25
886.20	4000	65439	66118	30
1337.44	5000	82611	83580	32
5531.86	10000	165630	167064	34

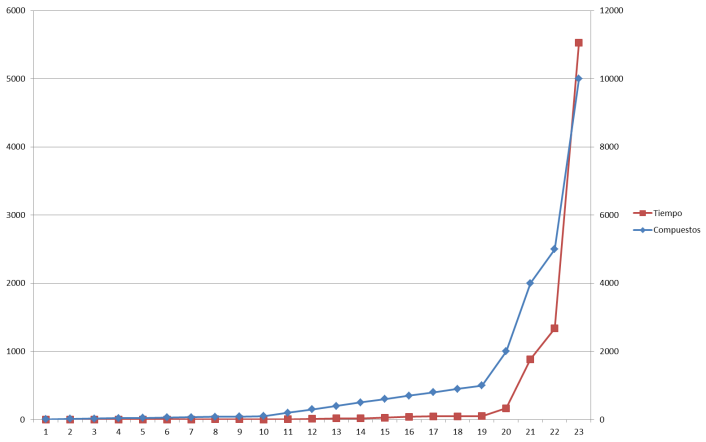


Figura 8. As quantidades de compostos utilizadas versus o tempo em calcular as subestruturas desses compostos. O tempo foi de 5531.86 segundos para 10,000 compostos químicos.

do grafo G entre o tamanho da sub-estrutura S mais o tamanho de G comprimido com S. O tamanho do grafo é a soma de vértices mais arestas. Esta medida é mais eficiente em tempo mas não é tão consistente como a medida *MDL* como se mostra na seguinte tabela 3:

Tabela 3. Se amostra a obtenção dos melhores sub-grafos obtidos pelas medidas *size* e *MDL*. Onde n é a quantidade de compostos químicos.

DTP - Obtenção de melhores sub-grafos com *MDL* e *Size*					
Grafo		MDL		Size	
n	$(\|V\|,\|E\|)$	$(\|V_s\|,\|E_s\|)$	δ_{size}	$(\|V_s\|,\|E_s\|)$	δ_{mdl}
30	(459,481)	(2,1)	0.16	(2,1)	0.33
40	(616,645)	(6,6)	0.16	(3,2)	0.33
50	(785,821)	(6,6)	0.16	(2,1)	0.5
60	(954,1003)	(6,6)	0.16	(3,2)	0.33
70	(1156,1217)	(6,6)	0.16	(5,4)	0.2
80	(1317,1381)	(6,6)	0.16	(5,4)	0.2
90	(1475,1540)	(6,6)	0.16	(5,4)	0.2
100	(1604,1669)	(6,6)	0.16	(5,4)	0.2
200	(3137,3209)	(6,6)	0.16	(2,1)	0.48
300	(4678,4783)	(6,6)	0.15	(2,1)	0.47
400	(6414,6567)	(2,1)	0.48	(2,1)	0.48
500	(7866,8037)	(2,1)	0.48	(2,1)	0.48
1000	(15330,15468)	(2,1)	0.47	(2,1)	0.47

Na figura 9 se observa o sub-gafo obtido de um grafo de 100 compostos, com a medida MDL e com uma medida estrutural, que se encontram na tabela 3. O grafo de 100 compostos pode ser observado na figura 6 (c). A medida MDL obteve um sub-grafo com um pouco mais de arestas e vértices o qual permite fornecer maior informação do composto original. Embora em outros casos, ambas medidas obtiveram sub-grafos semelhantes, em muitos outros casos a medida MDL obteve sub-grafos mais representativos pela extração de sub-grafos maiores em comparação à medida estrutural.

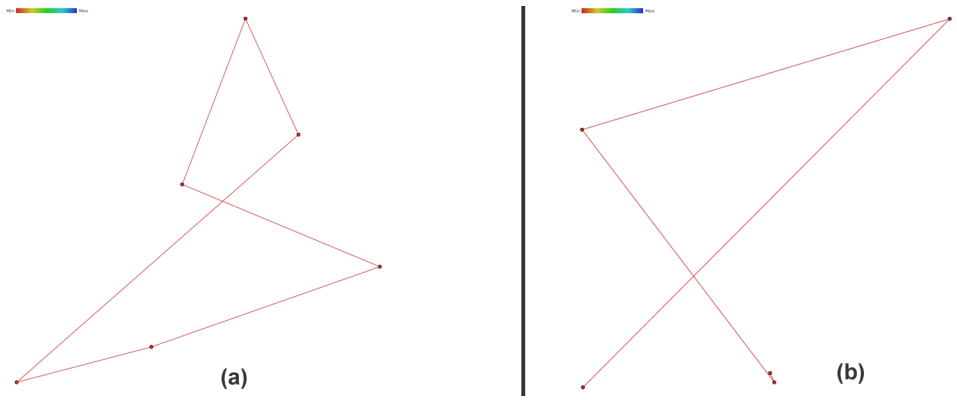


Figura 9. Obtenção do melhor sub-grafo representativo utilizando MDL (a) e utilizando uma medida estrutural (b).

5 Conclusões

- A medida MDL permite obter sub-grafos mais compactos que as medidas estruturais que são baseadas no tamanho do grafo.
- A utilização da medida MDL tem como desvantagem o tempo consumido para análise em redes muito complexas. Como no caso da análise de 10,000 compostos químicos, onde cada composto tinha uma média de 22 conexões (arestas), o tempo foi de mais de uma hora (5531.86 segundos exatamente).
- Os resultados também mostraram que é possível extrair dois sub-estruturas representativas onde interagem os mesmos pares de átomos com dois conexões distintas entre si. Em um caso mais geral, é possível extrair múltiplas patrões de múltiplas arestas entre apenas um par de vértices; o que permite uma análise mais real aos problemas do mundo real representados por grafos.

Referências

1. Remco R. Bouckaert, *Probabilistic network construction using the minimum description length principle*, RUU-CS-94-27 Utrecht University (1994).
2. H. Bunke and G. Allermann, *Inexact graph matching for structural pattern recognition*, Pattern Recognition Letters **1(4)** (1983), 245–253.
3. Jeffrey Coble, Runu Rathi, Diane J. Cook, and Lawrence B. Holder, *Iterative structure discovery in graph-based data*, International Journal on Artificial Intelligence Tools **14** (2005), no. 1-2, 101–124.
4. Diane J. Cook and Lawrence B. Holder, *Substructure Discovery using Minimum Description Length and Background Knowledge*, Journal of Artificial Intelligence Research **1** (1994), 231–255.
5. M. Derthick, *A minimal encoding approach to feature discovery*, In Proceedings of the Ninth National Conference on Artificial Intelligence, 1991, pp. 565–571.
6. Kriegel P. Borgwardt K. HÃ¼bler, C. and Z. Ghahramani, *Metropolis algorithms for representative subgraph sampling.*
7. E.P.D. Pednault, *Part segmentation for object recognition*, Neural Computation **1** (1989), 82–91.
8. A.C. Pifer and L.A. Guedes, *Aprendizagem Estrutural de Redes Bayesianas utilizando mï¿½trica MDL modificada*, IEEE Latin America Transactions **5(8)** (2007).
9. J.R. Quinlan and R.L. Rivest, *Inferring decision trees using minimum description length principle*, Information and Computation **80** (1989), 227–248.
10. J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing Company (1989).
11. J. Segen, *Graph clustering and model learning by data compression*, In Proceeding of the Seventh Conference on Machine Learning (1990), 93–101.
12. K. Thompson and P. Langley, *Conception formation in structured domains*, In D. H. Fisher and R. Pazzani, editors, Concept Formation: Knowledge and Experiencein Unsupervised Learning, chapter 5. Morgan Kaufman Publishers (1991).
13. P.H. Winston, *Learning structure descriptions from examples*, In P.H. Winston, editor, The Psycology of Computer Vision, MacGraw-Hill (1975), 157–210.
14. K. Yoshida, H. Motoda, and N. Indurkhya, *Unifying learning methods by colored digraphs*, In Proceedings of the Learning and Knowledge Acquisition Workshop at IJCAI-93 (1993).