DOI: http://dx.doi.org/10.5769/C2011019

Computação Forense via agrupamento hierárquico de documentos

Luís Filipe da Cruz Nassif[a,b] e Eduardo Raul Hruschka[b,c]
(a) Departamento de Polícia Federal (DPF), São Paulo, Brasil
(b)Universidade de Brasília (UnB), Brasília, Brasil
(c)Universidade de São Paulo (USP), São Carlos, Brasil
nassif.lfcn@dpf.gov.br; erh@icmc.usp.br

Abstract* — *In computer forensic analysis, hundreds of thousands of files are usually analyzed. Most of the data available in these files consists of unstructured text that are hard to be analyzed by human beings. In this context, the use of automated techniques, based on text mining, is of great relevance. In particular, clustering algorithms can help to find new, useful, and potentially actionable knowledge from text files. This work presents an approach that applies document clustering algorithms to forensic analysis of computers seized in police investigations. It was carried out a comparative study of three hierarchical clustering algorithms – Single Link, Complete Link and Average Link – when applied to five textual databases derived from real cases. In addition, it was used the Silhouette relative validity index for automatically estimating the number of groups. To the best of our knowledge, studies of this nature, especially considering the use of hierarchical algorithms and the automatic estimation of the number of clusters, have not been reported in the literature about computer forensics. This study can thus serve as a starting point for researchers interested in developing further research in this particular application domain. In brief, the experiments performed show that the algorithm Average Link provided the best performances. This study also presents and discusses several practical results for both researchers and practitioners of computer forensic analysis.

Keywords* — *Computer Forensics; data clustering; text mining.

Resumo* — *Em análises periciais de computadores, usualmente são examinados centenas de milhares de arquivos. Grande parte dos dados contidos nesses arquivos é constituída por texto não estruturado, cuja análise por parte dos peritos é difícil de ser realizada. Nesse contexto, o uso de métodos automatizados de análise baseados na mineração de textos é de grande interesse. Particularmente, algoritmos de agrupamento podem facilitar a descoberta de conhecimentos novos e úteis nos textos sob análise. Assim, este trabalho apresenta uma abordagem para aplicar agrupamento de documentos em análises periciais de computadores apreendidos em operações policiais. Para ilustrar tal abordagem, foi realizado um estudo comparativo de três algoritmos hierárquicos de agrupamento de dados (Single Link, Complete Link e Average Link) aplicados a cinco bases de dados textuais provenientes de investigações reais. Adicionalmente, foi utilizado o índice de validade relativo Silhueta para estimar automaticamente o número de grupos. Este trabalho estuda algoritmos hierárquicos de agrupamento, ainda não abordados pela literatura sobre Computação Forense. Além disso, estudos relacionados encontrados na literatura se mostram mais limitados do que o estudo aqui apresentado, especialmente ao se considerar que este trabalho considera a estimativa automática do número de grupos. Nesse contexto, o presente estudo poderá servir como ponto de partida para aqueles interessados em desenvolver pesquisas neste domínio de aplicação específico. Os experimentos realizados mostram que o algoritmo hierárquico Average Link proporcionou os melhores resultados. Este estudo também apresenta e discute diversos resultados práticos mais específicos que podem ser úteis para pesquisadores e praticantes de análises forenses computacionais.

Palavras-chave* — *Computação Forense; agrupamento de dados; mineração de textos.

1. Introdução

Estima-se que o volume de dados no universo digital aumentou de 161 hexabytes em 2006 para 988 hexabytes em 2010 [1] – aproximadamente 18 vezes a quantidade de informação presente em todos os livros já escritos – e continua crescendo de forma exponencial. Essa grande quantidade de dados tem impacto direto na área da *Computação Forense*, que pode ser definida como uma área de conhecimento interdisciplinar que combina elementos da ciência da computação e do direito para coletar e analisar dados de sistemas computacionais para produzir provas admissíveis judicialmente. No domínio de aplicação específico aqui estudado, as análises periciais de computadores envolvem examinar centenas de milhares de arquivos por computador, excedendo a capacidade de análise e interpretação dos vestígios por parte do perito e tornando imprescindível o uso de métodos automatizados de análise dos repositórios de dados sob exame. Nesse contexto, o uso de métodos de mineração de dados, com ênfase na descoberta de padrões de conteúdo nos arquivos, é promissor, pois pode contribuir para uma melhor organização dos dados por assunto e, consequentemente, para a descoberta de informações novas e úteis para as investigações. É importante ressaltar que a maior parte (cerca de 80%) dos dados de empresas e organizações, que frequentemente são objeto de exames periciais, se constitui de dados não estruturados [1], formados em grande parte por texto em linguagem natural. Assim, é muito promissora a utilização de métodos de mineração de dados textuais [2] nos exames periciais. Em particular, métodos

para agrupamento de dados textuais são de grande interesse para a perícia computacional, conforme será apresentado neste artigo.

Métodos para agrupamento (*clustering*) de dados têm como objetivo induzir grupos (*clusters*) de dados, de tal forma que objetos pertencentes ao mesmo grupo sejam mais semelhantes entre si do que objetos pertencentes a grupos distintos [3]. Algoritmos de agrupamento são normalmente utilizados em análises exploratórias de dados, nas quais se dispõe de pouco ou nenhum conhecimento sobre os dados [4]. Esse é precisamente o caso encontrado em exames periciais. Nesse tipo de aplicação, bases de dados formadas por objetos com rótulos de classes usualmente não estão disponíveis, pois não se conhece, *a priori*, as classes de documentos que poderiam ser encontradas. Mesmo assumindo que se disponha de uma base de dados rotulada, obtida a partir de perícias anteriores, há poucas chances de que as mesmas classes (possivelmente aprendidas anteriormente por um classificador, num processo de aprendizado supervisionado) continuem sendo válidas para novas amostras de dados, obtidas a partir de outros computadores e vinculados a processos de investigação diferentes. Mais precisamente, a probabilidade de que os dados seriam oriundos de diferentes populações seria alta, o que inviabilizaria a inferência estatística a partir de modelos existentes. Nesse contexto, algoritmos para aprendizado não supervisionado são promissores. Em particular, documentos previamente desconhecidos, mas com mesmo padrão de conteúdo, poderiam ser alocados no mesmo grupo, dessa forma, facilitando a análise dos documentos presentes em computadores apreendidos. Nesse sentido, o perito poderia, após analisar alguns documentos de grupos diferentes, descartar grupos contendo documentos irrelevantes e concentrar a análise nos grupos contendo documentos relevantes. Em outras palavras, o agrupamento de documentos pode evitar o exame de todos os documentos individualmente e, no pior caso, o perito ainda poderia optar por analisá-los em sua totalidade. Assim, a adoção da abordagem pericial baseada em agrupamento de documentos pode melhorar a eficiência do processo de análise pericial de computadores apreendidos, conforme será discutido em maiores detalhes no presente artigo.

Este trabalho investiga uma abordagem baseada em agrupamento de dados para aperfeiçoar análises periciais de computadores. Considerando que métodos para agrupamento de dados têm sido estudados há décadas, e que a literatura sobre o assunto é extensa, optou-se por escolher um conjunto de (três) algoritmos para verificar a viabilidade da abordagem proposta. Particularmente, foram comparados os algoritmos hierárquicos *Single Link*, *Complete Link* e *Average Link* [6]. Dessa forma, pode-se também comparar os desempenhos relativos desses algoritmos ao serem aplicados para analisar conteúdos de arquivos em computadores apreendidos. Todos os algoritmos foram avaliados em bases de dados textuais obtidas a partir de cinco casos reais de perícia realizados na *Polícia Federal*. A fim de tornar a análise comparativa entre os algoritmos mais realista do ponto de vista prático, foi utilizado um índice de validade relativo – Silhueta [5] – para estimar o número de grupos automaticamente a partir dos dados. É importante notar que o número de grupos é um parâmetro crítico de vários algoritmos e é normalmente desconhecido *a priori*. Não foram encontrados na literatura estudos sobre algoritmos de agrupamento hierárquicos aplicados à *Computação Forense*, nem sobre a estimativa automática do número de grupos. Essa característica do estudo realizado é, portanto, uma contribuição importante deste artigo, cujas demais seções estão organizadas como segue. A próxima seção aborda trabalhos relacionados, enquanto que a Seção 3 descreve resumidamente os algoritmos de agrupamento utilizados. A Seção 4 reporta a avaliação experimental realizada e, finalmente, a Seção 5 apresenta as principais conclusões derivadas do trabalho realizado.

2. Trabalhos relacionados

Existem poucos trabalhos na literatura reportando o uso de técnicas de agrupamento de dados para *Computação Forense*. Essencialmente, a maioria dos trabalhos adota alguns algoritmos clássicos para agrupamento de dados - e.g., *Expectation-Maximization* (EM) para aprendizado não supervisionado de mistura de gaussianas, *K-means*, *Fuzzy C-means* (FCM) e redes neurais do tipo mapas auto-organizáveis (*Self Organizing Maps* - SOM). Esses algoritmos possuem propriedades bem conhecidas e são amplamente usados na prática. Mais especificamente, o algoritmo particional *K-means* e o de partições sobrepostas FCM podem ser vistos como casos particulares do EM. Algoritmos do tipo SOM, por sua vez, em geral apresentam bias indutivo semelhante ao *K-means*.

Em [8] redes neurais artificiais do tipo SOM foram usadas no agrupamento de arquivos para auxiliar a tomada de decisão dos peritos e tornar o processo de análise mais eficiente. Os arquivos foram agrupados com base nas suas datas de criação, extensões e horários de criação. Em [9] foi proposto o uso de algoritmos do tipo SOM para agrupar tematicamente os resultados de buscas por palavras chave obtidos durante exames periciais. A hipótese subjacente é que o agrupamento dos resultados aumentaria a eficiência do processo de recuperação da informação, pois não seria necessário a revisão de todos os documentos encontrados pelo usuário. Um ambiente integrado de mineração de *e-mails* para análises periciais, utilizando algoritmos de classificação e agrupamento, foi apresentado em [10]. Posteriormente, foi proposta a extração de características de estilos de escrita de um conjunto de *e-mails* anônimos para posterior agrupamento das mensagens por autor [11]. Foram avaliadas várias combinações de características, como léxicas, sintáticas, estruturais e específicas de domínio, e três algoritmos de agrupamento (*K-means, Bisecting K-means* e *EM*). Agrupamento de dados de e-mails para fins perciais também foi discutido em [12], no qual foi aplicado uma variante do algoritmo *K-Means* baseada em funções do tipo *Kernel*. Os resultados dessa aplicação foram analisados subjetivamente, e os autores concluíram que os resultados são

interessantes e úteis do ponto de vista de uma investigação, pois proporcionam uma visualização geral dos dados, separados por assunto, sem a necessidade de se analisar individualmente o conteúdo de todos os arquivos. Mais recentemente, um método para inferir regras de associação a partir de dados forenses, supostamente fáceis de serem compreendidas por policiais e outros especialistas de domínio, foi descrito em [13]. O método é baseado no algoritmo de agrupamento probabilístico FCM. Os autores afirmam que foram obtidas regras de associação muito boas, mas foi difícil de gerar um significado semântico intuitivo para algumas das relações de pertinência, o que pode prejudicar a compreensão das regras por parte dos especialistas de domínio.

Todos os trabalhos reportados na literatura assumem que o usuário determina, *a priori*, o número de grupos (*clusters*) a serem obtidos. Evidentemente, tal parâmetro é de difícil escolha em situações práticas. No caso dos algoritmos hierárquicos, tal parâmetro não precisa ser determinado previamente. Além disso, essa limitação prática pode ser contornada pelo uso de métodos que basicamente aplicam determinado algoritmo de agrupamento de dados (e.g., *K-Means*) múltiplas vezes, variando-se a quantidade de grupos, e, a partir do conjunto de partições obtidas, escolhem, de acordo com algum critério numérico, a melhor partição dentre aquelas obtidas [14]. O presente trabalho faz uso de tais métodos que permitem estimar automaticamente o número de grupos a partir dos dados, facilitando o trabalho do perito, que, na maior parte das situações, dificilmente saberia estimar, *a priori*, o número de grupos presente em determinada base de dados. Além disso, surpreendentemente, a literatura não faz menção ao uso de algoritmos clássicos de agrupamento hierárquicos aplicados ao domínio de aplicação aqui abordado. O presente estudo considera tais algoritmos clássicos.

3. Pré-processamento e algoritmos

Antes da aplicação dos algoritmos e métodos de agrupamento de dados nas bases textuais, foi necessário realizar um pré-processamento dos documentos. Nessa etapa, foram eliminadas palavras sem significado semântico útil (*stopwords*), como artigos e preposições, e foi utilizado um algoritmo de radicalização (*stemming*) para palavras da Língua Portuguesa. Foi adotada então uma abordagem estatística tradicional para minerar textos. Nessa abordagem, os documentos são representados em um modelo de espaço vetorial [15], no qual cada documento é representado por um vetor que contém as frequências das ocorrências das palavras. Utilizou-se também de uma técnica de redução de dimensionalidade – *Term Variance* (TV) [16] – para aumentar a eficácia e eficiência dos algoritmos de agrupamento. A TV seleciona os atributos (palavras) com maiores variâncias de frequência nos documentos. Para a aplicação dos algoritmos de agrupamento, foi utilizada uma medida de distância entre os documentos baseada no cosseno entre vetores [15] e usualmente utilizada em aplicações de mineração de textos:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - cos(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \frac{x_i \cdot x_j^T}{\|x_i\| \|x_j\|} \quad (1)$$

onde $\mathbf{x}_i$ representa o i-ésimo documento da coleção de textos. Os algoritmos de agrupamento baseados na medida de dissimilaridade definida na Eq. (1) são aplicados às bases de dados. Este trabalho aborda uma categoria de algoritmos clássicos de agrupamento de dados, denominados algoritmos hierárquicos. Os algoritmos hierárquicos fornecem um conjunto de partições aninhadas [4], usualmente representadas sob a forma de uma estrutura hierárquica denominada dendrograma. Um algoritmo hierárquico aglomerativo pode ser descrito genericamente pelos seguintes passos [6]:

1. Seja $\widetilde{N}$ o número de grupos atual. Comece com $\widetilde{N} = N$ e calcule a respectiva a matriz de distâncias;
2. Seja $\boldsymbol{C}_i, 1 \leq i \leq \widetilde{N}$, um grupo qualquer. Faça $\boldsymbol{C}_i = \boldsymbol{C}_i \cup \boldsymbol{C}_j$ tal que $d(\boldsymbol{C}_i, \boldsymbol{C}_j) = d_{min}(\boldsymbol{C}_k, \boldsymbol{C}_l)\ \forall\ k, l \in \{1..N\}, k \neq l$ e descarte o grupo $\boldsymbol{C}_j$;
3. Faça $\widetilde{N} = \widetilde{N} - 1$ e atualize $d(\boldsymbol{C}_i, \boldsymbol{C}_j)\ \forall\ j \in \{1..\widetilde{N}\}$ na matriz de distâncias;
4. Repita os passos 2 e 3 até obter $\widetilde{N} = 1$.

onde *N* representa o número de objetos da base. O conceito de distância entre dois grupos $d(\boldsymbol{C}_i, \boldsymbol{C}_j)$ difere de acordo com o algoritmo hierárquico utilizado. A seguir, são apresentadas as definições de $d(\boldsymbol{C}_i, \boldsymbol{C}_j)$ dos algoritmos usados neste trabalho [6]:

- *Single Link* (SL):

$$d(\boldsymbol{C}_i, \boldsymbol{C}_j) = d_{min}(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in \boldsymbol{C}_i, \boldsymbol{y} \in \boldsymbol{C}_j \quad (2)$$

i.e., a distância entre dois grupos é a distância mínima entre dois objetos pertencentes a grupos diferentes. Esse algoritmo induz grupos contíguos, baseados nas adjacências entre os objetos, podendo assumir formas variadas. Entretanto é sensível a *outliers* e grupos pouco separados, podendo provocar encadeamento ou mistura entre os grupos [3];

- *Complete Link* (CL):

$$d(\boldsymbol{C}_i, \boldsymbol{C}_j) = d_{max}(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in \boldsymbol{C}_i, \boldsymbol{y} \in \boldsymbol{C}_j \quad (3)$$

i.e., a distância entre dois grupos é a distância máxima entre dois objetos pertencentes a grupos diferentes. Essa distância induz grupos de formas globulares. Ela é menos sensível a presença de *outliers*, porém tende a dividir grupos de diâmetros similares [7];

- *Average Link* (AL):

$$d(\boldsymbol{C}_i, \boldsymbol{C}_j) = |\boldsymbol{C}_i|^{-1} |\boldsymbol{C}_j|^{-1} \sum_{x \in \boldsymbol{C}_i, y \in \boldsymbol{C}_j} d(\boldsymbol{x}, \boldsymbol{y}) \quad (4)$$

i.e., a distância entre dois grupos é a distância média entre pares de objetos pertencentes a grupos diferentes. Essa distância representa uma abordagem intermediária entre as duas anteriores (SL e CL). Também induz grupos hiperesféricos, é menos sensível à presença de outliers e tende a formar grupos com diâmetros similares [7];

Em relação ao método para estimativa automática do número de grupos, *K*, foi utilizado o índice de validade relativo denominado Silhueta [5], o qual apresentou resultados muito bons num estudo recente [14]. A Silhueta pode ser definida da seguinte forma: seja *a* a distância média de um objeto $\boldsymbol{x}$ a todos os outros objetos pertencentes ao seu grupo e seja *b* a distância média de $\boldsymbol{x}$ a todos os objetos do grupo vizinho mais próximo. Então, a Silhueta individual do objeto $\boldsymbol{x}$ é definida por:

$$s = (b - a)/max(a, b) \quad (5)$$

Caso o objeto seja o único elemento do seu grupo (*singleton*), arbitrariamente atribui-se o valor "zero" para sua Silhueta [5]. A Silhueta (global) do agrupamento é dada pela média das Silhuetas de todos os objetos. O método para estimativa automática do número de grupos utilizado consiste em avaliar todas as partições aninhadas fornecidas por determinado algoritmo de agrupamento hierárquico – uma para cada valor de *K* – sendo escolhida a partição com maior valor de Silhueta.

4. Experimentos

4.1. Bases de dados

Nos experimentos realizados, foram utilizadas cinco bases de dados textuais obtidas a partir de cinco investigações reais. Cada base de dados foi obtida a partir de um disco rígido diferente, sendo selecionados todos os documentos não duplicados com extensões "doc", "docx" e "odt". Posteriormente, o texto plano dos documentos foi extraído para que pudessem ser aplicados os métodos de mineração de textos. Conforme detalhado na Seção 4.2, a medida de avaliação utilizada pressupõe a existência de uma partição de referência para cada base de dados. Entretanto, como partições de referência dificilmente estão disponíveis em bases provenientes de investigações reais, elas foram definidas manualmente por um especialista de domínio, através da inspeção do conteúdo dos documentos. As bases de dados contêm quantidades variadas de documentos (*N*), grupos (*K*), atributos (*D*), *singletons* (*S*) – grupos unitários – e número de documentos por grupo (#), conforme reportado na Tabela 1.

Tabela 1 – Propriedades das bases de dados utilizadas.

Base	*N*	*K*	*D*	*S*	# Maior Grupo
A	37	23	1744	12	3
B	111	49	7894	28	12
C	68	40	2699	24	8
D	74	38	5095	26	17
E	131	51	4861	31	44

4.2. Medida de avaliação

Do ponto de vista científico, o uso de partições de referência para avaliar algoritmos de agrupamento de dados é considerado o procedimento mais principiado. Nesse caso, as partições de referência são normalmente obtidas a partir de dados gerados sinteticamente, de acordo com alguma distribuição de probabilidades. Do ponto de vista prático, tais partições de referência são usualmente empregadas para se escolher um determinado algoritmo de agrupamento que seja mais apropriado para determinada aplicação, ou para calibrar seus parâmetros. Nesse caso, uma partição de referência é construída por um especialista de domínio e reflete as expectativas que ele tem sobre os grupos que deveriam ser encontrados numa determinada amostra da base de dados. Nesse sentido, o método de avaliação experimental utilizado neste trabalho se baseia no índice de validação externo *Ajusted Rand Index* (ARI) [17], o qual mede a correspondência entre a partição $\boldsymbol{P}$ obtida por um algoritmo de agrupamento e uma partição de referência $\boldsymbol{R}$. Mais formalmente, o ARI$\in$ [0,1] é definido como:

$$ARI = \left(a - \frac{(a+c)(a+b)}{a+b+c+d}\right) / \left(\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{a+b+c+d}\right) \quad (6)$$

onde *a* é o número de pares de objetos pertencentes ao mesmo grupo em $\boldsymbol{P}$ e em $\boldsymbol{R}$, *b* é o número de pares de objetos pertencentes ao mesmo grupo em $\boldsymbol{P}$ e a grupos diferentes em $\boldsymbol{R}$, *c* é o número de pares de objetos pertencentes a grupos diferentes em $\boldsymbol{P}$ e ao mesmo grupo em $\boldsymbol{R}$ e *d* é o número de pares de objetos pertencentes a grupos diferentes em $\boldsymbol{P}$ e em $\boldsymbol{R}$.

3.2. Resultados e discussões

De um modo geral, o algoritmo hierárquico *Average Link* (AL) apresentou os melhores resultados, tanto em relação à média de ARI, quanto em relação ao seu desvio padrão, o que indica uma maior consistência e estabilidade de resultados, conforme pode ser observado na Tabela 2. Note que valores de ARI próximos de 1 indicam que a partição obtida é bastante aderente à partição de referência. O algoritmo *Complete Link* (CL) apresentou desempenho muito próximo ao *Average Link*. Entretanto, o algoritmo *Single Link* (SL) obteve um desempenho relativamente inferior aos outros dois algoritmos hierárquicos, principalmente nas bases A e B. Isso pode ter sido provocado pela sensibilidade do *Single Link* à presença de *outliers*, que podem ocasionar um encadeamento de objetos durante o agrupamento, mesclando grupos diferentes [3] quando eles não são bem separados.

Tabela 2 – Valores de ARI e desvio padrão obtidos pelos algoritmos

Algoritmo	Base A	Base B	Base C	Base D	Base E	Média	σ
AL	0,94	0,83	0,89	0,99	0,90	0,91	0,06
CL	0,94	0,76	0,89	0,98	0,90	0,89	0,08
SL	0,54	0,63	0,90	0,98	0,88	0,79	0,19

Em relação ao índice de validade relativo Silhueta, utilizado para a estimativa automática do número de grupos, foram obtidos resultados muito promissores, conforme pode ser observado na Tabela 3. Mais especificamente nas bases de dados A, C e D, a Silhueta obteve estimativas de *K* muito próximas do número de grupos da partição de referência para os algoritmos hierárquicos A*verage Link* e *Complete Link*. Mas note que nem sempre melhores estimativas do número de grupos correspondem a valores mais altos de ARI, conforme

pode ser observado nos resultados para as bases C e E nas Tabelas 2 e 3.

Tabela 3 – Números de grupos estimados pela Silhueta para cada base e algoritmo

Algoritmo	Base A (K=23)	Base B (K=49)	Base C (K=40)	Base D (K=38)	Base E (K=51)
AL	24	60	43	36	32
CL	24	63	43	35	30
SL	19	77	47	41	56

De um ponto de vista prático, vários resultados interessantes foram obtidos em nosso estudo. Em todas as bases de dados utilizadas, as melhores partições foram formadas na sua maior parte por grupos contendo ou documentos relevantes ou documentos irrelevantes. Por exemplo, na base de dados A, o algoritmo AL obteve uma partição dos dados formada por alguns *singletons* e por outros 11 grupos de documentos, cujos conteúdos estão descritos na Tabela 4, e na base de dados C, o algoritmo AL também obteve uma partição dos dados formada por alguns *singletons* e por outros 15 grupos de documentos, cujos conteúdos estão descritos na Tabela 5.

Tabela 4 - Amostra das informações dos grupos encontrados na base A

Cluster	Informação
A_1	02 documentos de licença e ajuda de software
A_2	02 orçamentos de manutenção de veículo
A_3	02 Documentos de Arrecadação de Receitas Federais
A_4	02 propostas de renovação de seguros
A_5	02 relatórios de serviço de manutenção em embarcação
A_6	02 propostas de venda de condicionador de ar
A_7	03 recibos de pagamento
A_8	02 panfletos sobre fragmentadoras de papel
A_9	02 modelos de etiquetas da empresa
A_{10}	03 modelos de envelope da empres
A_{11}	02 documentos provenientes de SPAM

Tabela 5 – Amostra das informações dos grupos encontrados na base C

Cluster	Informação
C_1	3 documentos em branco
C_2	4 permissões para transações bancárias
C_3	2 informativos sobre salário-maternidade
C_4	2 listas de alimentos
C_5	1 aviso sobre operações cambiais 1 lista de documentos para atualização de cadastro
C_6	2 documentos de ratificação de operações cambiais
C_7	1 modelo de ficha cadastral de corretora de valores 1 modelo de contrato com a corretora
C_8	1 estatuto de clube de investimento 1 termo de adesão a clube de investimento
C_9	2 modelos de registro de movimentação maior que R$ 100k
C_{10}	8 comprovantes de operações cambiais de compra e venda
C_{11}	2 avisos de corretora sobre horário de expediente
C_{12}	3 modelos de etiquetas
C_{13}	1 aviso sobre horário de expediente 1 recibo de cheque
C_{14}	2 relatórios diários de compra e venda de câmbio
C_{15}	2 documentos de exemplo de aplicativo de escritório

Por questões de confidencialidade dos dados, não podem ser reveladas informações mais detalhadas sobre os grupos obtidos. Entretanto, pode ser mencionado que, nessas duas investigações reais de crimes financeiros, foram obtidos grupos contendo apenas documentos relevantes, tais como os grupos C_{10}, C_{14} e C_2 na base C e A_3 e A_7 na base A. Também foram obtidos grupos contendo apenas documentos irrelevantes, como o grupo C_{12} na base C e o grupo A_{11} na base A. Assim, o agrupamento de documentos se mostra eficaz em separar os documentos sob análise em grupos de documentos relevantes e grupos irrelevantes para a investigação. Nesse sentido, o perito poderia se concentrar inicialmente em analisar documentos representativos de cada grupo e, a partir dessa análise preliminar, eventualmente decidir pelo exame detalhado dos demais documentos de cada grupo em questão. Num cenário mais prático e realista, no qual especialistas de domínio (e.g., peritos criminais) são escassos e dispõem de tempo limitado, é razoável assumir que, após encontrar um documento relevante (e.g., em uma busca por palavras-chave), o perito poderia priorizar a análise dos outros documentos pertencentes ao grupo do documento encontrado, pois é provável que tais documentos também sejam relevantes para a investigação. Assim, o agrupamento de documentos se apresenta muito útil em análises periciais de computadores, ajudando os peritos a conduzir os exames de uma forma mais eficiente, focando nos documentos mais relevantes, sem precisar inspecionar todos os documentos individualmente.

Finalmente, uma característica desejável dos algoritmos hierárquicos que os tornam particularmente interessantes durante um exame pericial é a visualização resumida dos dados na forma de um dendrograma. Em um dendrograma, o nó raiz representa o conjunto completo de dados, enquanto as folhas representam cada objeto individual. Um nó interno representa um grupo formado pela união de dois grupos, unitários ou não. A altura de um nó interno é proporcional à distância entre os dois grupos que ele une. Essa representação visual provê uma descrição muito informativa da estrutura

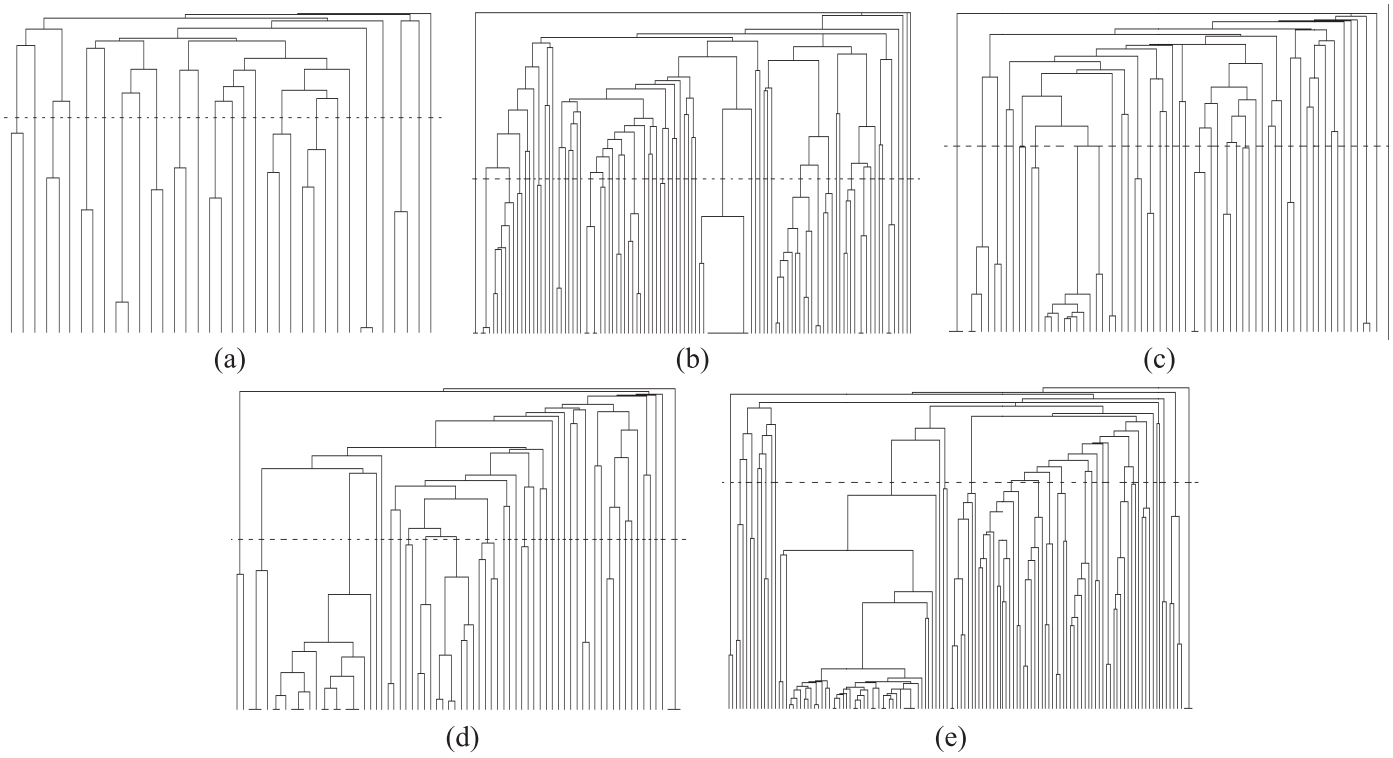

Figura 1 – Dendrogramas obtidos pelo algoritmo *Average Link* para as bases de dados A (a), B (b), C (c), D (d) e E (e).

dos dados, podendo ser utilizada na visualização de dados de alta dimensionalidade, como os documentos textuais analisados durante os exames periciais, e na análise dos padrões descobertos. Para ilustrar melhor essa forma de visualização, a Figura 1 apresenta os dendrogramas obtidos pelo algoritmo *Average Link* para as cinco bases de dados. Os dendrogramas foram cortados horizontalmente por uma linha tracejada indicando a partição aninhada correspondente ao número de grupos selecionado pela Silhueta. É importante destacar que sub-árvores com pequena altura e grande largura representam grupos coesos – contendo documentos muito similares – e numerosos, respectivamente. Tais grupos são bons candidatos para uma análise inicial por parte do perito. Além disso, uma abordagem de análise possível por parte do perito é, após encontrar um grupo de documentos relevantes, analisar o grupo de documentos mais semelhante ao grupo encontrado, pois é possível que ele também seja importante. Isso pode ser realizado "subindo" pelo dendrograma, i.e., acessando o grupo correspondente ao nó cujo pai seja o mesmo pai do nó correspondente ao grupo de documentos relevante encontrado anteriormente.

5. Conclusão

Os resultados obtidos neste trabalho mostram que o uso de agrupamento de dados para auxiliar a análise pericial é promissor, podendo contribuir com a descoberta de informações novas e úteis para as investigações, ao mesmo tempo em que pode acelerar de maneira significativa a atividade pericial, separando os documentos em grupos relevantes e irrelevantes sob o ponto de vista pericial. Nos experimentos realizados, de modo geral, o algoritmo hierárquico *Average Link* apresentou os melhores desempenhos e uma melhor estabilidade de resultados. Além disso, foi apresentado como os algoritmos hierárquicos podem facilitar o processo de análise durante os exames periciais. Os algoritmos hierárquicos apresentam diversas características interessantes para este domínio de aplicação específico, pois o resultado do agrupamento é independente da ordem de apresentação dos objetos, conseguem lidar com a presença de *outliers*, fornecem uma visualização final intuitiva e de fácil interpretação pelo usuário e não é necessário fixar o número de grupos previamente, diferentemente de outros algoritmos abordados na literatura sobre Computação Forense. O número de grupos pode ser definido pelo usuário pela inspeção visual do dendrograma ou com o auxílio de métodos numéricos, através de um corte horizontal do dendrograma. Em relação ao índice relativo utilizado para a estimativa automática do número de grupos, a Silhueta apresentou resultados muito promissores, conseguindo boas aproximações para esse parâmetro crítico, que dificilmente seria conhecido *a priori* durante um exame pericial.

Com o objetivo de adicionalmente alavancar o uso de algoritmos de agrupamento de dados em tais aplicações, sugere-se como trabalho futuro investigar formas de se rotular, automaticamente, os grupos obtidos. A atribuição de rótulos aos grupos permitiria ao perito identificar com maior rapidez o conteúdo semântico de cada grupo, mesmo antes de analisar o conteúdo dos documentos propriamente ditos. Por fim, a avaliação de algoritmos que induzem partições sobrepostas (e.g., *Fuzzy C-Means* e *EM*) e que atribuem aos documentos probabilidades de pertinência a cada grupo é interessante.

Agradecimentos

L. F. C. Nassif agradece ao Departamento de Polícia Federal e à Universidade de Brasília pelo apoio à pesquisa e ao Ministério da Justiça pelo financiamento por meio do Programa Nacional de Segurança Pública com Cidadania (PRONASCI). E. R. Hruschka agradece as agências de financiamento de pesquisa CNPq e FAPESP.

Referências

[1] Gantz, J. F., Reinsel, D., Chute, C., Schlichting, W., McArthur, J., Minton, S., Xheneti, I., Toncheva, A., e Manfrediz, A., The expanding digital universe: A forecast of worldwide information growth through 2010, External Publication of IDC Information and Data, (2007), 1 – 21.

[2] Ebecken, N. F. F., Lopes, M. C. S., e Costa, M. C. A., Mineração de textos, Em Rezende, S. O., Sistemas Inteligentes: Fundamentos e Aplicações, Manole, (2003), 337 – 372.

[3] Everitt, B. S., Landau, S., e Leese, M., Cluster Analysis, Arnold, (2001).

[4] Jain, A. K., Dubes, R. C., Algorithms for Clustering Data, Prentice-Hall., (1988).

[5] Kaufman, L, e Rousseeuw, P., Finding Groups in Data: An introduction to cluster analysis, Wiley-Interscience, (1990).

[6] Xu, R., Wunsch II, D. C., Clustering, Wiley / IEEE Press, (2009).

[7] Tan, P., Steinbach, M., e Kumar, V., Introduction to Data Mining, Addison-Wesley, (2006), pp. 515-526.

[8] Fei, B.K.L., Eloff, J.H.P., Venter, H.S. e Oliver, M.S., Exploring Forensic Data with Self-Organizing Maps, IFIP International Conference on Digital Forensics, (2005), 113 – 123.

[9] Beebe, N. L., Clark, J. G., Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results, Digital Investigation, Elsevier, 4:1 (2007), 49 – 54.

[10] Hadjidj R., Debbabi, M., Lounis, H., Iqbal, F., Szporer, A., Benredjem, D., Towards an integrated e-mail forensic analysis framework, Digital Investigation, Elsevier, 5:3-4 (2009), 124 – 137.

[11] Iqbal, F., Binsalleeh, H., Fung, B. C. M., Debbabi, M., Mining writeprints from anonymous e-mails for forensic investigation. Digital Investigation, Elsevier, 7:1-2 (2010), 56 – 64.

[12] Decherchi, S., Tacconi, S., Redi, J., Leoncini, A., Sangiacomo, F., Zunino, R., Text Clustering for Digital Forensics Analysis, Computational Intelligence in Security for Information Systems, (2009), 29 – 36.

[13] Stoffel, K., Cotofrei, P., Han, D., Fuzzy Methods for Forensic Data Analysis, IEEE International Conference of Soft Computing and Pattern Recognition, (2010), 23 – 28.

[14] Vendramin, L., Campello, R. J. G. B., Hruschka, E. R., Relative clustering validity criteria: A comparative overview, Statistical Analysis and Data Mining, 3 (2010), 209 – 235.

[15] Salton, G. e Buckley, C., Term weighting approaches in automatic text retrieval, Information Processing and Management, (1988).

[16] Liu, L., Kang, J., Yu, J., e Wang, Z., A comparative study on unsupervised feature selection methods for text clustering, IEEE International Conference on Natural Language Processing and Knowledge Engineering, (2005), 597-601.

[17] Hubert, L., e Arabie, P., Comparing partitions, Journal of Classification, 2 (1985), 193 – 218.