Modelo de espaço vetorial para trace clustering utilizando Doc2Vec

1st Carlos Eduardo Ferreira
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
03828-000 - Sao Paulo - SP - Brasil
cadu.ferreira@usp.br

2nd Gilmer Gomes
Escola de Artes, Ciências e Humanidades
Universidade de São Paulo
03828-000 - Sao Paulo - SP - Brasil
gilmer.gomes09@usp.br

***Abstract*—Process Mining has established itself as a technique for analyzing operational processes, based on event logs, to extract insights and propose improvements. Despite this, some processes are difficult to analyze due to their complex nature, often unstructured or knowledge intensive. In these scenarios trace clustering is used to break the event logs into sublogs making the analysis process easier. In this paper, we analyze the Vector space models technique, using the Doc2Vec for vector embedding, with an agglomerative clustering algorithm on real and synthetic data. The results obtained confirm that the method in question can be used to deal with complex "spaghetti" processes, but attention to the hyperparameters of the experiments must be observed.**

Index terms—process mining, trace clustering, vector space models, Doc2Vec, hyperparameters)

I. Introdução

A tarefa principal do Process Mining é, através de análises de log de eventos de processos de negócio, conseguir extrair informações úteis como conformidade de execução e até propor melhorias. Com base nessas análises é possível identificar como, por exemplo, os processos realmente funcionam, apontar desvios do que está previsto, validar conformidade, encontrar gargalos ou oportunidades de melhorias e até fazer previsões sobre conclusão de uma tarefa.

No entanto, é comum estarmos diante de um processo que é complexo e difícil de analisar. Chamamos esse tipo de processo como "spaghetti" devido ao emaranhado de ligações entre as atividades do processo.

A técnica de trace clustering visa a ajudar a entender essa complexidade atuando como um passo de pré-processamento que permite agrupar sequências de eventos (traces) semelhantes e permite apontar variações ou anomalias.

Nosso trabalho explora a técnica já bem documentada pela literatura da área que usa modelo de espaço vetorial como em [1][2]. Nosso foco, porém, é utilizar o Doc2Vec para embedding de vetores e analisar como os hiperparâmetros envolvidos podem alterar as métricas de qualidade da clusterização.

II. Fundamentação Teórica

A. Process Mining

Process mining é um conjunto de técnicas que extraem conhecimento de logs de eventos registrados por sistemas de informação. Envolve a descoberta, monitoramento e melhoria de processos reais por meio da análise de dados de eventos.
Mais formalmente, a mineração de processos é a aplicação de técnicas de mineração de dados para descobrir, monitorar e melhorar processos, extraindo conhecimento de logs de eventos [3].

Em Process Mining, os conceitos de evento, atividade, case e trace são fundamentais para compreender como os dados são coletados, analisados e transformados em insights sobre os processos de negócio.

Um evento representa uma ação específica que ocorre durante a execução de um processo. É o registro de algo que aconteceu em um determinado momento. Em um processo de aprovação de empréstimo, um evento pode ser: "Pedido de empréstimo submetido", "Análise de crédito iniciada", "Empréstimo aprovado", etc. Um evento é composto por atributos como Identificador, Atividade, Recurso, Data etc.
Uma atividade é uma tarefa ou passo específico dentro de um processo. Um case representa uma instância única de um processo. É como um "pedido" ou "caso" específico que passa por todas as etapas do processo.
Um trace é a sequência de eventos que ocorrem durante a execução de um case. É o caminho que um case percorre dentro do processo.

B. Trace clustering

Trace clustering é uma técnica usada em mineração de processos para agrupar sequências semelhantes de eventos, perfil de traces [4], em clusters. Ao identificar esses padrões, podemos obter insights valiosos sobre o comportamento, variações e anomalias do processo.

Recursos relevantes são extraídos de cada rastreamento, como sequências de atividades, durações e atribuições de recursos. Uma métrica de similaridade (por exemplo, distância de edição, similaridade de Jaccard) é usada para medir a

similaridade entre traces. Um algoritmo de agrupamento (por exemplo, k-means, agrupamento hierárquico) é aplicado para agrupar traços semelhantes em clusters. Na Fig. 1 podemos verificar os modelos de processos de um log de evento completo e de sublogs.

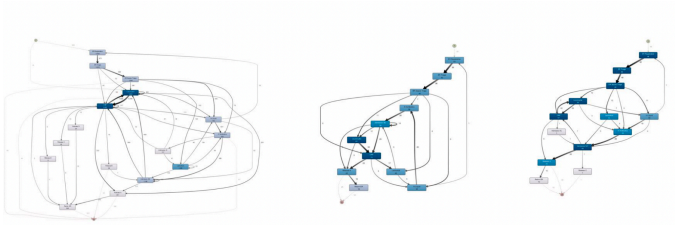

Fig. 1. Modelos de processos de um log de evento completo e de sublogs. Exemplo

Como mencionado em [2] e [5] estratégias de trace clustering que têm sido usadas em process mining podem ser dividas em três não excludentes categorias: *trace sequence similarity*, que é baseada na sintaxe do trace e na ordem das atividades, como uma sequências; *model similarity*, que é baseada não diretamente no trace, mas na qualidade do modelo de processo descoberto pelos traces; *feature vector similarity*, na qual os traces são mapeados para um espaço vectorial extraindo características específicas do perfil, como atividade, transição, recurso etc. Nosso trabalho utiliza essa última estratégia.

C. *Modelos de espaços vetoriais*

Modelos de espaços vetoriais são representações matemáticas que mapeiam objetos como palavras, documentos, imagens etc, para pontos em um espaço vetorial de alta dimensão. Cada dimensão desse espaço corresponde a uma característica ou atributo do objeto.

Seja X um conjunto de objetos. Um modelo de espaço vetorial φ é uma função que mapeia cada objeto (no nosso caso documento) $d \in D$ em um vetor $v \in \mathbb{R}^n$, onde $\mathbb{R}^n$ é o espaço euclidiano n-dimensional:

$\varphi: D \rightarrow \mathbb{R}^n$
$d \mapsto v = [v_1, v_2, ..., v_\square]$

D. *Doc2Vec*

Doc2vec [6] é uma extensão do algoritmo Word2Vec, que gera representações vetoriais para palavras. No entanto, enquanto Word2Vec foca em representar palavras individuais, Doc2Vec visa representar documentos inteiros. Existem duas variantes principais. ***CBOW*****,** onde um conjunto contínuo de palavras cria uma janela variável em torno da palavra atual, para fazer predição a partir do "contexto" em relação às palavras ao redor. Cada palavra é representada como um vetor de características. Após o treinamento, esses vetores se tornam vetores de palavras. A outra variante é a *Skip-Gram* que é na verdade o oposto de *CBOW*: em vez de prever uma palavra de cada vez, usa 1 palavra para prever todas as palavras vizinhas ("contexto"). Skip gram é muito mais lento que o *CBOW*, mas é considerado mais preciso com palavras pouco frequentes.

III. Trabalhos relacionados

O trabalho mencionado em [1] foi o precursor no uso de modelos de aprendizado para gerar embeddings para o uso em process mining. Neste trabalho os autores adaptaram as implementações de Word2Vec e Doc2Vec para o que eles chamam de Act2Vec e Trace2Vec para uso e incorporação de informações de atividades e traces respectivamente. Os resultados demonstram que Trace2Vec gera clusters com qualidade comparável com métodos tradicionais de representação.

Um estudo comparativo usando modelos de espaço vetorial foi realizado em [2]. Nesse estudo os autores comparam quatro modelos para trace clustering usando algoritmo de clustering aglomerativo, k-means, e os resultados indicam que modelos baseados em embeddings podem lidar com o problema de trace clustering em processos desestruturados.

IV. Metodologia

Neste trabalho realizamos experimentos tanto com logs sintéticos quanto com logs reais. O trabalho pode ser reproduzido através do repositório deste projeto[1]. Primeiro extraímos os traces relacionados aos cases e transformamos em um documento que pode ser convertido em vetor usando Doc2Vec como modelo de espaço, esse passo é conhecido como embedding.

Depois realizamos a clusterização de cada vetor e recuperação da informação quantitativa sobre os clusters e traces aglutinados em cada cluster.

Em seguida realizamos a extração dos sublogs dos perfis representativos de cada cluster para avaliação e aplicar técnicas de discovery para o modelo de processo em questão.

Para a avaliação da qualidade de clusterização utilizamos índice de silhouette que é definido da seguinte forma [7] :

$$SI(D,C) = \frac{1}{|D|} \sum_{o \in D} \frac{b(o,C) - a(o,C)}{max\{b(o,C),\, a(o,C)\}},$$

onde D é um conjunto não vazio, $C = \{C1, ..., Ck\}$ é o particionamento de D, $a(o, C)$ é a média Euclidiana entre o e outros objetos no cluster no qual o pertence, e $b(o, C)$ é a distância média mínima entre o para todos os objetos no cluster no qual o não pertence.

1. https://github.com/eduprivate/pm-vsm4trace-clustering

A. *Log de Eventos*

Para este trabalho utilizamos tanto log de eventos reais quanto sintéticos. O log de eventos real é derivado de um conjunto de dados de ações judiciais empresariais do Tribunal de Justiça do Estado de São Paulo, o maior tribunal do mundo [8]. Este log contém 430 atividades únicas, 4795 cases. Podemos verificar que o processo teve menos atividades entre os anos 2010 e 2016, indicando um provável período de implementação. A partir de 2016 a quantidade de eventos tem um enorme crescimento, conforme a Fig. 2.

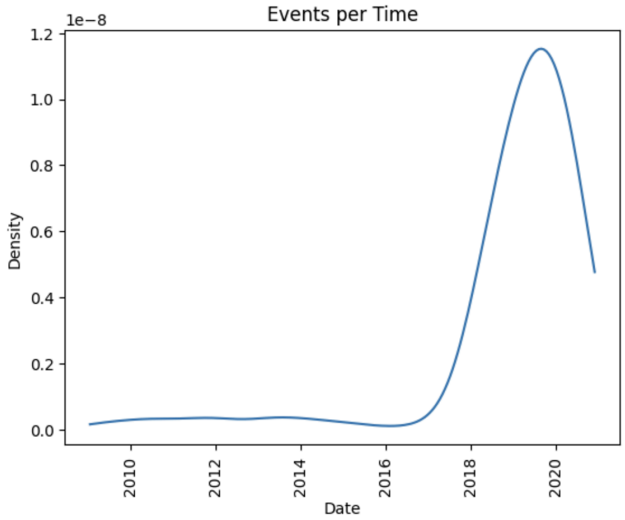

Fig. 2. Densidade de evento ao longo do tempo

Este log de eventos tem características “spaghetti” e são ideais para o tipo de análise que este trabalho se propõe. Na Fig. 3 podemos ver um recorte do modelo de processo extraído usando api pm4py[2] e para ilustrar a natureza complexa desse modelo de processo, podemos verificar pelo gráfico DFG na Fig. 4 e a Fig. 5, de difícil visualização e análise.

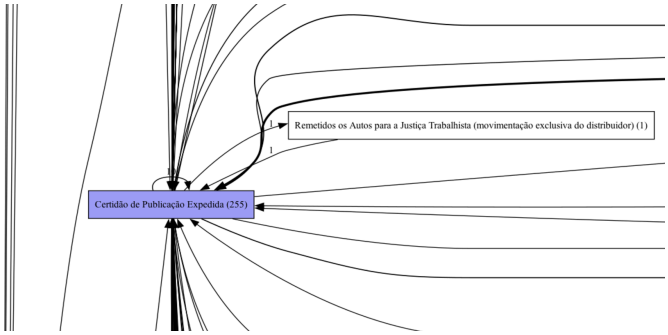

Fig. 3. Recorte de visão do DFG, detalhe de atividade

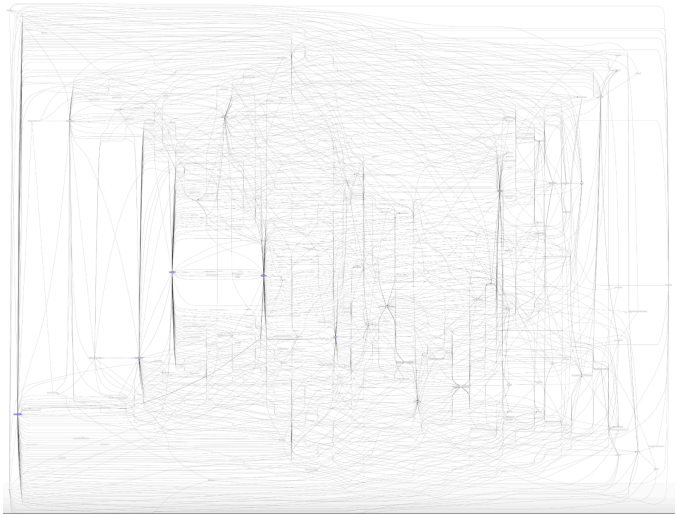

Fig. 4. Modelo de processo completo

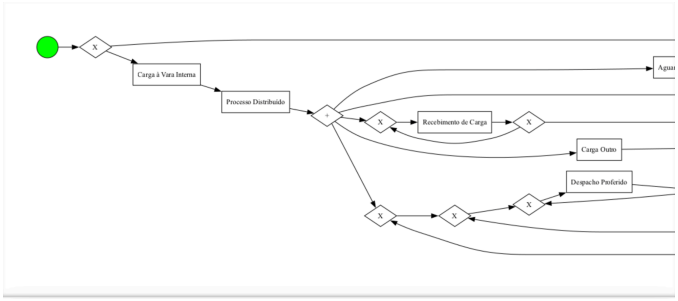

Fig. 5. Recorte de visão do modelo de processo

Os logs sintéticos foram gerados pela ferramenta BIMP Simulator[3], baseado no modelo de processo extraído do log de eventos reais e tem 430 atividades únicas, 4795 cases. As probabilidades de cada gateway foram calculadas pela ferramenta conforme modelo de processo. Além disso foi adicionado a esse log uma anomalia, um trace com sequência incomum.

B. *Configuração de Experimentos*

Os experimentos foram realizados pensando em testar como os hiperparâmetros afetam o índice de silhouette. Dessa forma realizamos os seguintes passos: (a) extraímos os traces do log de eventos e transformamos em um documento aceito pelo modelo; (b) mapeamos os documento no modelo de espaço vetorial e iteramos sobre a lista de documentos mapeando cada um gerando um representação vetorial; (c) aplicamos o algoritmo de clusterização k-means[4] e aplicamos algoritmo PCA para redução de dimensionalidade para visualização dos clusters.

O modelo Doc2Vec suporta um conjunto de hiperparâmetros e os experimentos foram projetados para testar em conjunto algumas variações testes.

2. https://github.com/pm4py
3. https://bimp.cs.ut.ee/simulator
4. https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html

Como o modelo em questão é um modelo não supervisionado as técnicas de otimização de hiperparâmetros torna-se mais complicada e por isso os experimentos visam testar um conjunto pré-selecionado destes e executar os passos anteriormente mencionados e analisar os resultados.

A seleção prévia dos hiperparâmetros de testes foram selecionados baseados em estudos prévios do modelo em questão como em [9] e testes empíricos realizados durante esse trabalho. A Tabela 1 resume os a configuração dos experimentos.

Parametros	Valores
Algoritmo	PV-DBOW
Embedding vector size	[2, 3, 4, 5, 6, 7]
Alpha	[0.001, 0.005, 0.01, 0.025]
Alpha-min	[0.001, 0.005, 0.01, 0.025]
epochs	[5, 10, 20, 30]
window	[3, 5, 7, 10]
k (número clusters)	[2, 3, 4, 5, 6, 7]

Tabela 1. Valores dos hiperparâmetros utilizados nos experimentos

O algoritmo de clusterização k-means utiliza a função de similaridade de distância Euclidiana e o número de clusters é o parâmetro k da Tabela 1. Uma vez calculados os clusters obtemos os k centróides e verificamos quais são os vetores mais próximos e com base em sua localização no dataframe que durante o processo guarda a informação de traces e ids, então extraímos os sublogs mais representativos de cada cluster. Com base no sublog podemos realizar o discovery do modelo de processo referente ao sublog.

V. RESULTADOS E ANÁLISES

Nossa análise é baseada na métrica de silhouette. Primeiro avaliamos quais os hiperparâmetros ideias e em seguida aplicamos o trace clustering nos logs sintéticos e reais, extraindo desses os traces mais significativos.

A análise de correlação entre hiperparâmetros não demonstra uma correlação direta forte, conforme a Fig. 6. Dessa forma, o que podemos fazer é ordenar os resultados pelo índice de silhouette recuperando os valores que chegaram a esse resultado do experimento.

	k	vector_size	alpha	alpha_min	epochs	window	silhouette_score
k	1.000000	0.000000	0.000000	0.000000	0.000000	-0.000000	-0.376535
vector_size	0.000000	1.000000	0.000000	0.000000	-0.000000	-0.000000	-0.521001
alpha	0.000000	0.000000	1.000000	0.000000	-0.000000	-0.000000	0.066314
alpha_min	0.000000	0.000000	0.000000	1.000000	0.000000	-0.000000	0.091447
epochs	0.000000	-0.000000	-0.000000	0.000000	1.000000	-0.000000	0.182728
window	-0.000000	-0.000000	-0.000000	-0.000000	-0.000000	1.000000	0.006581
silhouette_score	-0.376535	-0.521001	0.066314	0.091447	0.182728	0.006581	1.000000

Fig. 6. Correlação entre hiperparâmetros

Através dos box-plots podemos ver como o índice de silhouette varia conforme variam os valores dos hiperparâmetros, Fig. 7, Fig. 8 e Fig. 9. Já na Tabela 2 estão as estatísticas do silhouette_score.

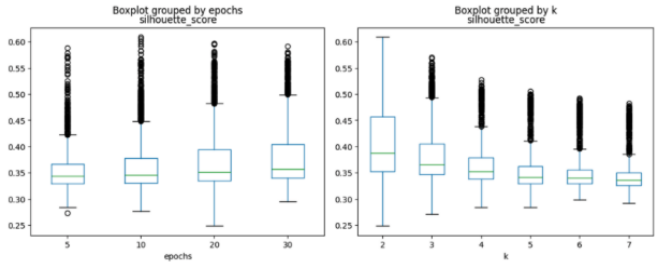

Fig. 7. Box-Plots [epochs, k] x silhouette_score

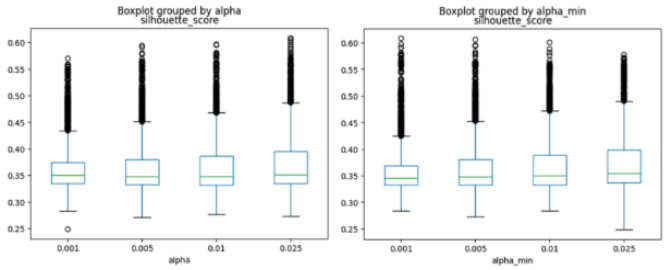

Fig. 8. Box-Plots [alpha, alpha_min] x silhouette_score

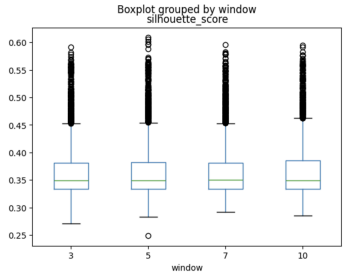

Fig. 9. Box-Plots window x silhouette_score

	silhouette_score
count	9216.000000
mean	0.369092
std	0.055417
min	0.248468
25%	0.333305
50%	0.349364
75%	0.382441
max	0.608946

Tabela 2. Estatísticas silhouette_score

Com base na Tabela 3, podemos estabelecer que os valores de hiperparâmetros ideias são: *vector_size = 2, alpha = 0.025, alpha_min = 0.001, epochs = 10, window = 5* e *k = 2*.

	k	vector_size	alpha	alpha_min	epochs	window	silhouette_score
1182	2	2	0.025000	0.001000	10	5	0.608946
1278	2	2	0.025000	0.005000	10	5	0.606245
1374	2	2	0.025000	0.010000	10	5	0.601313
822	2	2	0.010000	0.001000	20	5	0.597362
1284	2	2	0.025000	0.005000	10	7	0.596182
918	2	2	0.010000	0.005000	20	5	0.595572
2082	2	3	0.005000	0.005000	20	10	0.594706
1992	2	3	0.005000	0.001000	30	3	0.591449
1194	2	2	0.025000	0.001000	10	10	0.591418
9030	2	7	0.025000	0.010000	5	5	0.588566

Tabela 3. Primeiras 10 linhas ordenadas pelo índice de silhouette

A. Log de evento sintético

Como dito na seção *IV A, os logs* sintéticos foram gerados pela ferramenta BIMP Simulator, baseado no modelo de processo extraído do log de eventos reais.

Utilizando os valores recuperados anteriormente executamos a etapa de trace clustering dos logs sintéticos e trazemos aqui os resultados.

Podemos verificar pela Fig. 10 os dois clusters e a partir das centróides extraímos os traces mais significativos, e utilizamos a ferramenta Disco para gerar o gráfico DFG, conforme Fig 11.

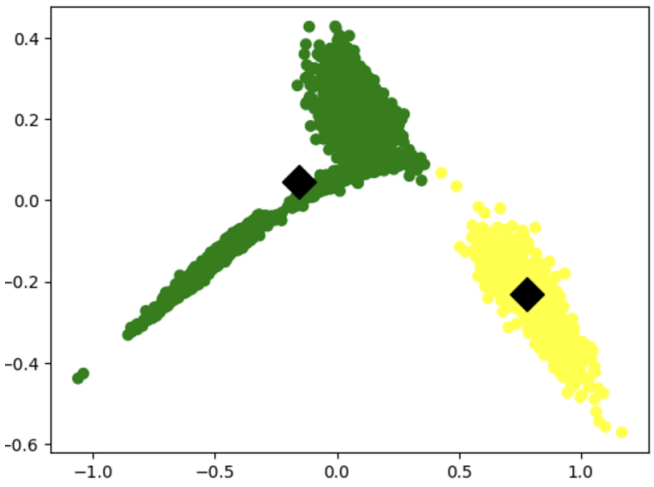

Fig. 10. Cluster k = 2, log sintético

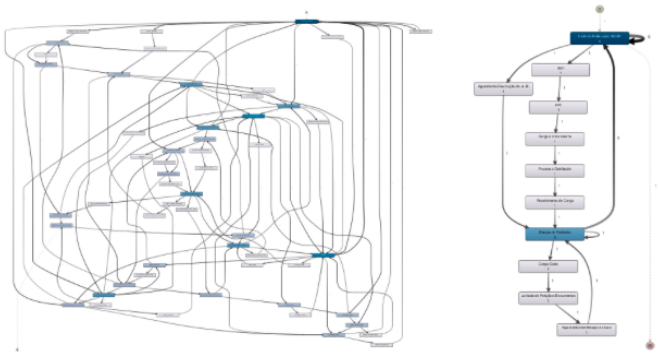

Fig. 11. Traces mais significativos extraídos de cada cluster, log sintético

Apenas como referência, trocando o algoritmo do Doc2Vec para a variante Skip-gram, mantendo os mesmo valores para os hiperparâmetros, obtemos uma clusterização diferente e com silhouette score menor (0.41), conforme Fig. 12.

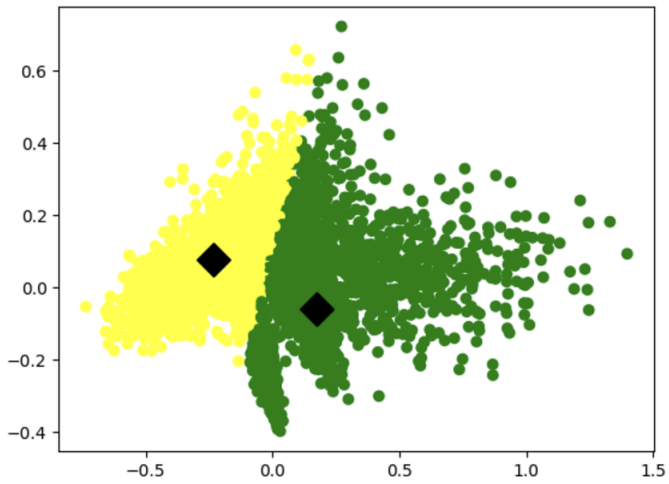

Fig. 12. Variante Skip-gram, log sintético

B. Log de evento real

De forma semelhante aplicado ao log sintético fizemos com o log real.

Podemos verificar pela Fig. 13 os dois clusters e a partir das centróides extraímos os traces mais significativos, conforme Fig. 14.

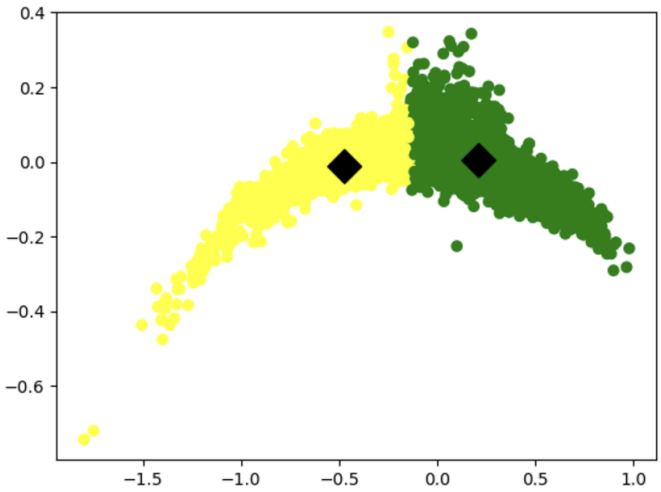

Fig. 13. Cluster k = 2, log real

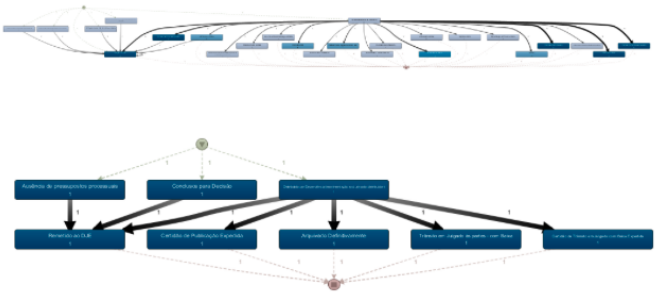

Fig. 14. Traces mais significativos extraídos de cada cluster, log real

De forma semelhante ao realizado no sintético, trocando o algoritmo do Doc2Vec para a variante Skip-gram, mantendo os mesmo valores de hiperparâmetros, obtemos uma clusterização diferente e com silhoutte score menor (0.32), conforme Fig. 15.

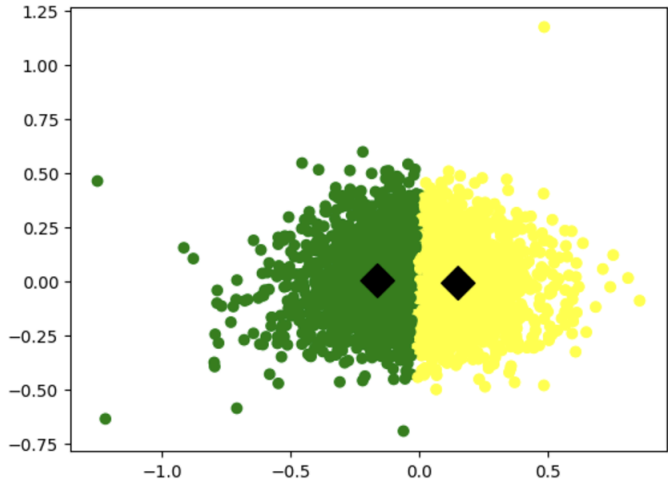

Fig. 15. Variante Skip-gram, log real

C. *Detecção de anomalias*

A anomalia introduzida no log sintético é o trace com o case 6686, com um fluxo incomum de processo, que traz uma atividade não mapeada em outros cases. Tentamos identificar esse trace pegando o ponto mais distante de qualquer centróide, mas essa técnica trouxe outro case (328) que é um fluxo longo mas não incomum. A natureza contextual do Doc2Vec pode produzir vetores próximos mesmo que o contexto seja diferente e acreditamos que outra abordagem possa ser utilizada e requer mais investigações.. Na Fig. 16 temos DFG do trace 6686 e 328.

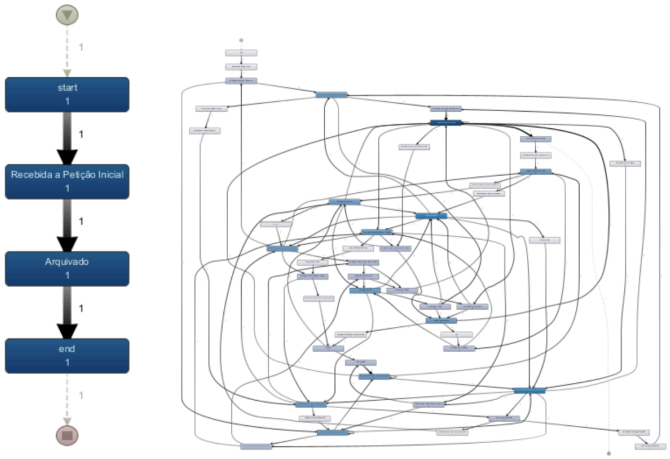

Fig. 16. Traces 6686 e 328, esse último o mais distante das centróides

VI. CONCLUSÃO

Neste trabalho realizamos uma avaliação empírica de como os hiperparâmetros de modelos de espaço vetorial podem afetar os resultados quando aplicado a técnica de trace clustering. Embora existam diferentes métodos de trace encoding conforme demonstrado em [10] e cada método pode ser melhor dependendo do cenário e objetivo do trabalho, a escolha do método por um pesquisador ou profissional da área de process mining não encerra as possibilidades de decisões para a tarefa de trace clustering. É necessário ainda realizar o ajuste dos hiperparâmetros para otimizar as métricas de qualidade ou para melhor atingir os objetivos pretendidos.

Neste trabalho escolhemos o Doc2Vec como modelo de espaço vetorial para trace clustering. Pelo fato desse algoritmo de aprendizagem ser não supervisionado dificulta a aplicação de técnicas de optimização de hiperparâmetros, como GridSearch[5].

Soluções têm sido utilizadas como a construção de pipelines que automatizam os treinamentos e fazem uma classificação baseada em uma métrica objetivo.

Alguns trabalhos na área de process mining estudam o problema de otimização de hiperparâmetros como em [11] que foca em monitoramento preditivo de modelo de negócios. Já para trace clustering em [12] e [13] os autores lidam diretamente com o problema enfrentado neste trabalho e propõem um Automatic Machine Learning (AutoML) framework que recomenda um pipeline mais adequado dado um log de evento.

5. https://scikit-learn.org/1.5/modules/grid_search.html

REFERENCES

[1] Koninck, P., Broucke, S., and Weerdt, J. act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. In Bus. Process Manage., volume 11080 of Lect. Notes Comput. Sci., pages 305–321, Berlin. Springer, 2018.

[2] Mateus A.S. Luna, André P. Lima, Thais R. Neubauer, Marcelo Fantinato, Sarajane M. Peres. Vector space models for trace clustering: a comparative study. In: XVIII Encontro Nacional de Inteligência Artificial e Computacional, 2021, São Paulo. Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional, 2021. p. 446-457.

[3] van der Aalst,W. M. P.. Process Mining: Data Science in Action. Springer, Berlin, 2nd edition, 2016.

[4] Song, M., Günther, C.W., van der Aalst, W.M.P. Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds) Business Process Management Workshops. BPM 2008. Lecture Notes in Business Information Processing, vol 17. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-00328-8_11.

[5] Lu, X. Using behavioral context in process mining: exploration, preprocessing and analysis of event data. PhD dissertation, Eindhoven University of Technology, 2018.

[6] Le, Q. and Mikolov, T. Distributed Representations of Sentences and Documents. International Conference on Machine Learning, Beijing, 21-26 June 2014, 1188-1196.

[7] Han, J., Pei, J., and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kauffman, Waltham, 3rd edition, 2012.

[8] Unger, A. J. ; Santos Neto, J. F. ; Trecenti, J. ; Hirota, R. ; Fantinato, M. ; Peres, S. M. . Process Mining-Enabled Jurimetrics: Analysis of a Brazilian Court's Judicial Performance in the Business Law Processing. In: 18th International Conference on Artificial Intelligence and Law, 2021, São Paulo (Online). Proceedings of the 18th International Conference on Artificial Intelligence and Law.

[9] Jey Han Lau and Timothy Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proceedings of the 1st Workshop on Representation Learning for NLP, pages 78–86, Berlin, Germany. Association for Computational Linguistics, 2016.

[10] G. M. Tavares, R. S. Oyamada, S. Barbon, and P. Ceravolo, “Trace encoding in process mining: A survey and benchmarking,” Eng. Appl. Artif. Intell., vol. 126, 2023, Art. no. 107028.

[11] Di Francescomarino, Chiara & Dumas, Marlon & Federici, Marco & Ghidini, Chiara & Maggi, Fabrizio & Rizzi, Williams & Simonetto, Luca. Genetic Algorithms for Hyperparameter Optimization in Predictive Business Process Monitoring. Information Systems. 10.1016/j.is.2018.01.003.

[12] Barbon Junior, Sylvio & Ceravolo, Paolo & Damiani, Ernesto & Tavares, Gabriel. Selecting Optimal Trace Clustering Pipelines with AutoML. 10.48550/arXiv.2109.00635, 2021.

[13] Grigore, I.M.; Tavares, G.M.; Silva, M.C.d.; Ceravolo, P.; Barbon Junior, S. Automated Trace Clustering Pipeline Synthesis in Process Mining. *Information* 2024, *15*, 241. https://doi.org/10.3390/info150