Estudo do cálculo paralelo das métricas trustworthiness e continuity para uso em tempo real

Anderson da Conceição da Silva, Tácito Trindade de Araújo Tiburtino Neves

Núcleo de Ciências Exatas - Universidade Federal de Alagoas (UFAL) / Campus Arapiraca – Arapiraca – AL – Brazil

anderson.conceicao@arapiraca.ufal.br, tacito.neves@arapiracafal.br

Abstract. *As a result of the abundant generation of data on the internet, the need to extract information has also grown and tools to aid this analysis are essential. Multidimensional Projection techniques are resources that help in data analysis. They allow the user to make a visual analysis of the data. There are several metrics to assess the quality of a generated projection. One of the problems in using these metrics is their high computational cost. In this work, the projection technique LAMP and the metrics Trustworthiness and Continuity were chosen. Promising results were obtained in both parallelized algorithms in a comparison with non-parallel algorithms.*

Resumo. *Em consequência da abundante geração de dados na internet, a necessidade de extrair informações também cresceu e ferramentas que auxiliassem essa análise são essenciais. As técnicas de Projeção Multidimensional são recursos que auxiliam na análise dos dados. As mesmas possibilitam ao usuário fazer uma análise visual dos dados. Existem varias métricas para avaliar a qualidade de uma projeção gerada. Um dos problemas de utilização dessas métricas é seu alto custo computacional. Neste trabalho foi escolhido a técnica de projeção LAMP e as métricas Trustworthiness e Continuity. Foram obtidos resultados promissores em ambos algoritmos paralelizados em uma comparação com os algoritmos não paralelos.*

1. INTRODUÇÃO

Com o passar dos anos a computação continua a evoluir, gerando assim abundantes quantidades de dados, como mostra o estudo feito pela domo [Dom 2021], onde apresenta que o consumo global de dados em 2021 até julho foram consumidos 79 Zettabyte e a projeção para 2025 é de 180 Zettabytes. Extrair alguma informação relevante desse elevado volume é algo bastante requisitado, mas uma tarefa difícil, principalmente se realizada sem o auxílio de ferramentas computacionais.

Abordagens como técnicas de Projeção Multidimensional são bastante promissoras nesse sentido. São conjuntos de técnicas que utilizam cálculos matemáticos para mapear dados de um espaço com N dimensões para dimensões menores, geralmente para um espaço bidimensional. A redução de dimensionalidade busca manter as principais características dos dados, tentando manter elementos semelhantes próximos ou/e agrupados, e elementos diferentes mais afastados.

Atualmente existem formas distintas de avaliar o resultado das técnicas de Projeção Multidimensional, sendo *Continuity* [Mateus Espadoto and Telea 2019]

e *Trustworthiness* [Mateus Espadoto and Telea 2019] bastante utilizadas, e por consequência disso foram escolhidas para este projeto. *Continuity* e *Trustworthiness* são métricas nas quais o resultado gerado por elas está entre $[0, 1]$, sendo 1 o melhor valor para um mapa. Entretanto, essas métricas tendem a ser extremamente custosas computacionalmente.

2. METODOLOGIA

A primeira etapa da metodologia foi realizar pesquisas bibliográficas para encontrar técnicas e métricas de projeção, métricas com características passiveis de serem paralelizadas.

Desse estudo foi escolhida a técnica de projeção aplicada como base nesse trabalho, a LAMP [Joia et al. 2011], por apresentar um ótimo custo computacional e boa acurácia.

Outras escolhas importantes foram as métricas de avaliação. *Trustworthiness* e *Continuity* por serem bastante utilizadas nos trabalhos encontrados na revisão da literatura realizada. As métricas estão representadas nas equações 1 e 2:

$$Mt = 1 - \frac{2}{NK(2n - 3K - 1)} \sum_{i=1}^{N} \sum_{j \epsilon U^i(K)} (r(i, j) - K), \tag{1}$$

Trustworthiness Mt com valores entre [0, 1], onde 1 o melhor resultado, mede a proporção de falsos vizinhos ao ponto P em uma projeção. Essa métrica é descrita na Equação 1 e o código é apresentado pelo Algoritmo 1. Onde $U^i(K)$ é o conjunto de pontos que estão entre os K vizinhos mais próximos do ponto i no espaço $2D$, mas não estão entre os K vizinhos mais próximos do ponto i em R^n, n é o número de dimensões, N é o número de amostras(pontos), $r(i, j)$ é a função que determina o $ranking$ dos vizinhos mais próximos de forma ordenada.

$$Mc = 1 - \frac{2}{NK(2n - 3K - 1)} \sum_{i=1}^{N} \sum_{j \epsilon V^i(K)} (r(i, j) - K), \tag{2}$$

Continuity Mc com valores entre [0, 1], onde 1 o melhor resultado, mede a proporção de falsos vizinhos ao ponto P no conjunto original. A equação dessa métrica é descrita na Equação 2 e o código é apresentado pelo Algoritmo 2. Onde $V^i(K)$ é o conjunto de pontos que estão entre os K vizinhos mais próximos do ponto i em R^n, mas não entre os K vizinhos mais próximos no 2D. Sendo capaz com o resultado de estimar o quão distinto estão os vizinhos dos dados no R^n comparados com a projeção, n é o número de dimensões, N é o número de amostras(pontos), $r(i, j)$ é a função que determina o $ranking$ dos vizinhos mais próximos de forma ordenada.

A maior dificuldade de utilizar essas métricas esta em seu alto custo computacional, principalmente para encontrar os vizinhos mais próximos ao ponto. Para resolver esse problema é utilizada a técnica de paralelização do código.

O processamento paralelo consiste em dividir em *threads* uma parte do processo. A ideia é fazer com que os somatórios das equações possam ser executados de forma simultânea, sendo cada um associado a uma *thread*.

A linguagem Python foi utilizada por fornecer diversas bibliotecas voltadas a computação científica. Entre as bibliotecas está Numpy [num 2020], sendo um pacote para trabalhar com vetores e matrizes multidimensionais e contem várias funções matemáticas já prontas. Outra tecnologia importante é a Numba [Num 2020], sendo um compilador `JIT` (*Just In Time*) e transforma o código Python em código de máquina. Toda a implementação foi realizada no *Google Colab*, que fornece recursos computacionais de maneira gratuita, possibilita o trabalho colaborativo e controle de versão.

Os pontos passiveis de serem paralelizados nos dois algorítimos são as somas suscetíveis de vizinhos dos pontos não contidos nos espaços originais e projetados. Observando os pseudos códigos podemos identificar esse ponto na linha 7. Onde todos os algoritimos estão disponiveis no link Colab.

Algoritmo 1 Trustworthiness

1: $N \leftarrow Quantidadedepontos$
2: $K \leftarrow Quantidadedevizinhos$
3: $n \leftarrow Numerodedimensoes$
4: $U \leftarrow Conjuntodevizinhosnoespaço2D$
5: $V \leftarrow Conjuntodevizinhosnoespaçon$
6: **para** i $\leftarrow$ 1 **até** N **faça**
7: **para** j $\leftarrow$ 0 **até** K **faça**
8: **se** $V[i][j] \notin U[i]$ **então**
9: $S1 \leftarrow S1 + Rank$
10: **fim se**
11: **fim para**
12: $S0 \leftarrow S0 + S1$
13: **fim para**
14: $tw \leftarrow 1 - \frac{2}{NK(2n-3K-1)} * S0$ ▷ Onde tw é o valor do Trustworthiness

Algoritmo 2 Continuity

1: $N \leftarrow Quantidadedepontos$
2: $K \leftarrow Quantidadedevizinhos$
3: $n \leftarrow Numerodedimensoes$
4: $U \leftarrow Conjuntodevizinhosnoespaçon$
5: $V \leftarrow Conjuntodevizinhosnoespaço2D$
6: **para** i $\leftarrow$ 1 **até** N **faça**
7: **para** j $\leftarrow$ 0 **até** K **faça**
8: **se** $U[i][j] \notin V[i]$ **então**
9: $S1 \leftarrow S1 + Rank$
10: **fim se**
11: **fim para**
12: $S0 \leftarrow S0 + S1$
13: **fim para**
14: $cn \leftarrow 1 - \frac{2}{NK(2n-3K-1)} * S0$ ▷ Onde cn é o valor do Continuity

Para validar a implementação, foram coletados diferentes conjuntos de dados com características distintas de tamanho e dimensionalidade. Esses conjuntos são descritos na Tabela 1.

Tabela 1. Conjuntos de dados utilizados nas comparações.

Nome	Instâncias	Dimensões	Fonte
Shuttle	43.500	9	[Dua and Graff 2017]
Mammals	50.000	72	[Dua and Graff 2017]
Corel	68.040	32	[Dua and Graff 2017]
Viscontest	100.000	10	[Whalen and Norman 2008]
Quantum	150. 000	78	[Caruana et al. 2004]
Fibers	250.000	30	[Paulovich et al. 2011]

A validação foi realizada através de comparativo direto de tempo entre o código original e o código paralelizado.

3. RESULTADOS

As técnicas de Projeção Multidimensional se utilizam de cálculos matemáticos para converter uma instancia de dado N-dimensional em uma representação bidimensional, realizando um mapeando visual [Nonato and Aupetit 2018]. O resultado deste mapeamento deve manter características dos dados originais, tais como: dados semelhantes devem manter-se próximos e dados distintos afastados. Mas nem sempre uma técnica representa visualmente os dados com boa qualidade.

A ferramenta Numba foi utilizada para paralelização. O Numba apresenta como característica principal para o usuário sua automatização no processo de paralelização. O paralelismo das métricas foi aplicado no somatório das equações 1 e 2, como citado na secção anterior. Para melhoria na eficiência na manipulação dos dados foi utilizado o Numpy, já que esses dados seriam manipulados diversas vezes tanto pela técnica de projeção multidimensional, quanto pelo cálculo dos algorítimos 1 e 2. Os gráficos apresentados nas figuras 1 e 2, demonstram a eficiência dos algorítimos paralelos em relação aos não paralelos.

Analisando Figura 1 que contem média de tempo de execução para 10 rodadas de testes, tanto para o algorítimo original da *Continuity* e o algorítimo paralelizado, podemos observar a distinta melhora nos resultados do algorítimo paralelo em relação ao não paralelo.

Em análise a Figura 2, onde foi seguida a mesma abordagem da técnica anterior, apresentando um contraste semelhante em seus resultados, onde em conjunto com a tabela 1 podemos inferir que o algorítimo paralelo da técnica *Trustworthiness* tem uma melhoria em sua eficiência quando testado com conjunto de dados maiores.

4. Conclusões e trabalhos futuros

Neste trabalho a proposta era de melhorar o tempo de execução dos algorítimos que quantificam a qualidade de mapas visuais gerados por técnicas de Projeção Multidimensional. Esse objetivo foi atingindo com sucesso, utilização tecnologias como Python, Numpy e o conceito de paralelismo de $threads$.

Mesmo com o desenvolvimento e execução dos experimentos prejudicados devido à pandemia causada pelo COVID-19, pela impossibilidade acesso a recursos disponíveis nos laboratórios. Parte desse problema foi contornada, utilizando o $Colab$.

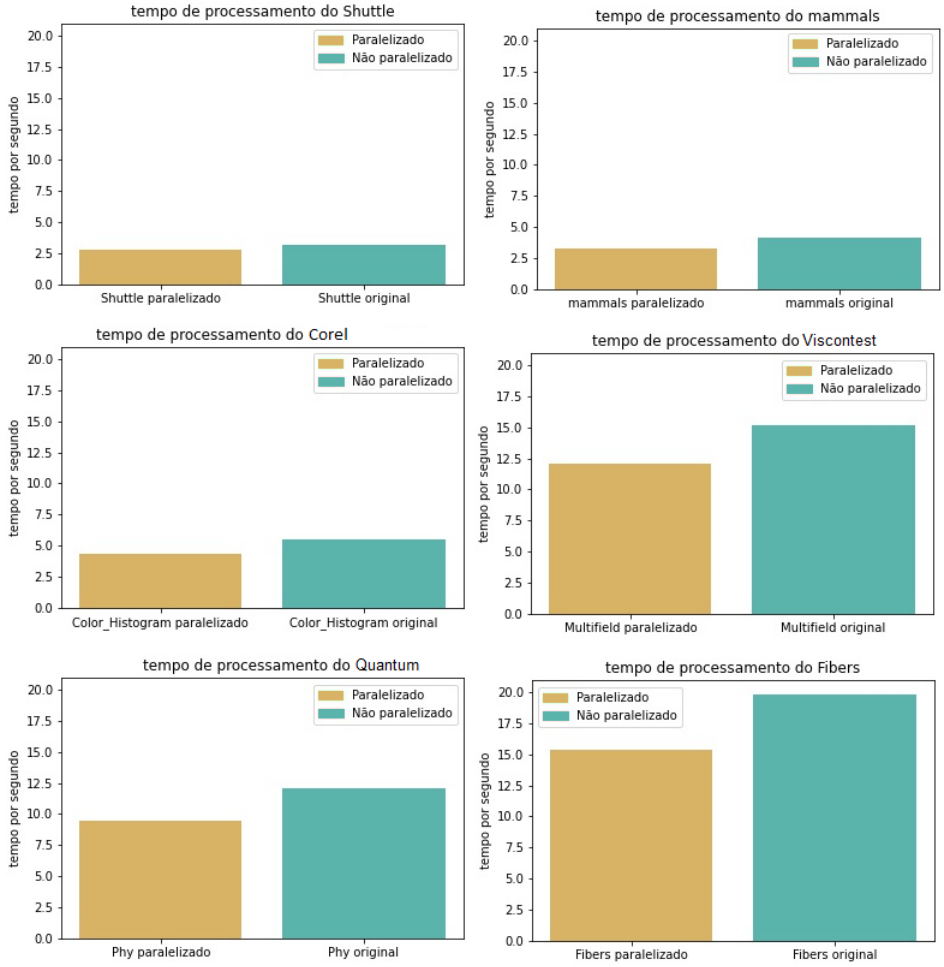

Figura 1. Médias de tempo do algoritmo Continuity.
Fonte: Autoria própria.

Para trabalhos futuros, uma sugestão é explorar a implementação de módulos para utilização desses algorítimos de forma simples para o usuário. Outra proposta seria otimização de outras métricas de avaliação, através da reutilização dos conceitos estudados neste ciclo.

Referências

(2020). Numpy reference — numpy v1.19 manual.

(2020). Writing cuda kernels — numba for cuda.

(2021). Domo - data never sleeps 9.0.

Caruana, R., Joachims, T., and Backstrom, L. (2004). KDD-Cup 2004: results and analysis. *ACM SIGKDD Explorations Newsletter*, 6(2):95–108.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Joia, P., Coimbra, D., Cuminato, J. A., Paulovich, F. V., and Nonato, L. G. (2011). Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17:2563–2571.

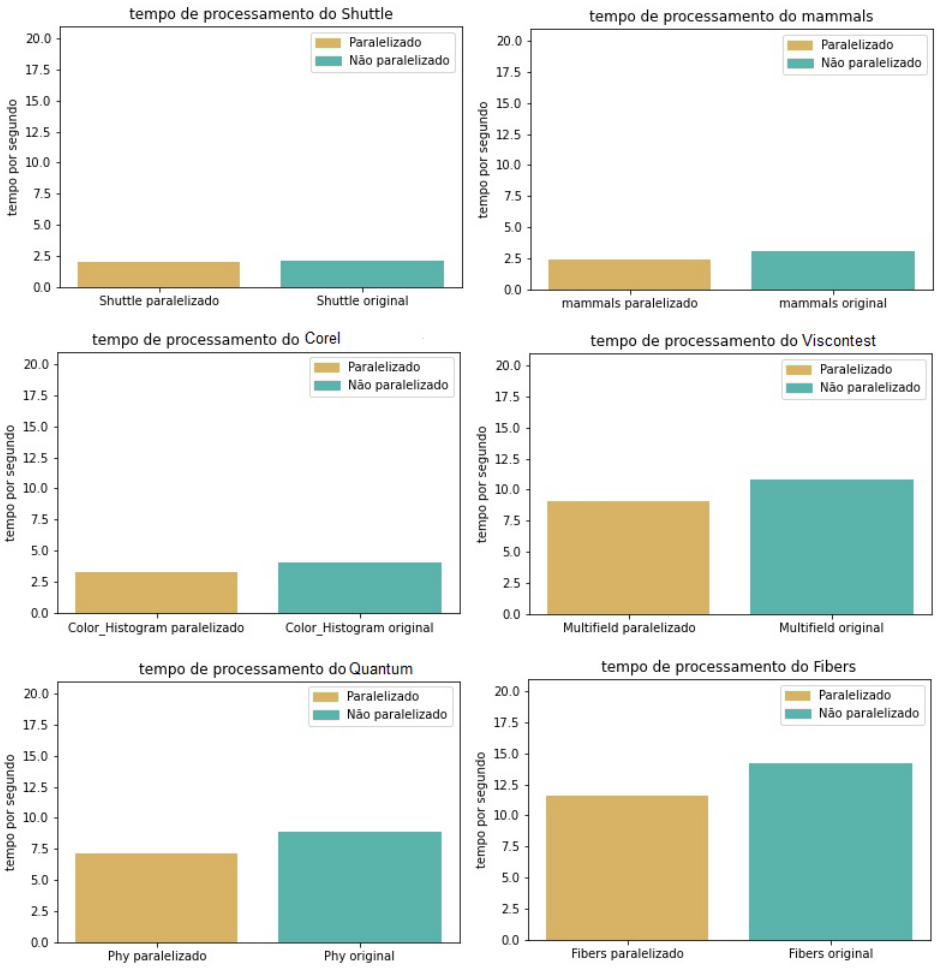

Figura 2. Média de tempo do algoritmo Trustworthiness.
Fonte: Autoria própria.

Mateus Espadoto, Rafael M. Martins, A. K. N. S. T. H. and Telea, A. C. (2019). Toward a quantitative survey of dimension reduction techniques. *IEEE transactions on visualization and computer graphics*, 27(3):2153–2173.

Nonato, L. G. and Aupetit, M. (2018). Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics*, 25(8):2650–2673.

Paulovich, F. V., Eler, D. M., Poco, J., Botha, C. P., Minghim, R., and Nonato, L. G. (2011). Piece wise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30(3):1091–1100.

Whalen, D. and Norman, M. L. (2008). Competition data set and description. 2008 IEEE Visualization Design Contest.