Métodos de projeção multidimensional e aplicações

Aline Martins Nascimento Belchior
Federal University of Espirito Santo
Vitoria, Espirito Santo 29075-910
Email: alinemartinsnb2@gmail.com

Alcebíades Dal Col
Federal University of Espirito Santo
Vitoria, Espirito Santo 29075-910
Email: alcebiades.col@ufes.br

***Abstract*—Neste artigo apresentaremos técnicas de projeções multidimensionais que podem ser utilizadas para reduzir dados de um espaço de dimensão alta para um espaço visual. Além disso, estudaremos algumas métricas com o objetivo de determinar a qualidade desses métodos de projeção. Diante disso, seremos capazes de entender as principais características dos métodos estudados. E, finalmente, aplicaremos essas técnicas em certos conjuntos de dados para análise dos mesmos.**

***Abstract*—In this article we will present multidimensional projection techniques that can be used to reduce the data from a high-dimesional space to a visual space. Furthermore, we will study a few metrics in order to establish the quality of said projection methods. Considering that, we will be able to comprehend the foremost characteristics of the studied methods. At last, we will apply these techniques on certain datasets for analysis.**

I. Introdução

Conforme o número de variáveis envolvidas em um problema matemático aumenta, mais complicado se torna trabalhar com ele. Por exemplo, lidar com pandemias e fenômenos meteorológicos demanda o trato de diversos fatores. Nesse contexto, surgem os Métodos de Projeção Multidimensional (MPM) com o objetivo de simplificar o tratamento desse tipo de objeto de estudo. Dados são objetos em um espaço m-dimensional, sendo m o número de variáveis deles. Assim, os MPM permitem representar dados de um espaço m-dimensional em uma dimensão menor de forma que a análise e estudo deles seja facilitada. Todo o processo de projeção é executado visando preservar no espaço de dimensão menor as relações de vizinhança dos dados no espaço m-dimensional.

No presente trabalho, será utilizado o conjunto de dados Íris [1], esse conjunto possui dados de 150 flores do tipo Íris, separadas de acordo com a sua espécie sendo as 50 primeiras da espécie Íris Versicolor, as 50 seguintes da Íris Virgínica e as 50 últimas da Íris Setosa. Além disso, a cada flor são atribuídas como variáveis as seguintes medidas em centímetros: 1) Comprimento da sépala; 2) Largura da sépala; 3) Comprimento da pétala; 4) Largura da pétala. Seja p_1 um vetor que representa a primeira flor do conjunto Íris, por exemplo, temos que $p_1 = (5,1 \quad 3,5 \quad 1,4 \quad 0,2) \in \mathbb{R}^4$.

Note que trabalhar com esse conjunto de dados no espaço de dimensão 4 torna complicado inferir traços de relações ou particularidades das espécies de flores. Assim, serão utilizados os seguintes métodos de projeção para que haja uma assimilação mais efetiva do conjunto: Nearest Neighbor Projection (NNP) [2], Force [2], Least Square Projection (LSP) [3] e Local Affine Multidimensional Projection (LAMP) [4].

Segue a definição formal do processo objetivo dos MPMs: Seja S em $\mathbb{R}^m$ um conjunto de dados tal que $S = \{p_1, ..., p_n\}$. A dissimilaridade entre dois dados quaisquer p_i e p_j do espaço m-dimensional é dada por $\delta(p_i, p_j)$. A dissimilaridade quantifica o quanto os elementos são diferentes e depende da aplicação em questão. A escolha mais comum para a dissimilaridade é a distância euclidiana. Seja S' um conjunto de pontos em dimensão baixa, uma projeção multidimensional é uma função bijetiva $f : S \to S'$ que tem como objetivo tornar $|\delta(p_i, p_j) - d(f(p_i), f(p_j))|$ o mais próximo de zero possível para quaisquer p_i e p_j pertencentes a S, sendo d referente a distância euclidiana entre as projeções dos pontos de S no espaço dimensional menor. O ponto $f(p_i) = p'_i$ é chamado de projeção de p_i.

II. Nearest Neighbor Projection

O método *Nearest Neighbor Projection* (NNP) [2] tem como objetivo redimensionar um conjunto de dados S, isto é, o que antes possuía m dimensões irá se converter para $\mathbb{R}^2$, de forma que as distâncias entre dois pontos vizinhos mais próximos em $\mathbb{R}^m$ se mantenham na projeção. Note que, para o conjunto Íris $m = 4$.

A projeção será realizada em diversas etapas bem definidas que se repetem ao longo do processo: primeiro, é necessário escolher dois pontos vizinhos em $\mathbb{R}^m$ e calcular a distância entre eles. Após isso, a distância encontrada será mantida no novo espaço dimensional, resultando em um novo conjunto geométrico. Para as próximas etapas, escolha um ponto vizinho e realize as mesmas operações com os pontos já projetados. Os cálculos das distâncias entre dois pontos vizinhos é feito utilizando conceitos da Geometria Analítica e da Álgebra Linear. Assim, chame de $S' \subset S$ o conjunto dos pontos que já possuem coordenadas em $\mathbb{R}^2$ e seja S o conjunto de todos os pontos no espaço m-dimensional original. Seja $p \in S$ o ponto que queremos projetar em $\mathbb{R}^2$, $\exists\ q,\ r \in S$ (que já possuem projeções q', $r' \in \mathbb{R}^2$) tais que suas distâncias até p são menores que as de todos os outros elementos de S'. A partir desses pontos, iremos utilizar duas circunferências C_q e C_r com centros em q' e r' e seus raios serão as distâncias no espaço m-dimensional $\delta(p, q)$ e $\delta(p, r)$, respectivamente. A partir dessas circunferências, encontraremos a projeção de p em $\mathbb{R}^2$, a qual chamaremos de p'.

Três casos diferentes poderão acontecer em relação a C_q e C_r: as duas circunferências se tangenciam (caso i), as duas circunferências possuem dois pontos de intersecção (caso ii) ou

então não há interseção entre as duas circunferências no espaço dimensional menor (caso iii), isto é, uma circunferência está contida na outra (caso $iii\ a$) ou a distância entre seus centros é maior que a soma dos raios (caso $iii\ b$).

i) Quando as duas circunferências são tangentes, isto é, quando elas se encontram em um único ponto, esse ponto é o que queremos encontrar, ou seja, calculando o ponto de tangência entre as circunferências C_q e C_r, encontramos p'.

ii) Quando as duas circunferências possuem dois pontos de intersecção o ponto $p' \in S'$ é escolhido aleatoriamente entre os dois pontos de intersecção. Essa escolha aleatória pode ser feita a partir da biblioteca $random$ e, mais especificamente, a função $np.random.randint$ em Python mas, a fim de facilitar a transcrição do problema, escolhemos sempre o primeiro ponto.

iii) Quando não há ponto de encontro entre as circunferências existem duas opções:

a) Se uma circunferência está contida na outra, traça-se um segmento de reta a partir do centro da circunferência maior e que passa pelo centro da circunferência menor e termina em um ponto contido na primeira circunferência, de forma que sua medida seja equivalente ao raio da maior circunferência. Desse segmento, iremos subtrair a distância entre os dois centros e a medida do raio da menor circunferência, restando, assim, a menor distância entre C_q e C_r. O ponto p' será o ponto médio da distância obtida;

b) Se a distância entre os centros é maior que a soma dos raios das circunferências, traça-se um segmento de reta ligando os dois centros e, dele, subtrai-se a medida dos dois raios, assim, restará a distância entre as duas circunferências. O ponto p' que queremos calcular é o ponto médio da distância obtida.

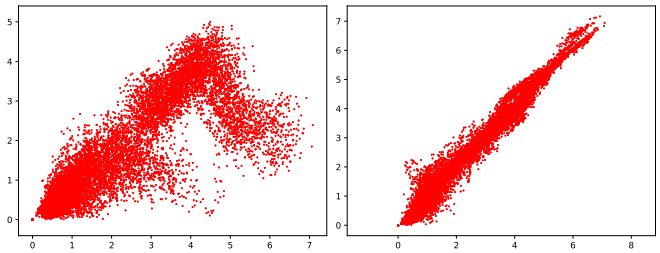
Como o conjunto Íris possui 150 pontos, a projeção será realizada em 150 etapas, resultando em um gráfico com a representação de todos os pontos $p' \in \mathbb{R}^2$, como mostra a Figura 3a. Note que foi possível visualizar os elementos das três espécies de Íris, e consequentemente, será possível analisar a distribuição dos seus elementos. As flores representadas com pontos de cor azul estão mais separadas, isso ocorre pois suas características são bem distintas das outras; quanto mais distintas forem as características de um objeto, mais facilmente será feita a separação.

III. Force

Para além do NNP, o método Force [2] surge na forma de implementações que podem ser feitas a partir da projeção criada pelo NNP com o objetivo de melhorar a projeção dos dados. A ideia central desse método é afastar pontos que foram projetados muito próximos e aproximar pontos que foram projetados muito distantes.

O esquema de melhoria Force funciona da seguinte forma: Em primeiro lugar, ele recebe como entrada os dados em dimensão m e sua projeção em dimensão menor. Em seguida, será realizada a comparação de cada ponto projetado em dimensão menor com todos os demais pontos projetados (em relação aos seus dados de origem no espaço m-dimensional). Sendo $S = \{p_1, \ldots, p_n\}$ o conjunto de dados m-dimensional, $S' = \{p'_1, \ldots, p'_n\}$ o conjunto de dados projetado e p'_i o ponto em dimensão menor que está sendo analisado, por exemplo, será feita a comparação de $p_i \in S$ com $p_j \in S$ onde j percorre o intervalo de 1 até 150 discretamente. A depender do resultado obtido esse pontos se tornarão mais próximos ou mais distantes. Para isso, é definido o vetor $v_{ij} = p'_j - p'_i$ e $\Delta_{ij} = \delta(p_i, p_j) - d(p'_i - p'_j)$, onde δ é uma função de dissimilaridade e d a distância euclidiana. Assim, é aplicado no ponto $p'_j \in P$ na direção do vetor v_{ij} uma perturbação que depende de Δ_{ij}. Portanto, se $\Delta_{ij} > 0$, o ponto p'_j é afastado de p'_i na projeção; se $\Delta_{ij} < 0$, o ponto p'_j se aproxima de p'_i; e, por fim, se $\Delta_{ij} = 0$ a posição de p'_j permanece a mesma. O artigo [2] sugere que seja utilizado uma fração do Δ_{ij}, entretanto nós utilizamos ele inteiro nos experimentos.

A Figura 1a mostra o gráfico de dispersão do conjunto antes da aplicação do Force, de forma que a abscissa dos pontos é $\delta(p_i, p_j)$ e a ordenada é $d(p'_i, p'_j)$. Note que, em uma projeção ideal, o gráfico de dispersão estaria o mais próximo possível da função identidade. Já a Figura 1b representa o gráfico de dispersão do conjunto de dados após a aplicação do método Force e a Figura 3b exibe a projeção dos dados após a melhoria ser aplicada.



(a) NNP (b) Force

Fig. 1. Gráficos de dispersão para o conjunto Íris ($\delta(p_i, p_j) \times d(p'_i, p'_j)$).

IV. Least Square Projection

O Least Square Projections (LSP) [3] é um método que se diferencia dos anteriores pois ele utiliza pontos de controle para orientar a projeção. Assim, sendo S um conjunto de dados a ser projetado em dimensão menor, no geral $\mathbb{R}^2$, é preciso: 1) Determinar e projetar em $\mathbb{R}^2$ os pontos de controle de S; 2) Determinar a vizinhança dos pontos $p_i \in S$; 3) A partir dos passos anteriores, construir sistemas lineares, sendo as soluções deles as coordenadas das projeções p'_i em $\mathbb{R}^2$. Para a primeira etapa, deve-se definir a quantidade n_c de pontos de controle para que o algoritmo *k-means* [5] separe S em n_c aglomerados. Cada um desses aglomerados possui um centroide e eles são encontrados por meio da biblioteca *sklearn* do Python. Dessa forma, o $p_i \in S$ mais próximo do centroide de seu aglomerado é escolhido como ponto de

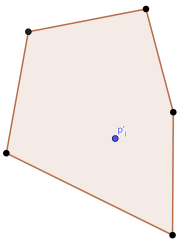


Fig. 2. Representação de uma projeção no fecho convexo de seus $n_{viz} = 5$

controle. Tem-se assim o conjunto S_c dos pontos de controle, que é então projetado em $\mathbb{R}^2$ usando o NNP, em nosso caso. Na segunda etapa, é preciso definir a quantidade n_{viz} de vizinhos para os $p_i \in S$. Para cada p_i é definida uma lista $V_i = \{p_{i1}, \ldots, p_{in_{viz}}\}$ dos n_{viz} pontos mais próximos dele em $\mathbb{R}^m$. A ideia por trás da definição dos vizinhos é a de que ao realizarmos a projeção, o ponto projetado estará localizado no fecho convexo de seus vizinhos. Matematicamente, isso significa:

$$p'_i - \sum_{j=1}^{n_{viz}} \alpha_{ij} p'_{ij} = 0;\ 0 \leq \alpha_{ij} \leq 1;\ \sum \alpha_{ij} = 1. \quad (1)$$

Observe o exemplo na Figura 2 que elucida melhor a ideia supracitada.

Por fim, na última etapa são montados os sistemas lineares $LX_1 = 0$ e $LX_2 = 0$, tais que $X_1 = (x_{11}\ x_{12}\ \ldots\ x_{1n})^T$, $X_2 = (x_{21}\ x_{22}\ \ldots\ x_{2n})^T$ para $p'_i = (x_{1i}, x_{2i})$, e L é a matriz $n \times n$ tal que

$$l_{ij} = \begin{cases} 1 & \text{, se } i = j \\ -\alpha_{ij} & \text{, se } p_j \in V_i \\ 0 & \text{, caso contrário} \end{cases}, \quad (2)$$

sendo que $\alpha_{ij} = 1/n_{viz}$.

Após isso, alteramos os sistemas usando uma matriz A de forma que $A = \begin{pmatrix} L \\ C \end{pmatrix}$ com $C_{n_c \times n}$ dado por:

$$c_{ij} = \begin{cases} 1, & \text{se } p_j \text{ é o ponto de controle associado a linha } i \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

e $b_i = (b_{i1}, b_{i2}, ..., b_{i(n+n_c)})^T$, sendo que $b_{i1}, b_{i2}, ..., b_{in} = 0$ e para $n < j \leq n + n_c$, b_{ij} corresponde a i-ésima coordenada do ponto de controle referente a linha j de A. Consideremos, por exemplo, p_3 e p_6 pontos de controle de um conjunto $S = \{p_1, p_2, ..., p_6\}$. Onde a sétima e a oitava linha de A são dadas por $(0, 0, 1, 0, 0, 0)$ e $(0, 0, 0, 0, 0, 1)$, respectivamente. Suponhamos que as projeções de p_3 e p_6 sejam dadas por $p'_3 = (0.5, 0.6)$ e $p'_6 = (2, 1.1)$. Nesse caso os vetores b_1 e b_2 são dados por: $b_1 = (0, 0, 0, 0, 0, 0, 0.5, 2)^T$ e $b_2 = (0, 0, 0, 0, 0, 0, 0.6, 1.1)^T$.

Por fim, busca-se solucionar os sistemas dados por $AX_1 = b_1$ e $AX_2 = b_2$ usando $x_1 = (A^TA)^{-1}A^Tb_1$ e $x_2 = (A^TA)^{-1}A^Tb_2$. Os sistemas lineares são resolvidos no sentido de mínimos quadrados, o que significa que nós queremos minimizar $||AX_1 - b_1||$ e $||AX_2 - b_2||$. Observe que a matriz A não é quadrada. A Figura 3c mostra a projeção do conjunto de dados Íris usando o LSP.

V. Local Affine Multidimensional Projection

O método Local Affine Multidimensional Projection (LAMP) [4] é uma técnica de projeção que busca uma família de transformações afins $f : \mathbb{R}^m \to \mathbb{R}^2$ dada por $f(x) = xM + t$ para reduzir a dimensão dos dados.

Mais especificamente, seja $S = \{p_1, \ldots, p_n\} \subset \mathbb{R}^m$ o conjunto de dados cujos elementos queremos reduzir a dimensão para $\mathbb{R}^2$. Suponha que foram selecionados pontos de controle $S_c = \{a_1, \ldots, a_{n_c}\}$ a partir do conjunto S. Por exemplo, selecionando n_c pontos de controle através do método k-means [5] (no Python usamos o comando *from sklearn import cluster.KMeans*).

Selecionados os pontos de controle é necessário projetá-los, para isso utilizamos um método de projeção já existente. Optamos por utilizar o NNP, visto que já o havíamos estudado. Desse modo temos um conjunto $S'_c = \{a'_1, \ldots, a'_{n_c}\} \subset \mathbb{R}^2$ dos pontos de controle projetados que auxiliarão na projeção dos demais.

Seja $p \in S$, nós queremos construir uma transformação afim $f_p(x) = xM + t$ para p que torna mínimo o somatório $\sum_{i=1}^{n_c} \alpha_i ||f_p(a_i) - a'_i||^2$, onde $M_{m\times 2}$ é uma matriz ortogonal e $t_{1\times 2}$ é um vetor ambos a serem determinados. O coeficiente α_i é definido por $\alpha_i = \frac{1}{||a_i - p||^2}, i = 1, \ldots, n_c$. Já a matriz M deve ser ortogonal pois assim os dados serão apenas rotacionados ou transladados durante o processo de projeção, o que é importante visto que queremos preservar distâncias.

Para solucionar tal problema de minimização, inicialmente, substituímos a expressão de f_p no somatório obtendo assim $\sum_{i=1}^{n_c} \alpha_i ||a_i M + t - a'_i||^2$. Pode-se mostrar que o vetor t é dado por $t = \tilde{b} - \tilde{a}M$, onde $\tilde{a} = \frac{\sum_{i=1}^{n_c} \alpha_i a_i}{\alpha}$ e $\tilde{b} = \frac{\sum_{i=1}^{n_c} \alpha_i a'_i}{\alpha}$, sendo α o somatório dos α_i's. Observe que precisamos obter a matriz M afim de construir a função f_p. Joia et al [4] mostram em seu artigo que $M = UV$, onde U e V são dados pela decomposição em valores singulares da matriz A^TB, mais especificamente $A^TB = UDV$. Sendo que:

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{a}_1 \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{a}_{n_c} \end{bmatrix}, B = \begin{bmatrix} \sqrt{\alpha_1}\hat{b}_1 \\ \vdots \\ \sqrt{\alpha_{n_c}}\hat{b}_{n_c} \end{bmatrix}, \quad (4)$$

com $\hat{a}_i = a_i - \tilde{a}$ e $\hat{b}_i = a'_i - \tilde{b}$. Diante disso, a projeção p' de um dado ponto p pode ser obtida por:

$$p' = f_p(p) = pM + t = pM + \tilde{b} - \tilde{a}M = (p - \tilde{a})M + \tilde{b}. \quad (5)$$

Note que para cada $p \in S$ devemos achar uma transformação afim dessa forma para que consigamos realizar a projeção. A Figura 3d mostra a projeção do conjunto de dados Íris usando o LAMP.

VI. Resultados e Comparações

Nesta seção, iremos apresentar os resultados obtidos ao realizar a projeção multidimensional do conjuntos de dados

TABLE I
MÉTRICAS DAS PROJEÇÕES NO CONJUNTO ÍRIS

	NNP	Force	LSP	LAMP
Tempo	0.145	0.703	0.221	0.358
Stress	0.123	0.013	0.037	0.677
Silhueta	0.393	0.541	0.507	0.541

Íris. Os métodos de projeção utilizados são os apresentados no artigo. Mais especificamente, *NNP*, *Force*, *LSP* e *LAMP*. As implementações dos métodos foram feitas na linguagem de programação Python e os experimentos foram conduzidos por meio do Google Colab que tipicamente disponibiliza uma máquina virtual com 12GB de memória RAM e 107GB de memória em disco.

As métricas utilizadas para atestar a qualidade das projeções foram *Stress* [6], *Silhueta* [7] e o *Tempo* de processamento, em segundos. O Stress quantifica o quanto uma projeção distorce os dados em termos de distância, sua fórmula é dada por:

$$Stress = \frac{\sum_{i,j=1}^{n}(d_{ij} - d'_{ij})^2}{\sum_{i,j=1}^{n}(d_{ij})^2}, \tag{6}$$

onde n é igual ao número de elementos do conjunto, d_{ij} é a distância entre os pontos p_i e p_j no espaço de dimensão alta e d'_{ij} corresponde à distância entre as projeções p'_i e p'_j. Perceba que, quanto menor o Stress melhor a projeção. Por outro lado, a Silhueta determina numericamente o quanto os grupos se mantêm após a realização de uma projeção. Sua fórmula é dada por:

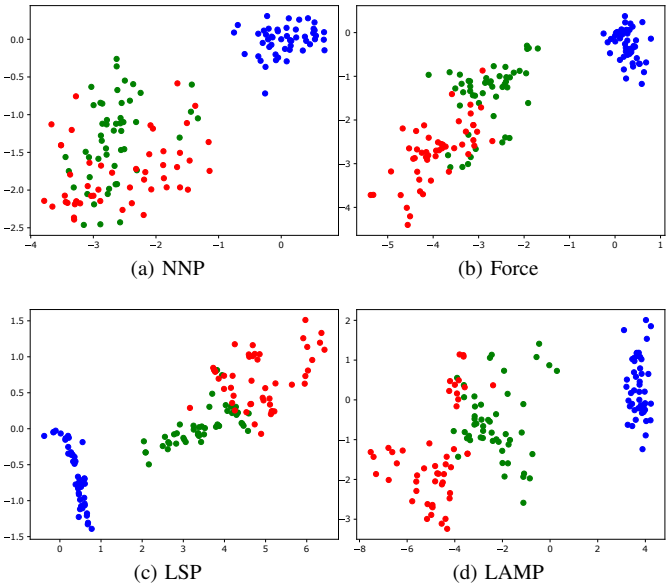
$$Silhueta = \frac{1}{n}\sum_{i=1}^{n}\frac{b_i - a_i}{max(a_i, b_i)}, \tag{7}$$

onde a_i é a média das distâncias de p_i a todos os pontos do mesmo grupo de p_i e b_i é a menor distância de p_i a todos os pontos de outros grupos. Note que, valores de Silhueta próximos de 1 indicam projeções que preservam bem seus grupos.

A Tabela I mostra os resultados obtidos ao aplicar as técnicas de projeção multidimensional no conjunto de dados Íris. Os resultados apresentados na tabela foram obtidos por meio da média de 20 iterações das projeções. A título de exemplo, uma projeção de cada método é exibida na Figura 3.

VII. CONCLUSÃO

Diante do exposto, é possível perceber a importância dos métodos de projeção multidimensional no estudo de conjuntos de dados multivariáveis, visto que eles permitem a projeção desses conjuntos em uma dimensão menor. Apesar do lento processamento, o Force, quando aplicado em conjuntos de dados, proporciona uma boa projeção de acordo com as métricas utilizadas. Tanto o LSP quanto o LAMP se utilizam de um fator não observado nos métodos anteriores: os pontos de controle, que orientam a projeção. Apesar disso, o cálculo das métricas revela que a projeção obtida pelo LAMP apresenta certa distorção em relação às distâncias entre os dados quando comparado com os demais métodos de projeção. Destarte, o



(a) NNP (b) Force (c) LSP (d) LAMP

Fig. 3. Projeções do conjunto de dados Íris.

método mais adequado à escolher irá depender do conjunto de dados que é trabalhado e da finalidade para que eles são manuseados.

Ademais, gostaríamos de frisar a vontade de seguir com a pesquisa. Em especial, pretendemos nos empenhar no estudo de novos métodos, o primeiro deles sendo o Principal Component Analysis (PCA), e também de outras métricas para qualificar as projeções.

AGRADECIMENTOS

Os autores desse artigo gostariam de agradecer ao Programa de Educação Tutorial - FNDE.

REFERÊNCIAS

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "**Scikit-learn: Machine Learning in Python**," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] E. Tejada, R. Minghim, and L. G. Nonato, "**On improved projection techniques to support visual exploration of multi-dimensional data sets**," *Information Visualization*, vol. 2, no. 4, pp. 218–231, 2003.

[3] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz, "**Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping**," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, 2008.

[4] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, "Local affine multidimensional projection," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2563–2571, 2011.

[5] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[6] I. Borg and P. J. Groenen, *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[7] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.