VI ERMAC
ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL
unesp

Projeções multidimensionais, por onde começar?

Adriano Barbosa
UFGD; FACET
adrianobarbosa@ufgd.edu.br

Resumo: Este artigo pretende apontar os primeiros passos para o leitor que deseja estudar ou utilizar técnicas de projeção multidimensional para análise de dados de alta dimensão. Apresentamos modelos matemáticos para dados de alta dimensão, formulação matemática de três técnicas de projeção multidimensional modernas, suas vantagens, desvantagens, bem como experimentos com conjuntos de dados reais e métricas para quantificação as projeções.

Palavras-chave: Computação Gráfica. Projeções Multidimensionais. Visualização.

Introdução

Atualmente a quantidade de dados coletadas por serviços de internet, aplicativos e redes sociais são gigantescas e o desenvolvimento de ferramentas adequadas para analisar esses dados se tornou extremamente importante. Uma classe de ferramentas existentes para auxiliar nessa tarefa é a das projeções multidimensionais, projeções capazes de mapear os dados de um espaço de alta dimensão num espaço visual, usualmente de dimensão dois ou três.

A formulação matemática robusta, versatilidade, capacidade de interação com o usuário e o apelo visual são as características mais atrativas das projeções multidimensionais (BARBOSA et al., 2016).

Nas próximas seções apresentaremos como caracterizar os dados, três técnicas de projeção multidimensional, suas limitações e experimentos. As técnicas foram escolhidas por conta de sua formulação matemática, precisão e eficiência computacional ao lidar com conjuntos de dados grandes.

Caracterizando os dados

Imagine que se deseja analisar o resultado de um exame de câncer. Uma forma de descrevê-lo é por meio de características do tumor como seu raio (média das distâncias do centro a pontos da borda), textura (variação numa escala de 0 a 1), perímetro e área, ou seja, uma coleção de (usualmente) números reais. Dessa forma, podemos modelar matematicamente cada tumor como um vetor num espaço $\mathbb{R}^d$, onde d (dimensão do espaço) é número de características extraídas do exame. Quanto mais características são extraídas (usualmente dezenas ou centenas) maior será a dimensão d desse espaço, justificando assim a nomenclatura dada a esse tipo de dado (*dados de alta dimensão*). O espaço onde os dados estão caracterizados é usualmente chamado espaço original dos dados ou espaço de características.

Outra forma de representar um conjunto de dados com n instâncias é utilizar uma matriz $M = [m_{ij}]_{n\times n}$ de similaridades ou distâncias, onde m_{ij} (entrada de M localizada na linha i e coluna j) é um número real que indica a similaridade ou distância entre a i-ésima e j-ésima instâncias do conjunto de dados. Matrizes de similaridade podem ser úteis, por exemplo, para representar as relações de amizade numa rede social, onde cada indivíduo (nó de um grafo) é uma instância de dado e as ligações (arestas do grafo) determinam quão próximos são os indivíduos, ou seja, a similaridade entre os indivíduos.

Técnicas de projeção multidimensional

Uma vez que os conjuntos de dados são representados por vetores num espaço de alta dimensão fica difícil analisar e entender as características e padrões desse conjunto. Para nos ajudar a visualizar os padrões contidos nos dados podemos utilizar ferramentas como as projeções multidimensionais.

Essas técnicas têm como objetivo projetar as instâncias de dado do espaço original num espaço visual, usualmente o plano ($\mathbb{R}^2$). A forma como a projeção é feita depende do que se deseja priorizar na visualização, existem, por exemplo, técnicas que visam preservar a relação de distância entre as instâncias no espaço original (TEJADA; MINGHIM; NONATO, 2003) e outras cujo objetivo é manter a relação de vizinhança existente originalmente nos dados (PAULOVICH et al., 2008). Veremos agora alguns exemplos dessas técnicas, suas formulações matemáticas, bem como suas vantagens e limitações.

Force Scheme

A primeira técnica que descreveremos é conhecida como *Force Scheme* (TEJADA; MINGHIM; NONATO, 2003) e visa preservar as distâncias entre as instâncias no espaço original. Inicialmente, as instâncias de dado são posicionadas (projetadas) no espaço visual de maneira aleatória e o algoritmo percorrerá todas as instâncias de modo a – segundo as distâncias no espaço original – separar pontos posicionados muito próximos e juntar pontos posicionados muito distantes um do outro. Para cada ponto x' no espaço visual (projeção da instância x do espaço original), o algoritmo percorre todas as demais projeções q' ($q' \neq x'$), calcula o vetor $v = \overrightarrow{x'q'}$ e move q' na direção v por uma porção Δ de v, onde

$$\Delta = \frac{d(x,q) - d_{min}}{d_{max} - d_{min}} - d(x',q')$$

é uma aproximação da diferença entre as distâncias de x a q no espaço original e no espaço visual, d_{min} e d_{max} são o mínimo e o máximo das distâncias entre as instâncias no espaço original, respectivamente.

Como a formulação da *Force Scheme* usa apenas as distâncias entre as instâncias de dado, ou seja, ela é capaz de lidar com dados descritos através de suas coordenadas em $\mathbb{R}^d$ ou de uma matriz de distâncias.

A *Force Scheme* é uma técnica bastante precisa e consegue refletir muito bem as relações de distância do espaço original no espaço visual. Entretanto, como o algoritmo tem que percorrer, para cada ponto, todas as demais instâncias de dado, seu uso se torna proibitivo quando o conjunto de dados é muito grande devido ao alto custo computacional envolvido. Para conjuntos de dados pequenos é um método de projeção bastante poderoso e faremos uso dele para auxiliar outras técnicas de projeção.

Local Affine Multidimensional Projection

Para contornar os problemas da *Force Scheme* com relação ao número de instâncias, a *Local Affine Multidimensional Projection* (LAMP) (JOIA et al., 2011) utiliza uma pequena amostra do conjunto de dados (chamados *pontos de controle*), suas projeções no espaço visual e calcula, para cada instância x, uma transformação afim. Uma vez encontrada a transformação afim mais adequada, a instância x é projetada utilizando essa transformação.

Para realizar a projeção dos pontos de controle podemos utilizar qualquer outra técnica de projeção, inclusive a *Force Scheme*, pois a quantidade de pontos de controle é usualmente pequena ($\sqrt{n}$, onde n é o número de instâncias do conjunto de dados).

Dados os pontos de controle $\{x_1, x_2, \ldots, x_k\} \subset \mathbb{R}^d$ e suas projeções $\{x'_1, x'_2, \ldots, x'_k\} \subset \mathbb{R}^2$, a LAMP calcula, para cada x a ser projetado, uma transformação afim $f_x : \mathbb{R}^d \to \mathbb{R}^2, f_x(p) = pM + t$ que minimiza:

$$\sum_i \alpha_i \| f_x(x_i) - x'_i \|^2, \text{ sujeito a } M^T M = I, \tag{1}$$

onde a matriz M e o vetor t são as incógnitas, I é a matriz identidade e α_i são dados por $\alpha_i = 1/\|x - x_i\|^2$.

A restrição $M^T M = I$ garante que a transformação afim seja uma transformação rígida, ou seja, efetue apenas rotações e translações nos dados. Dessa forma, f_x deve preservar as relações de distância entre as instâncias no espaço original (LIMA, 1995). Além disso, os escalares α_i definem pesos para as parcelas

do problema (1) de modo que os pontos de controle mais próximos de x tenham maior importância na escolha de f_x, enquanto que a influência dos pontos de controle mais distantes de x acaba sendo pequena.

Tomando as derivadas parciais de (1) com respeito a t iguais a zero, escreve-se t em função de M

$$t = \tilde{x}' - \tilde{x}M, \text{ onde } \tilde{x} = \frac{\sum_i \alpha_i x_i}{\sum_i \alpha_i}, \tilde{x}' = \frac{\sum_i \alpha_i x_i'}{\sum_i \alpha_i}.$$

Assim, (1) pode ser reescrito como

$$\sum_i \alpha_i \|\hat{x}_i M - \hat{x}_i'\|^2, \text{ sujeito a } M^T M = I,$$

onde $\hat{x}_i = x_i - \tilde{x}_i$ e $\hat{x}_i' = x_i' - \tilde{x}_i'$. E esse novo problema pode ser escrito na forma matricial como

$$\|AM - B\|_F^2, \text{ sujeito a } M^T M = I, \tag{2}$$

em que $\|\cdot\|_F$ é a norma de Frobenius e

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{x}_1 \\ \sqrt{\alpha_2}\hat{x}_2 \\ \vdots \\ \sqrt{\alpha_k}\hat{x}_k \end{bmatrix}, \; B = \begin{bmatrix} \sqrt{\alpha_1}\hat{x}_1' \\ \sqrt{\alpha_2}\hat{x}_2' \\ \vdots \\ \sqrt{\alpha_k}\hat{x}_k' \end{bmatrix}.$$

O problema (2) é um dos problemas de Procrustes (GOWER; DIJKSTERHUIS, 2004) e sua solução é dada por $M = UV$, sendo UDV a decomposição em valores singulares (SVD) do produto $A^T B$. Portanto, para cada x, a LAMP efetua sua projeção calculando x' como

$$x' = f_x(x) = xM + t = xM + \tilde{x}' - \tilde{x}M = (x - \tilde{x})M + \tilde{x}'.$$

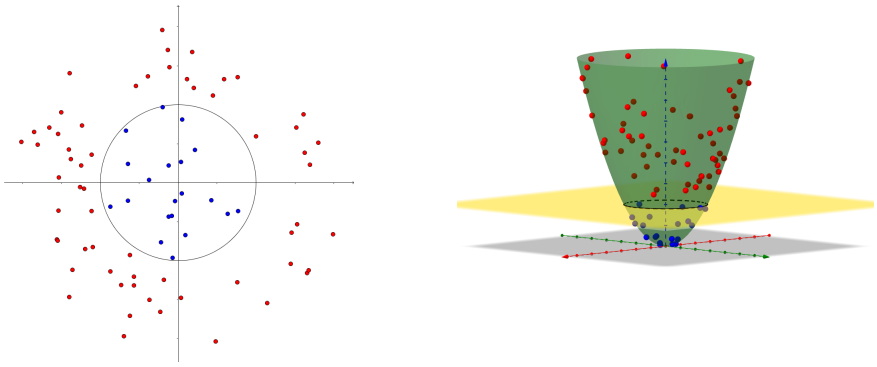
Diferente da *Force Scheme*, que usa apenas a informação de distâncias para projetar todas as instâncias do conjunto de dados, a LAMP calcula uma transformação afim diferente para cada ponto a ser projetado. Dessa forma, a LAMP consegue tirar proveito de características individuais de cada instância, como a influência da distância de cada x aos pontos de controle, por exemplo.

Kernel-based Linear Projection

A última técnica que apresentaremos é especializada em dados descritos por uma matriz de similaridade definida através de um *kernel*. Dado um conjunto de dados $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$, uma função $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ que associa a cada par de instâncias $x_i, x_j \in X$ um número real $k(x_i, x_j)$ é dita ser um *kernel* se a matriz $K = [k(x_i, x_j)]$ (*matriz do kernel*) é positiva definida. Um *kernel* define uma transformação implícita $\phi : \mathbb{R}^d \to \mathscr{H}$ tal que $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ é um produto interno no espaço de Hilbert $\mathscr{H}$ (SCHÖLKOPF et al., 2002).

A ideia por trás da *Kernel-based Linear Projection* (Kelp) (BARBOSA et al., 2016) é utilizar um *kernel* e sua transformação implícita para mapear os dados num espaço de Hilbert, de modo a conseguir desembaralhar os dados e perceber padrões intrínsecos dos dados de forma mais fácil e, daí, projetar esses padrões no espaço visual de modo claro. Imagine, por exemplo, que existam dois grupos distintos nos dados, mas que a separação entre eles é não-linear no espaço original. A Kelp, utilizando o *kernel* adequado, é capaz de mapear esses dados num espaço de Hilbert de modo que essa separação passe a ser linear (Figura 1). A partir daí a projeção conseguirá exibir essa separação de modo mais claro.

Dados um conjunto de dados X, a matriz K do *kernel*, pontos de controle $\{x_1, x_2, \ldots, x_n\} \subset X$, suas projeções $\{x_1', x_2', \ldots, x_n'\} \subset \mathbb{R}^2$ e denotando por K_s a matriz do *kernel* restrita aos pontos de controle, a Kelp busca uma transformação linear $M : \mathscr{H} \to \mathbb{R}^2$ tal que $M\phi(x_i) = x_i'$, $\forall i = 1, \ldots, n$, onde $\phi : \mathbb{R}^d \to \mathscr{H}$



(a) Dados no espaço original. (b) Dados no espaço de Hilbert.

Figura 1: Exemplo de um conjunto de dados com separação não-linear espaço original e seu mapeamento com separação linear num espaço de Hilbert.

é a transformação implícita definida pelo *kernel*. Ponto Φ como a matriz cujas colunas são $\phi(x_i)$, $i = 1,\ldots,n$, e X_s' a matriz com colunas x_i', $i = 1,\ldots,n$, podemos escrever a equação acima de forma matricial:

$$M\Phi = X_s' \Rightarrow M\Phi\Phi^T = X_s'\Phi^T \Rightarrow nMC = X_s'\Phi^T,$$

em que $C = \frac{1}{n}\Phi\Phi^T = \frac{1}{n}\sum_i \phi(x_i)\phi(x_i)^T$ é a matriz de covariância das imagens dos pontos de controle por ϕ. Como C é simétrica, $C = UDU^T$ (diagonalizável), onde as colunas de U são autovetores ortonormais de C e D é diagonal contendo os autovalores de C (LIMA, 1995). Multiplicando pela pseudo inversa $C^+ = U\tilde{D}^{-1}U^T$ ($\tilde{D}^{-1}$ é a inversa dos valores não nulos de D), tem-se

$$nMC = X_s'\Phi^T \Rightarrow M = \frac{1}{n}X_s'\Phi^T C^+ = \frac{1}{n}X_s'\Phi^T\left(U\tilde{D}^{-1}U^T\right).$$

Se A é a matriz cujas colunas são formadas pelos autovetores de K_s, é possível mostrar que $U = \Phi A \Rightarrow \Phi^T U = \Phi^T\Phi A \Rightarrow \Phi^T U = K_s A$, que $U^T\phi(x) = (\Phi A)^T\phi(x) = A^T\Phi^T\phi(x) = A^T k_x$, onde $k_x = [k(x,x_1)\ k(x,x_2)\ \cdots\ k(x,x_n)]^T$ e que $\gamma_i = n\lambda_i$, em que γ_i e λ_i são os autovalores de K_s e C, respectivamente (BARBOSA et al., 2016).

Portanto, a projeção de uma instância $\phi(x)$ é dada por

$$M\phi(x) = \frac{1}{n}X_s'\Phi^T U\tilde{D}^{-1}U^T\phi(x) = X_s'K_sA\Gamma^{-1}A^T k_x, \tag{3}$$

onde Γ^{-1} é a matriz diagonal cujas entradas são $1/\gamma_i$.

É notável que, entre as técnicas de projeção descritas aqui, a Kelp é a que possui a base matemática mais sofisticada. Entretanto, sua implementação é tão simples quanto as demais técnicas e requer apenas o que está apresentado na equação (3), ou seja, precisamos apenas da matriz do *kernel*, seus autovalores e autovetores e as projeções dos pontos de controle.

A Kelp é uma técnica de projeção focada em dados caracterizados pela similaridade dada por *kernel*. Dessa forma, como é possível ver na equação (3), a Kelp não faz uso das coordenadas euclidianas dos dados no espaço original, necessitando apenas da similaridade dada pelo *kernel*. Naturalmente, se os dados não estão caracterizados por um *kernel*, ainda podemos fazer uso da Kelp "kernelizando" os dados, ou seja, basta calcular a matriz K do *kernel* usando uma função de *kernel*.

O *kernel* escolhido dita a força da Kelp, pois é o *kernel* que determina a relação de similaridade que será utilizada na projeção. Assim, é possível obter resultado excelentes ao utilizar o *kernel* adequado ou resultados não tão bons (ou até mesmo desastrosos) se utilizarmos um *kernel* genérico ou inadequado.

Experimentos

Nesta seção apresentaremos algumas projeções utilizando as técnicas descritas acima, os conjuntos de dados utilizados nessas projeções e algumas métricas que servem para quantificar uma projeção.

Os conjuntos de dados

Iris: O conjunto de dados Iris (FRANK; ASUNCION, 2010) é composto por 150 instâncias de dado com dimensão 4, cada uma representando um exemplar da flor Iris. A partir de cada uma das flores foram extraídas quatro características, a saber, comprimento e largura da pétala e da sépala. A partir dessas características é possível classificar as flores em três espécies: setosa, virgínica e versicolor.

WBCD: O segundo conjunto de dados que utilizaremos corresponde a resultados de exames de câncer de mama, *Wisconsin Breast Cancer Database* (FRANK; ASUNCION, 2010). Foram catalogados 699 resultados de exames, cada um com 9 atributos e classificados em benigno ou maligno.

As métricas

A primeira métrica é o *stress* e é responsável por indicar o quanto das distâncias entre instâncias de dado no espaço original foram preservadas na projeção. A função de *stress* que utilizaremos é dada da seguinte forma:

$$\frac{1}{\sum_{ij} d_{ij}} \sum_{ij} \frac{(d_{ij} - d'_{ij})^2}{d_{ij}^2},$$

onde d_{ij} e d'_{ij} são as distâncias entre as instâncias i e j no espaço original e no espaço visual, respectivamente. Se estamos interessados em preservar as distâncias, a diferença no numerador deve ter sempre o menor valor possível para que a projeção tenha tal característica, ou seja, quanto menor o valor do *stress* (sempre não negativo) melhor a preservação das distâncias.

A segunda conta, para cada instância de dado, quantos dos k vizinhos mais próximos no espaço original estão entre os k vizinhos mais próximos no espaço visual. Essa métrica é conhecida como *neighborhood preservation*.

A última mética é conhecida como *silhueta* e mede a coesão e separabilidade das classes de dados. Sendo a_i (coesão) a média das distâncias no espaço visual da instância i às outras instâncias pertencentes a mesma classe, b_i (separação) a menor das distâncias no espaço visual entre a instância i e as instâncias pertencentes as outras classes, o valor da silhueta é dado por

$$\frac{1}{n} \sum_{i} \frac{b_i - a_i}{\max\{a_i, b_i\}},$$

em que n é o número de instâncias no conjunto de dados. Os valores da silhueta variam no intervalo $[-1, 1]$ e quanto maior o valor, melhor a coeção e separabilidade das classes.

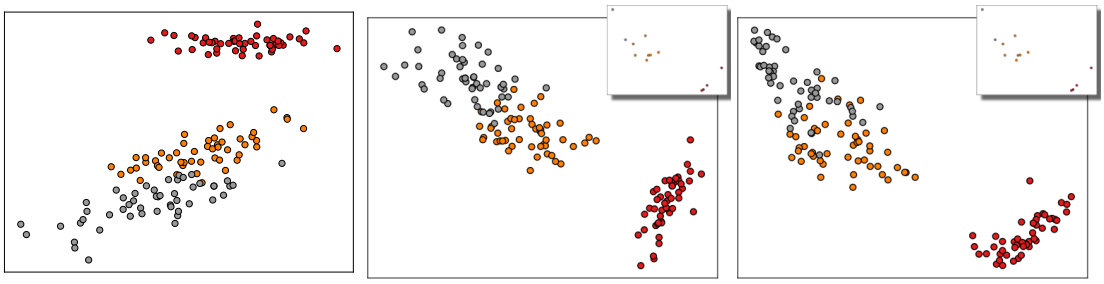
As métricas servem para quantificar as projeções. São utilizadas para comparar duas projeções entre si e não para definir um limiar de qualidade por si só.

Resultados e comentários

Os resultados apresentados aqui têm como objetivo validar as caractarísticas individuais das técnicas de projeção, uma vez que, pela maneira como foram formuladas, cada técnica tem características próprias que foram priorizadas na projeção. Dessa forma, apontaremos as vantagens e desvantagens de cada técnica utilizando as métricas para quantificar essas diferenças. As projeções da Kelp foram feitas usando o *kernel* Gaussiano $k(x_i, x_j) = exp(-\|x_i - x_j\|^2/2\sigma^2)$, onde o valor de σ foi obtido pela variância média das instâncias do conjunto de dados. As projeções dos pontos de controle foram calculadas utilizando

a *Force Scheme* com 10 iterações. O número de pontos de controle utilizados em todas as projeções foi escolhido como o menor inteiro maior que $\sqrt{n}$, onde n é o número de instâncias do conjunto de dados.

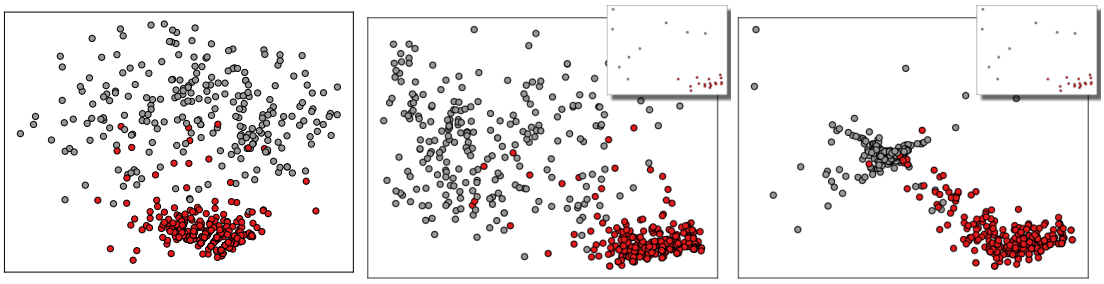
Analisando as projeções visualmente, a Figura 2 mostra que uma das espécies (grupo vermelho) da flor Iris fica totalmente separada, enquanto que as outras duas estão um pouco embaralhadas. Esse é o comportamento esperado de qualquer projeção desse conjunto de dados, pois, segundo os quatro atributos que estamos utilizando, biologicamente duas das espécies são mais semelhantes entre si. Os quadros no canto direito superior são a projeção dos pontos de controles utilizados para gerar as projeções.



(a) Projeção com a *Force Scheme*. (b) Projeção com a LAMP. (c) Projeção com a Kelp.

Figura 2: Projeção do conjunto de dados Iris.

Para o conjunto de dados WBCD (Figura 3) temos uma separação não tão clara das duas classes (benigno e maligno), mas ainda é possível distinguir dois grupos, um mais coeso e outro mais espalhado. Dessa forma, para ambos os conjuntos de dado, visualmente, qualquer uma das três técnicas de projeção reflete os padrões reais do conjunto de dados.



(a) Projeção com a *Force Scheme*. (b) Projeção com a LAMP. (c) Projeção com a Kelp.

Figura 3: Projeção do conjunto de dados WBCD.

A Tabela 1 resume os resultados das três métricas e valida as características individuais de cada técnica de projeção. Para o conjunto de dados Iris, o menor valor do *stress* é obtido na projeção da *Force Scheme*, como era esperado, além do melhor resultado quanto a preservação das relações de vizinhança. Entretanto, essa é a técnica que tem maior tempo de execução, seguida da LAMP (numa escala muito menor) já que essa última tem que calcular uma transformação afim para cada ponto a ser projetado. O valor destoante do *stress* no conjunto de dados WBCD para a *Force Scheme* se deve ao fato de que as 10 iterações do algoritmo não foram suficientes para que ele posicionasse os pontos no espaço visual.

Os resultados da Kelp apresentados, apesar de semelhantes aos demais, não fazem jus a capacidade da técnica, pois utilizamos um mesmo *kernel* sem refinamento algum e para conjuntos de dados de naturezas bem distintas. Ainda assim, a Kelp apresentou o melhor resultado para a silhueta no conjunto de dados WBCD, indicando que a coesão e separação dos grupos no espaço original deve ser mais próximo da projeção da Kelp do que das demais projeções.

Tabela 1: Resultados das métricas. As colunas indicam o valor do *stress*, quantidade de vizinhos preservados, silhueta e tempo de execução de cada uma das técnicas para os conjuntos de dados indicados.

Conj. de dados	Técnica	Stress	NP	Silhueta	Tempo
Iris	Force	**0.00669**	**91.7%**	0.51596	5.06s
	LAMP	0.01068	90.6%	**0.54943**	0.17s
	Kelp	0.03392	77.2%	0.49489	**0.01s**
WBCD	Force	1.7e+36	**77.2%**	0.62925	115s
	LAMP	**0.00949**	74.8%	0.66633	1.59s
	Kelp	0.02044	63.5%	**0.69416**	**0.06s**

Conclusão

Nesse trabalho apresentamos os conceitos fundamentais para o estudo de projeções multidimensionais, tais como caracterização dos dados, exemplos de técnicas de projeção multidimensional e métricas para quantificação das projeções.

Vimos que, apesar dos resultados visuais das três técnicas serem semelhantes, as métricas mostram que cada uma tem características próprias, seja quanto ao que se deseja priorizar na projeção ou, simplesmente, ao tempo de execução. Dessa forma, não é possível determinar uma técnica que seja melhor que as demais em todas as situações, ficando a cargo do usuário decidir a mais adequada ao seu problema.

Referências

BARBOSA, A. et al. Visualizing and interacting with kernelized data. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 3, p. 1314–1325, 2016.

FRANK, A.; ASUNCION, A. UCI Machine Learning Repository. 2010. Disponível em: ⟨http://archive.ics.uci.edu/ml⟩. Acesso em: 13 abr. 2019.

GOWER, J.; DIJKSTERHUIS, G. **Procrustes Problems**. Oxford: Oxford University Press, 2004. (Oxford Statistical Science Series).

JOIA, P. et al. Local affine multidimensional projection. **IEEE Transactions on Visualization and Computer Graphics**, v. 17, n. 12, p. 2563–2571, 2011.

LIMA, E. **Álgebra linear**. Rio de Janeiro: IMPA, 1995. (Coleção matemática universitária).

PAULOVICH, F. V. et al. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. **IEEE Transactions on Visualization and Computer Graphics**, v. 14, n. 3, p. 564–575, 2008.

SCHÖLKOPF, B. et al. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. Cambrige: MIT Press, 2002. (Adaptive computation and machine learning).

TEJADA, E.; MINGHIM, R.; NONATO, L. G. On improved projection techniques to support visual exploration of multidimensional data sets. **Information Visualization**, v. 2, n. 4, p. 218–231, 2003.