UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MATHEUS FELIPIN YOKOYAMA

COMPARATIVO VISUAL E ANALÍTICO DE ALGORITMOS DE REDUÇÃO DE DIMENSIONALIDADE

PATO BRANCO

2023

MATHEUS FELIPIN YOKOYAMA

COMPARATIVO VISUAL E ANALÍTICO DE ALGORITMOS DE REDUÇÃO DE DIMENSIONALIDADE

VISUAL AND ANALYTICAL COMPARISON OF DIMENSIONALITY REDUCTION ALGORITHMS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Dalcimar Casanova

PATO BRANCO

2023

MATHEUS FELIPIN YOKOYAMA

COMPARATIVO VISUAL E ANALÍTICO DE ALGORITMOS DE REDUÇÃO DE DIMENSIONALIDADE

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de Aprovação: 22 de Junho de 2023.

Prof. Dr. Dalcimar Casanova
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luiz Fernando Puttow Southier
Universidade Tecnológica Federal do Paraná

Profa. Dra. Viviane Dal Molin
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2023

RESUMO

Dados de alta dimensionalidade geralmente apresentam desafios quando se tratam de visualização e utilização em algoritmos de classificação. O processamento dos dados pode ser demorado e exigir grande poder computacional. Todavia, existem dados de baixa dimensionalidade que possuem estruturas utilizadas no campo conhecido como Manifold Learning. Para estudar as estruturas dos dados, bem como reduzir a entrada para classificadores, otimizar o tempo de processamento e diminuir a complexidade dos dados, existem algoritmos de redução de dimensionalidade. Neste trabalho, utilizou-se uma seleção de algoritmos como PCA, MDS, Isomap, LLE, Random Trees, t-SNE e Autoencoder para identificar a abordagem que melhor representava as informações de forma visual, verificando se a natureza dos dados era afetada por distorções que ocorriam no processo de redução de dimensionalidade e avaliou-se a qualidade dos resultados de acordo com a métrica Estresse de Kruskal. Para isso, fez-se o uso das bases de dados sintéticas como Rolo Suíço, Curva S, Hello, e bases reais como MNIST e Iris. Realizou-se a aplicação das mesmas nos algoritmos de redução de dimensionalidade. Os resultados deste trabalho demonstraram que, de acordo com o Estresse de Kruskal, os algoritmos obtiveram uma distorção, exceto PCA para a base Hello. Quanto à comparação visual, para as bases Rolo Suíço, Curva S, Hello e Iris, os algoritmos PCA e MDS resultaram em uma melhor visualização interpretativa, enquanto para MNIST, o t-SNE obteve esse resultado, diferentemente dos demais algoritmos para ambas as bases.

Palavras-chave: Aprendizagem de Manifold; t-SNE; PCA; Visualização de dados; Estresse de Kruskal.

ABSTRACT

High-dimensional data often pose challenges in terms of visualization and utilization in classification algorithms. Processing such data can be time-consuming and computationally demanding. However, there exist low-dimensional datasets that exhibit structures utilized in the field known as Manifold Learning. To study the structures of the data, as well as reduce the input for classifiers, optimize processing time, and decrease data complexity, dimensionality reduction algorithms are employed. In this work, a selection of algorithms including PCA, MDS, Isomap, LLE, Random Trees, t-SNE, and Autoencoder were used to identify the approach that best represented the information visually, while assessing whether the nature of the data was affected by distortions occurring during the dimensionality reduction process. Synthetic datasets such as Swiss Roll, S Curve, and Hello, as well as real datasets like MNIST and Iris, were utilized to apply these dimensionality reduction algorithms. The results of this study showed that, based on Kruskal Stress metric, the algorithms exhibited distortions, except for PCA on the Hello dataset. In terms of visual comparison, for Swiss Roll, S Curve, Hello, and Iris datasets, PCA and MDS algorithms yielded better interpretability, whereas t-SNE achieved superior results for MNIST, contrasting with other algorithms for both datasets.

Keywords: Manifold Learning; t-SNE; PCA; Data Visualization; Kruskal Stress.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

2D	Bidimensional
3D	Tridimensional
KL	Kullback-Leibler
LLE	Local Linear Embedding
MDS	Multidimensional Scaling
PCA	Principal Component Analisys
RD	Redução de Dimensionalidade
SNE	Stochastic Neighbor Embeddeding
t-SNE	T-Student Stochastic Neighbor Embedding

LISTA DE SÍMBOLOS

LETRAS LATINAS

$cov(X)$	Matriz de covariância de X
M	Autovetor

LETRAS GREGAS

η	Taxa de aprendizagem
γ	Mapa de baixa dimensão
λ	Autovalor

SUMÁRIO

1 INTRODUÇÃO

No mundo atual, com a constante evolução da tecnologia, a quantidade de dados disponíveis tem sido utilizada de forma inteligente em diversos campos, desde a Astrofísica até a Medicina. Esses dados podem conter informações valiosas para serem utilizadas em aprendizado de máquina e visualizações de dados para análise e identificação de agrupamentos.

Dados do mundo real, como fotografias digitais, sinais de áudio e imagens de ressonância magnética, costumam conter alta dimensionalidade. Para a análise de dados e reconhecimento de padrões, um processo simples realizado é o de redução de dimensionalidade (KASKI; PELTONEN, 2011). Esse processo, para aplicações no âmbito de *Machine Learning*, é conhecido como Redução de Dimensionalidade (RD). RD, ou redução de dimensionalidade, nada mais é do que um método de obtenção de um conjunto de características reduzidas, seja por um determinado critério de seleção ou transformação. Entretanto, a representação reduzida deve corresponder à dimensionalidade intrínseca desses dados. Essa dimensionalidade intrínseca é, basicamente, a quantidade mínima de características necessárias para poder representar as propriedades dos dados observados (MAATEN; POSTMA; HERIK, 2007).

Todavia, dados de alta dimensionalidade costumam apresentar problemas em aplicações de *Machine Learning*. Analisar e processar dados com muitas características pode ser um processo lento, computacionalmente custoso e de difícil representação visual. Entre os problemas com dados de alta dimensionalidade, existe algo conhecido como *The Curse of Dimensionality* (REMESH; V, 2017). *The Curse of Dimensionality*, ou Maldição da Dimensionalidade, é definido como o aumento exponencial da quantidade necessária de dados para categorizar com precisão à medida que o número de características ou a dimensionalidade aumenta (REMESH; V, 2017).

Portanto, a escolha da técnica de redução de dimensionalidade a ser utilizada depende da aplicação específica. Entre essas técnicas, a principal diferença entre seleção e transformação de características é que, no método de seleção, busca-se identificar o melhor subconjunto de características do conjunto original de dados, enquanto no método de transformação, ocorre a criação de um novo conjunto de características projetadas para preservar estruturas locais ou globais dos dados por meio do processo de transformação.

Geralmente, reduzir-se a dimensão dos dados busca retirar redundância nas características, reduzir o espaço de armazenamento, diminuir o tempo de computação, melhorar os resultados de classificações dos algoritmos de aprendizagem e obter uma melhor visualização das estruturas presentes nos dados, as quais são de difícil observação no mapa de alta dimensão

(REMESH; V, 2017).

Para esse propósito, existem técnicas lineares de transformação de características, como Análise de Componentes Principais (*Principal Component Analysis* - PCA) (PEARSON, 1901) e Escalonamento Multidimensional Clássico (*Classical Multidimensional Scaling* - MDS) (KRUSKAL; WISH, 1978), além de técnicas não lineares como Projeção de Vizinhança Estocástica (*Stochastic Neighbor Embedding* - SNE) (HINTON, G. E.; ROWEIS, 2003), Mapeamento de Sammon (*Sammon Mapping*) (SAMMON, 1969), Isomap (SILVA; TENENBAUM, 2003), entre outras que visam preservar estruturas locais (MAATEN; HINTON, G., 2008). Serão utilizadas algumas dessas técnicas para melhor compreender os resultados obtidos ao utilizá-las.

1.1 JUSTIFICATIVA

O processo de redução de dimensionalidade desempenha um papel crucial no campo do *Machine Learning*. Quando lida-se com conjuntos de dados de alta dimensionalidade como entrada para classificadores, por exemplo, isso pode afetar tanto os resultados quanto o tempo de processamento. No entanto, é importante destacar que, ao reduzir a dimensionalidade de uma base de dados, introduz então distorções no processo.

Nesse contexto, o objetivo deste trabalho é aplicar algoritmos de redução de dimensionalidade em conjuntos de dados com estruturas internas, reduzindo-os para uma projeção em 2D ou 3D, além de compreender as distorções que ocorrem ao reduzir a dimensionalidade desses conjuntos de dados.

1.2 OBJETIVO GERAL

O objetivo deste trabalho é aplicar algoritmos em diferentes conjuntos de dados para a redução de dimensionalidade e analisar as distorções resultantes na visualização em um plano 2D.

1.2.1 OBJETIVOS ESPECÍFICOS

- Selecionar diferentes conjuntos de dados, incluindo bases sintéticas e bases de dados reais, que possuem estruturas visíveis ou propriedades internas, para a aplicação dos algoritmos de redução de dimensionalidade;

- Analisar os resultados obtidos após a redução de dimensionalidade por meio da visualização dos dados no plano 2D, investigando a capacidade dos algoritmos em preservar a estrutura e as relações entre os pontos no espaço de menor dimensionalidade;
- Utilizar o Estresse de Kruskal para realizar comparações entre os resultados obtidos pelos diferentes algoritmos de redução de dimensionalidade. Essa métrica permite uma avaliação quantitativa das distorções e da qualidade das projeções em relação aos dados originais, facilitando a conclusão sobre a adequação da métrica aplicada para uso comparativo, tanto visual quanto quantitativamente;

1.3 ORGANIZAÇÃO DO TRABALHO

A estrutura do trabalho consiste da seguinte forma:

1. Fundamentação Teórica (Capítulo 2): são apresentadas informações relacionadas à compreensão de dados de alta dimensão, gráficos utilizados, técnicas existentes para redução de dimensionalidade, algoritmos de redução de dimensionalidade e métricas para análise da redução de dimensionalidade;
2. Materiais e métodos (Capítulo 3): são abordadas informações sobre a base de dados utilizada para os testes e os resultados obtidos a partir da aplicação das técnicas de redução de dimensionalidade;
3. Análise e discussão dos resultados (Capítulo 4): são abordadas informações relacionadas a análise dos resultados obtidos e comparações entre os algoritmos de redução de dimensionalidade;
4. Conclusão (Capítulo 5): são apresentadas as conclusões e considerações finais sobre os resultados obtidos, encerrando o trabalho com uma análise crítica dos achados e possíveis direções para pesquisas futuras;

2 FUNDAMENTAÇÃO TEÓRICA

2.1 DADOS DE ALTA DIMENSÃO

A Ciência, como um campo de pesquisa que realiza a obtenção de informações que necessitam ser armazenadas, tem se aproveitado da vantagem de computadores com capacidade massiva de armazenamento para guardar dados. A Biologia, por exemplo, com a capacidade de medir expressão genética em microarranjo de DNA, tem produzido conjuntos de dados imensos, juntamente com dados de transcrição proteica e árvores filogenéticas que relacionam espécies entre si (MARSLAND, 2014).

Esses dados, considerados como dados de alta dimensionalidade, costumam conter uma quantidade significativa de características. Um conjunto de dados como este contém informações úteis para compreender os dados, mas, por outro lado, também contém características irrelevantes e redundantes que reduzem o desempenho, a qualidade dos dados e a eficiência computacional (JINDAL; KUMAR, 2017).

No entanto, a representação desses dados pode ser obtida na forma de um gráfico de dispersão, que permite apresentar, a partir do relacionamento entre duas ou três características em coordenadas cartesianas, os valores de um conjunto de dados. Contudo, quando um conjunto de dados comporta quatro características ou mais, a representação do relacionamento entre todas as características torna-se impossível visualmente, pois ultrapassa o limite do espaço compreensível visualmente (PREETI SHARMA, 2017).

No âmbito de *Machine Learning*, as pessoas costumam inserir mais características em seus dados para buscar melhores resultados em suas classificações. No entanto, a adição de características pode resultar em um problema conhecido como maldição da dimensionalidade. Esse problema se aplica a essa área, de modo que, quanto maior o número de dimensões, mais dados são necessários para permitir que os algoritmos generalizem suficientemente bem. Por essa razão, é necessário ter cuidado com as informações fornecidas aos algoritmos e é essencial possuir um conhecimento prévio sobre os dados (MARSLAND, 2014).

2.2 VISUALIZAÇÃO DE DADOS

Nesta seção, abordam-se diferentes formas de transformar dados que estão na forma numérica para uma representação visual, a fim de obter uma melhor compreensão das estruturas existentes nos dados.

A visualização de dados é definida como a representação visual obtida a partir dos dados, sendo que esses dados são mapeados de forma numérica e traduzidos para uma determinada representação gráfica. Assumindo-se que os dados possuem n-dimensões, sendo um número inteiro positivo, todas as visualizações são projetadas em uma superfície de exibição. Da mesma forma, existem poucos métodos de exibição em 3D que também se aplicam a essa superfície de exibição. No entanto, todo o plano de visualização dos dados de n-dimensões é reduzido para 2D. Embora isso seja correto, existe uma maneira de diferenciar a dimensionalidade entre um meio físico e a representação lógica. Um exemplo que pode ser considerado é o gráfico de dispersão em 3D (GRINSTEIN; TRUTSCHL; CVEK, 2002).

2.2.1 TABELA

Como uma definição para a visualização em forma de tabelas, as linhas se referem ao registro dos dados e as colunas são conhecidas como dimensões, variáveis, atributos ou características desses dados. Portanto, esses dados podem ser representados por diferentes tipos, como inteiros, reais, categóricos, nominais, entre outros. No entanto, na maioria das visualizações, todos os dados são convertidos para um tipo real antes de serem interpretados para determinada visualização (HOFFMAN; GRINSTEIN, 2001).

Abaixo, na Tabela 1, é possível visualizar um exemplo com atributos de valores binários e na Tabela 2, um exemplo retirado do conjunto de dados Iris.

Tabela 1 – Exemplo de tabela de dados com características binárias

Índice	Característica 1	Característica 2	Característica 3	Característica 4	Característica 5
Classe 2	0	1	1	0	1
Classe 3	0	1	1	1	1
Classe 1	1	0	1	0	1
Classe 2	1	0	1	1	1
Classe 1	1	0	0	0	0
Classe 3	0	1	1	1	0

Fonte: Autoria própria.

Tabela 2 – Exemplo retirado do conjunto de dados Iris

Índice	Comprimento sépala (cm)	Largura sépala (cm)	Comprimento pétala (cm)	Largura pétala (cm)
Classe 0	5.1	3.5	1.4	0.2
Classe 0	4.9	3.0	1.4	0.2
Classe 1	6.4	2.9	4.3	1.3
Classe 1	6.6	3.0	4.4	1.4
Classe 2	5.9	3.0	5.1	1.8

Fonte: Autoria própria.

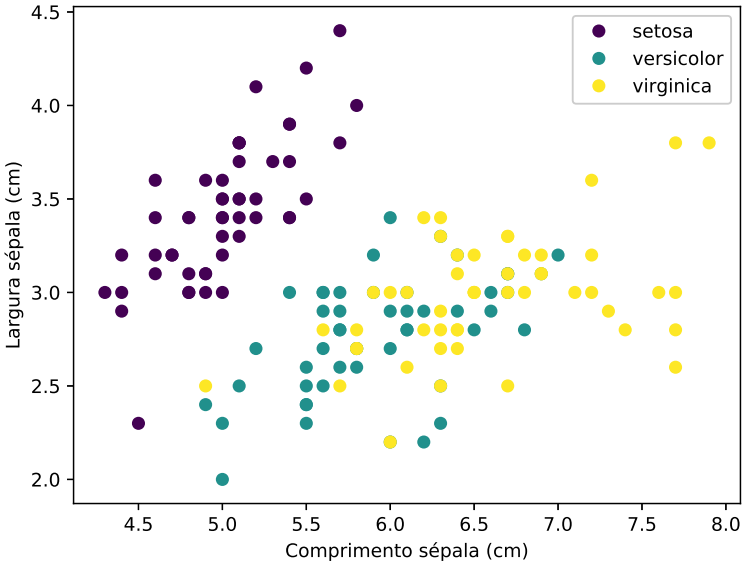
2.2.2 GRÁFICO DE DISPERSÃO

Scatter Plot ou Gráfico de dispersão é um método de projeção de pontos em espaços bidimensionais (2D) ou tridimensionais (3D) no formato clássico de (X,Y) ou (X,Y,Z) (GRINSTEIN; TRUTSCHL; CVEK, 2002).

Esse tipo de visualização ajuda a encontrar agrupamentos, pontos discrepantes ou valores anormais, tendências e correlações. De outra forma, a limpeza e coloração dos pontos por classe podem ser usadas para tornar a compreensão dos dados mais intuitiva. Quando muitos pontos estão sobrepostos ou próximos devido à resolução, cria-se a ilusão de que os pontos estão na mesma coordenada (x,y), o que pode ser melhorado ampliando ou realizando um deslocamento dos dados (HOFFMAN; GRINSTEIN, 2001).

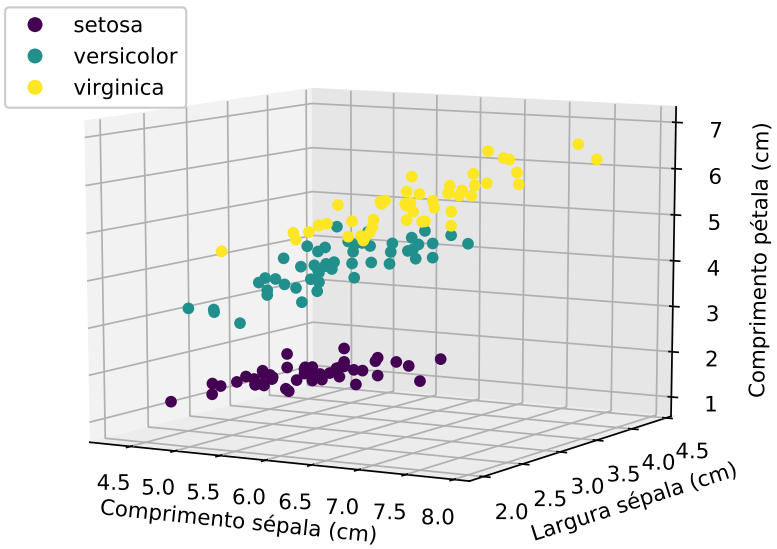
As estruturas presentes nos dados podem ser observadas na Figura 1 e Figura 2, onde é possível identificar os agrupamentos das classes 0, 1 e 2. No entanto, nesse caso, pode-se observar e perceber que existem três classes diferentes, pois os dados são rotulados. No caso em que não há rótulos, tem-se a Figura 3, onde ainda é possível observar agrupamentos. No entanto, nas regiões em que ocorre sobreposição entre dois agrupamentos, torna-se impossível distinguir entre os dois grupos.

Figura 1 – Gráfico de dispersão do conjunto de dados iris 2D com rótulo



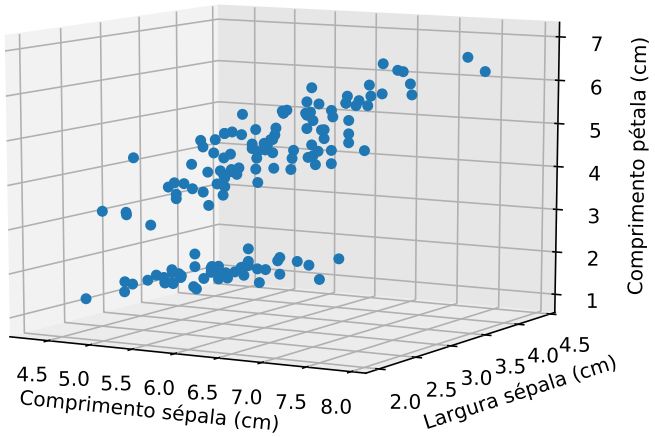
Fonte: Autoria própria.

Figura 2 – Gráfico de dispersão do conjunto de dados iris 3D com rótulo



Fonte: Autoria própria.

Figura 3 – Gráfico de dispersão do conjunto de dados iris 3D sem rótulo



Fonte: Autoria própria.

2.3 REDUÇÃO DE DIMENSIONALIDADE

Redução de dimensionalidade (RD) é um tipo de pré-processamento, também conhecido como uma forma de redução de características dos dados, que transforma dados de alta dimensionalidade em dados de baixa dimensionalidade, sem comprometer a essência na representação desses dados.

No entanto, a representação obtida pela redução deve apresentar a essência dos dados, e a quantidade mínima de características necessárias para representar essa essência é chamada de Dimensionalidade Intrínseca (FUKUNAGA, 1990).

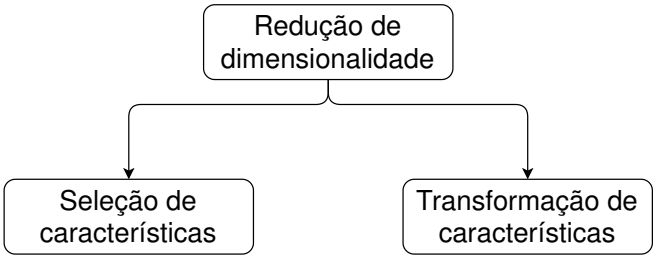
Por exemplo, considerando um conjunto de dados representado em uma matriz X de tamanho nxD, onde n é a quantidade de vetores de dados e D é a quantidade de características. Assim, tem-se x_i como o vetor de dados, em que ($i \in \{1,2,...,n\}$) com dimensão D. Novamente, supondo que haja uma dimensionalidade intrínseca d nesse conjunto de dados (em que $d<D$, e frequentemente $d \ll D$), essa dimensionalidade intrínseca significa que os pontos no conjunto de dados X estão sobre ou próximos a um objeto com dimensionalidade d, que está localizado no espaço dimensional D (MAATEN; POSTMA; HERIK, 2007).

Portanto, as técnicas de RD visam alterar um conjunto de dados X com dimensionalidade

D, para um novo conjunto de dados Y com dimensionalidade d, tentando manter a geometria dos dados o máximo possível. Em geral, não se tem conhecimento prévio sobre a geometria do objeto de dados nem sobre a dimensionalidade intrínseca d. No entanto, a RD é um problema bem-definido que pode ser resolvido apenas quando se assume uma determinada propriedade dos dados (MAATEN; POSTMA; HERIK, 2007). Os vetores de dados de alta dimensionalidade são representados como x_i, em que x_i é a enésima linha da matriz X de dimensão D. Por outro lado, no espaço de baixa dimensão, utiliza-se a notação dos vetores de dados como y_i, em que y_i é a enésima linha da matriz Y de dimensão d (MAATEN; POSTMA; HERIK, 2007).

Entre os métodos de RD, existem dois tipos: seleção de características (*feature selection*) e transformação de características (*feature extraction*). A Figura 4 apresenta um diagrama dos métodos de RD. O método de seleção de características é abordado na seção Seção 2.3.1, e o método de transformação de características é abordado na seção Seção 2.3.2.

Figura 4 – Métodos de RD



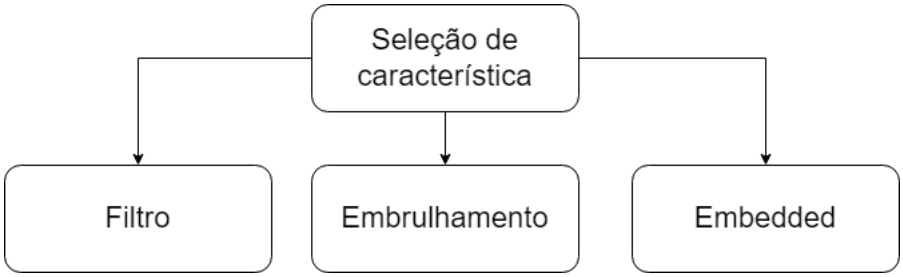
Fonte: Adaptado de Maaten, Postma e Herik (2007).

2.3.1 SELEÇÃO DE CARACTERÍSTICAS

O processo de seleção de características consiste em quatro etapas: geração, avaliação, critério de parada e validação dos resultados de cada subconjunto. Em outras palavras, na etapa de geração, busca-se por um subconjunto de características candidatas para avaliação, com base em uma estratégia de pesquisa específica. Cada subconjunto é avaliado e comparado com os melhores subconjuntos anteriores, de acordo com um critério de avaliação pré-determinado. Esse processo é repetido, de modo que, sempre que uma solução melhor é encontrada, substitui-se a melhor solução anteriormente encontrada, até que um critério de parada seja satisfeito. Portanto, o melhor subconjunto encontrado requer uma validação, que é realizada por meio de um conjunto de dados de teste (LIU; YU, 2005 apud PIROLLA, 2012).

As técnicas de seleção de características podem ser subdivididas em três principais categorias: Filtro, Embrulhamento e Embedded. A Figura 5 apresenta um diagrama das técnicas de seleção de características.

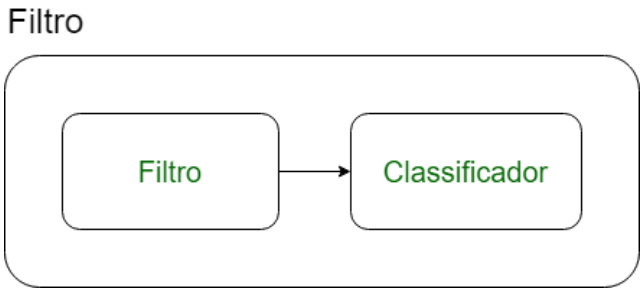
Figura 5 – Técnicas de seleção de características



Fonte: Adaptado de Miao e Niu (2016).

- Método Filtro: a técnica de filtro utiliza-se de heurísticas baseadas em características gerais de um dado, de maneira oposta a de um algoritmo de aprendizado para validar o valor do subconjunto de características. Consequentemente esta é, geralmente, mais rápida que a técnica de Embrulhamento e mais práticos para uso em dados de alta dimensionalidade (HALL, 1999).

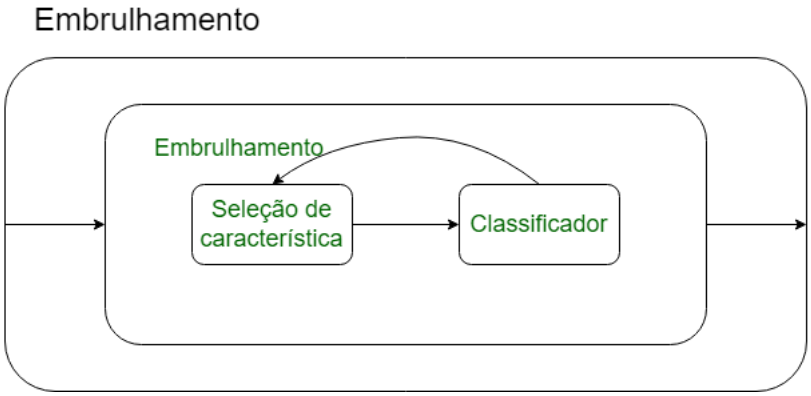
Figura 6 – Fluxograma técnica de Filtro



Fonte: Adaptado de Bolón-Canedo, Sánchez-Maroño e Alonso-Betanzos (2012).

- Método de Embrulhamento: esta técnica tem como base algoritmos indutivos para estimar o valor do subconjunto de características. A justificativa para o uso da técnica de embrulhamento é que o método indutivo irá, em última instância, utilizar o subconjunto de características para fornecer a melhor estimativa da acurácia, ao contrário de uma medida separada que possui um viés indutivo completamente diferente (BLUM; LANGLEY, 1997). Essa técnica tem maior probabilidade de obter melhores resultados do que a técnica de filtro, devido à interação entre o algoritmo indutivo e os dados de treinamento. No entanto, ela costuma ser mais lenta que a técnica de filtro, pois requer a chamada repetida do algoritmo indutivo e precisa ser executada novamente quando um novo algoritmo é utilizado (HALL, 1999).

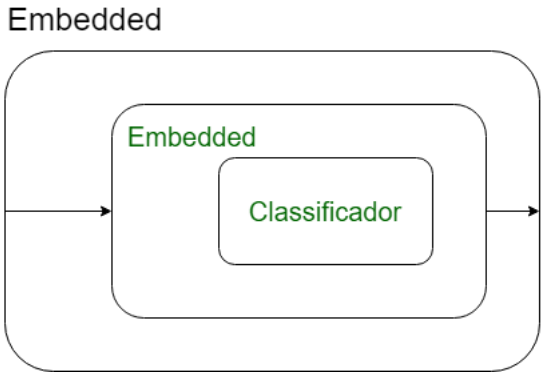
Figura 7 – Fluxograma técnica de embrulhamento



Fonte: Adaptado de Bolón-Canedo, Sánchez-Maroño e Alonso-Betanzos (2012).

- Método Embedded: esta técnica difere de outras técnicas de seleção de características na forma como ocorre essa seleção e a interação de aprendizagem. Ao contrário do Método de Filtro e do Método de Embrulhamento, o Método Embutido não realiza a segregação do aprendizado da parte de seleção de características (GUYON *et al.*, 2006).

Figura 8 – Fluxograma do técnica de projeção



Fonte: Adaptado de Bolón-Canedo, Sánchez-Maroño e Alonso-Betanzos (2012).

2.3.2 TRANSFORMAÇÃO DE CARACTERÍSTICAS

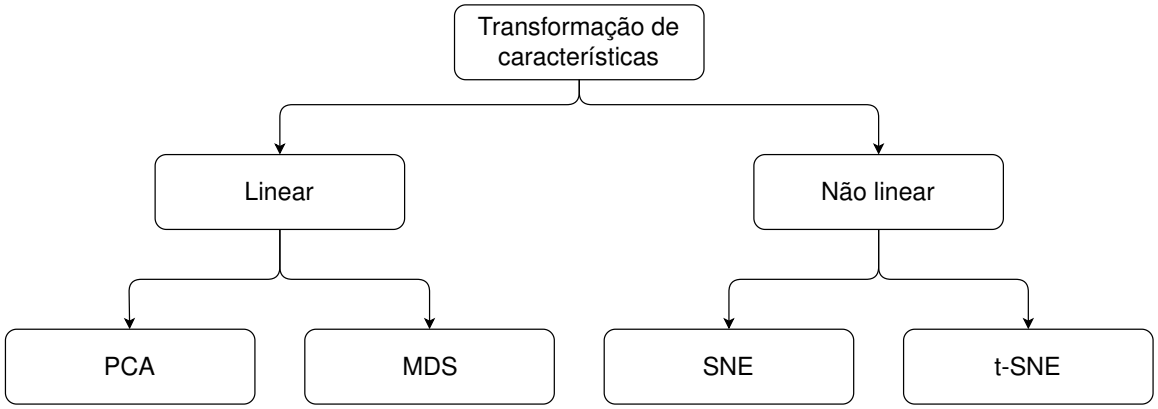
Neste método de redução, simplifica-se as mudanças nos eixos do sistema de coordenadas do gráfico por meio de movimentação ou rotação do conjunto de dados, o qual pode ser descrito por uma simples matriz. Portanto, a motivação para o uso deste método de redução vem do fato de permitir realizar combinações das características e identificar quais são úteis e quais não são (MARSLAND, 2014).

Diversas técnicas têm sido propostas para o problema de RD, diferindo no tipo de

estrutura que elas preservam. Cada técnica realiza a redução de diferentes maneiras. Por exemplo, o PCA minimiza a redundância medindo a covariância e maximiza as informações por meio da variância (NASREEN, 2014). Por outro lado, o MDS clássico utiliza a distância Euclidiana para medir a dissimilaridade entre pontos e realizar o mapeamento no mapa de baixa dimensão (TORGERSON, 1952). O algoritmo do SNE tenta posicionar os objetos no mapa de baixa dimensão de forma a preservar as identidades da vizinhança de maneira ótima (HINTON, G. E.; ROWEIS, 2003). Entre outras técnicas não lineares que visam preservar estruturas locais dos dados, como *Sammom Mapping* e *Isomap*, conseguem obter bons resultados quando utilizadas em dados artificiais. No entanto, quando aplicadas em dados reais de alta dimensão, essas técnicas não são bem-sucedidas em obter visualizações dos dados de alta dimensão. Assim, a maioria das técnicas não é capaz de preservar as estruturas locais e globais dos dados em um único mapa (MAATEN; HINTON, G., 2008).

Existem técnicas de transformação de características lineares e não lineares. Dentre as lineares, o PCA é a mais conhecida, descrita na seção Seção 2.3.2.1. Já nas técnicas não lineares, aborda-se o SNE na seção Seção 2.3.2.2, para que seja possível então compreender a técnica t-SNE. Dessa forma, abaixo na Figura 9 visualiza-se um diagrama para ilustrar as técnicas de transformação de características.

Figura 9 – Métodos de transformação de características



Fonte: Adaptado de Maaten, Postma e Herik (2007).

2.3.2.1 LINEAR

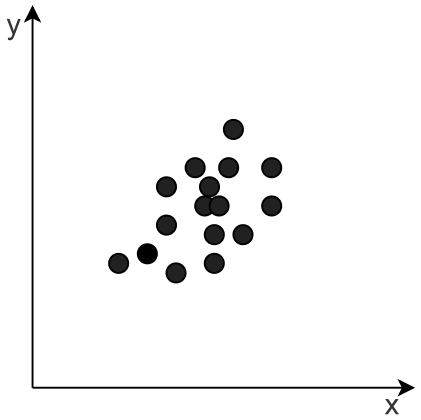
As técnicas lineares são técnicas que permitem realizar RD, adaptando os dados para um espaço linear de baixa dimensionalidade. Apesar de existirem diversas técnicas, neste trabalho aborda-se apenas o PCA como uma referência para as técnicas lineares por ser aplicado em

diversas áreas científicas (WOLD; ESBENSEN; GELADI, 1987).

O componente principal é definido como a direção em que os dados contém a maior variação. Inicialmente o algoritmo centraliza os dados subtraindo a média. Escolhe assim a direção com a maior variação e posiciona um eixo para esta direção. Em seguida, procura-se dentre as variações restantes outro eixo que é ortogonal ao primeiro componente e abrange o máximo possível de variações restantes. Esse processo se repete até ficar sem possíveis eixos para se posicionar no mapa. No fim, o resultado será que, toda a variação ao longo dos eixos do conjunto de coordenadas, assim como a matriz de covariância, será diagonal, isto é, cada característica (variável) nova não é correlacionada umas às outras, com exceção de si própria. Em alguns casos, os últimos eixos encontrados contêm uma variação minúscula. Neste caso pode-se remover estes eixos com variação pequena sem afetar a variabilidade nos dados (MARSLAND, 2014).

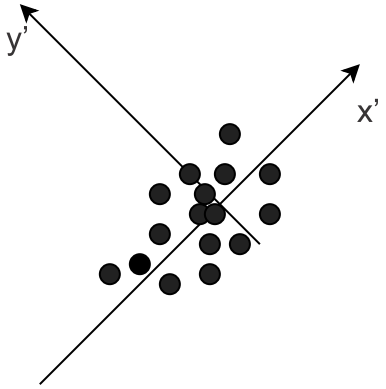
Por exemplo, na Figura 10, os dados estão organizados próximo a um formato elíptico em que os dados estão espalhados em torno de 45º em relação ao eixo x, enquanto na Figura 11, tem-se os eixos deslocados em que os dados seguem acima do eixo x e são centrados na origem. Portanto, conforme observado na Figura 11, a dimensão y não demonstra conter muita variabilidade. Assim, seria possível ignorar esse eixo e utilizar apenas os valores do eixo x sem comprometer os resultados do algoritmo de aprendizagem (MARSLAND, 2014).

Figura 10 – Ilustração de um conjunto de dados



Fonte: Adaptado de Marsland (2014).

Figura 11 – Eixos deslocados da Figura 10



Fonte: Adaptado de Marsland (2014).

Em termos matemáticos, o PCA tenta encontrar um mapeamento linear M que maximiza $M^T cov(X) M$, em que $cov(X)$ é a matriz de covariância dos dados X. O mapeamento linear do PCA é formado por d principais autovetores, em que, estes autovetores são os principais componentes da matriz de covariância dos dados de média zero. Entretanto, PCA resolve o *eigenproblem* com a Equação 1 (MAATEN; POSTMA; HERIK, 2007).

$$cov(X)M = \lambda M \tag{1}$$

O *eigenproblem* é resolvido para os d principais autovalores λ. A representação dos dados de baixa dimensionalidade y_i dos pontos x_i, são computados mapeando eles para uma base linear M, i.e., $Y = (X - \bar{X})M$. Portanto, a principal desvantagem do PCA é que o tamanho da matriz de covariância é proporcional à dimensionalidade dos pontos. Como resultado, o cálculo dos autovetores pode ser inviável para dados de alta dimensionalidade. Em vista disso, no conjunto de dados no qual $n < D$, esta desvantagem pode ser superada calculando os autovetores da matriz de distância Euclidiana quadrática $(X - \bar{X})(X - \bar{X})^T$ em vez de autovetores da matriz de covariância (MAATEN; POSTMA; HERIK, 2007).

2.3.2.2 NÃO LINEAR

Como uma referência para compreender t-SNE, aborda-se nesta seção a técnica de redução conhecida como SNE.

O SNE é uma das técnicas que baseia-se em uma vizinhança provável dos dados para realizar o mapeamento no mapa de baixa dimensão. Este algoritmo procura posicionar a amostra (dado) no espaço de baixa dimensão assim como preservar a identidade dos vizinhos

eficientemente (HINTON, G. E.; ROWEIS, 2003).

Este algoritmo começa convertendo a Distância Euclidiana entre pontos de alta dimensão em probabilidades condicionais que representam a similaridade entre eles. A similaridade entre o ponto x_j para o ponto x_i é dado pela probabilidade condicional, $p_{j|i}$, em que x_i escolherá x_j como seu vizinho caso os vizinhos fossem escolhidos proporcionalmente à densidade probabilística abaixo da Gaussiana centrada em x_i. Para pontos próximos, $p_{j|i}$ é relativamente alta, no qual para pontos distantes, $p_{j|i}$, é quase que infinitesimal para valores razoáveis de variância σ_i da Gaussiana (MAATEN; HINTON, G., 2008).

Matematicamente a probabilidade condicional $p_{j|i}$ é conhecida pela Equação 2.

$$p_{j|i} = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \tag{2}$$

A variância σ_i é a variância da Gaussiana centrada no ponto x_i. O método para determinar o valor de σ_i é apresentado posteriormente nesta seção. Como o único interesse desta técnica é modelar a similaridade entre pares de pontos, o valor de $p_{i|i}$ é definido como zero. Para a baixa dimensão, os pontos y_i e y_j, correspondentes a x_i e x_j na alta dimensão, podem ter a probabilidade condicional similar calculada, denotada por $q_{j|i}$. Novamente, escolhe-se a variância da Gaussiana utilizada no cálculo da probabilidade condicional $q_{j|i}$ como $\frac{1}{\sqrt{2}}$. Dessa forma, modela-se a similaridade dos pontos y_i e y_j no mapa de acordo com a Equação 3 (MAATEN; HINTON, G., 2008).

$$q_{j|i} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)} \tag{3}$$

O interesse é modelar a similaridade entre pares de pontos, define-se então $q_{i|i} = 0$. Se os pontos do mapa y_i e y_j modelam corretamente a similaridade dos pontos x_i e x_j do mapa da alta dimensão, as probabilidades condicionais $p_{j|i}$ e $q_{j|i}$ serão iguais. A medida natural da exatidão na qual, $q_{j|i}$ modela $p_{j|i}$, é a divergência de Kullback-Leibler (KL) (MAATEN; HINTON, G., 2008).

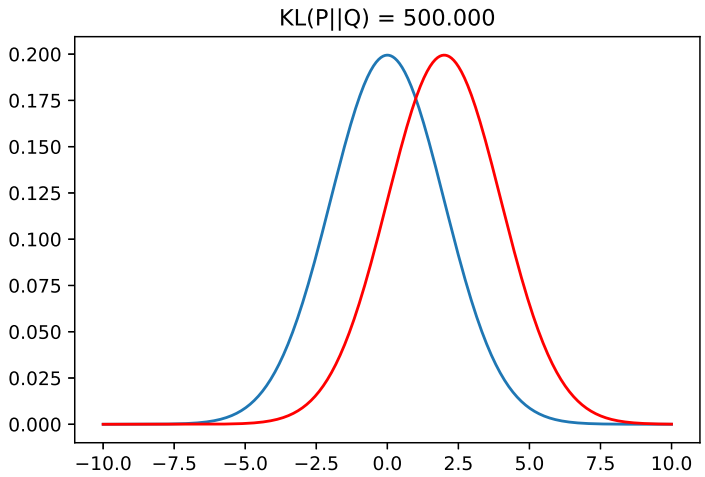
A definição para a divergência de KL é a medida para o quão similar ou diferente duas distribuições probabilísticas são (CLIM; ZOTA; TINICĂ, 2018). Entretanto, SNE minimiza a soma da divergência de KL em relação a todos os pontos usando o método de gradiente descendente. A função de custo C é dado pela Equação 4.

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}} \tag{4}$$

Na Equação 4, P_i representa a distribuição da probabilidade condicional de todos os pontos em relação ao ponto x_i e Q_i representa a distribuição da probabilidade condicional do mapa de todos os outros pontos do mapa em relação ponto y_i. Pelo fato da divergência de KL não ser simétrica, tipos diferentes de erros nos pontos pareados no mapa de baixa dimensão não são ponderados igualmente. Isto pode causar, particularmente, alto custo por usar pontos separados distantemente no mapa de pontos para representar pontos próximos, i.e, utilizar-se de valores pequenos de $q_{j|i}$ para modelar valores grande de $p_{j|i}$. Mas em contrapartida, há apenas um pequeno custo para usar pontos próximos para representar pontos distantes separados. Porém, este custo pequeno vem do desperdício de alguns conjuntos de probabilidade relevante da distribuição Q. Em outras palavras, a função de custo do SNE se destaca em conservar estruturas locais dos dados no mapa para valores aceitáveis de variância, σ_i, na Gaussiana do espaço de alta dimensão (MAATEN; HINTON, G., 2008).

Por exemplo, duas distribuições probabilística P e Q com desvio padrão igual a 2, mas uma delas com média centrada em 0 e a outra em 2, tem sua divergência de KL igual a 500 como observa-se na Figura 12 (MAKLIN, 2019).

Figura 12 – Exemplo das 2 distribuições P e Q

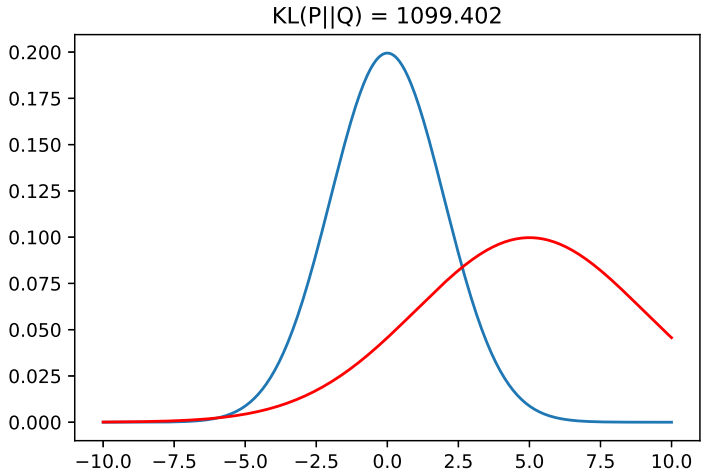


Fonte: Maklin (2019).

Portanto, se medir a divergência da distribuição P que tem a média centrada em 0 com desvio padrão 2, com uma outra distribuição Q que tem a média centrada em 5 com desvio padrão 4, o esperado é que, o valor da divergência seja maior do que o exemplo da Figura 12, como nota-se na Figura 13 (MAKLIN, 2019).

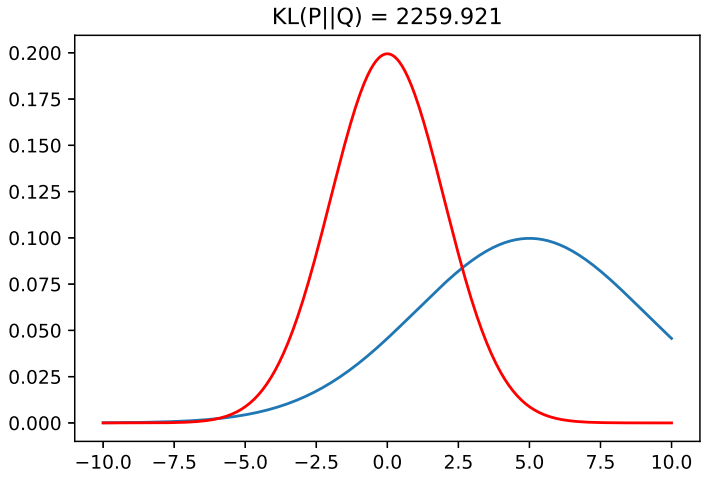
É importante notar que, a divergência de KL não é simétrica, como visualiza-se na Figura 14 (MAKLIN, 2019).

Figura 13 – Exemplo das 2 distribuições com média em 0 e 5, desvio padrão 2 e 4.



Fonte: Maklin (2019).

Figura 14 – Exemplo da Figura 13 com P e Q invertidos

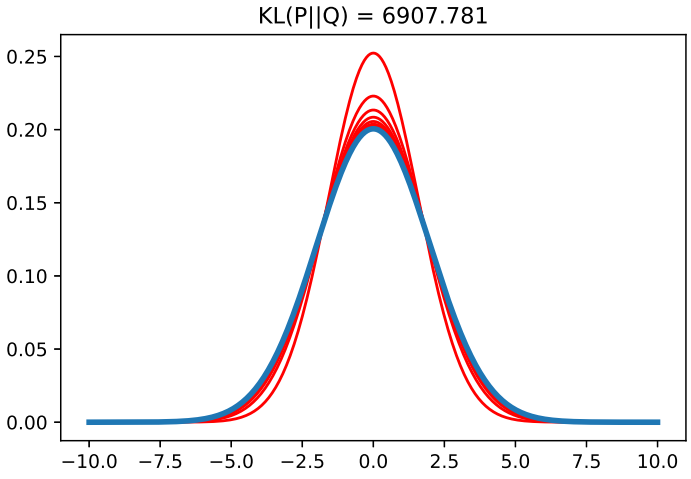


Fonte: Maklin (2019).

Para um outro exemplo foi criado uma distribuição conhecida com média centrada em 0 e variância 2. E então, foi criado uma outra distribuição com parâmetros aleatórios. Portanto, para usar o gradiente descendente especifica-se alguns hiper-parâmetros como *learning rate* (taxa de aprendizado) e *epochs* (épocas). Para este exemplo foi definido como 0.001 e 100. E então é realizado todo o cálculo anterior da divergência de KL e então ilustrado em um gráfico. Na Figura 15 abaixo, observa-se a divergência de KL em diferentes instantes de tempo da minimização (MAKLIN, 2019).

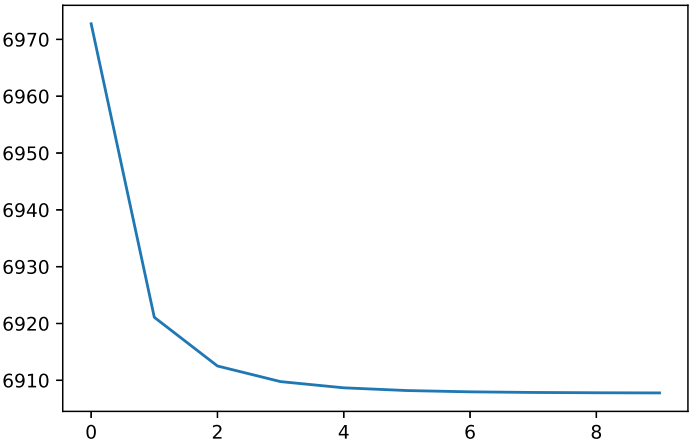
O parâmetro restante para ser selecionado é a variância σ_i da Gaussiana centrada em cada ponto x_i de alta dimensionalidade. Portanto, não é como se tivesse um único valor para σ_i que otimiza todos os pontos do conjunto de dados, pois é provável que a densidade dos

Figura 15 – Instantes da divergências de KL



Fonte: Maklin (2019).

Figura 16 – Minimização da divergência de KL



Fonte: Maklin (2019).

dados varie. Em regiões densas, valores pequenos de σ_i costumam ser mais apropriados do que em regiões dispersas. Entretanto, qualquer outro valor particular de σ_i induz a distribuição probabilística P_i sobre todos os outros pontos. Esta distribuição tem uma entropia que cresce conforme a σ_i cresce. Todavia, SNE realiza uma busca binária para o valor de σ_i, que produz P_i, com uma perplexidade fixa especificada pelo usuário. A Equação 5 da perplexidade é definida como:

$$Perp(P_i) = 2^{H(P_i)} \tag{5}$$

em que $H(P_i)$ é a entropia de Shannon de P_i medida em *bits*. Esta entropia de Shannon é dada

pela Equação 6 (MAATEN; HINTON, G., 2008):

$$H(P_i) = -\sum_j p_{j|i} log_2 p_{j|i} \tag{6}$$

Esta perplexidade na Equação 5 pode ser interpretada como uma medida *smooth* efetiva do número de vizinhos. O desempenho do SNE é robusto o suficiente para mudar a perplexidade e com valores em torno de 5 a 50 (MAATEN; HINTON, G., 2008).

Portanto, a minimização da função de custo da Equação 4 é realizado usando o método de gradiente descendente. O gradiente é dado pela Equação 7 da seguinte forma:

$$\frac{\delta C}{\delta y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j) \tag{7}$$

Fisicamente, o gradiente descendente pode ser interpretado como a força resultante criado por um conjunto de vetores entre o ponto y_i e todos os outros pontos y_j. Todos estes vetores exercem uma força ao longo da direção $(y_i - y_j)$. Estes vetores entre y_i e y_j repele ou atrai os pontos, dependendo se pequenoa ou grande a distância entre os dois pontos no mapa para representar a similaridade entre o par de pontos altamente dimensionais. Esta força exercida por um vetor entre y_i e y_j é proporcional ao comprimento e também à inflexibilidade, no qual é a incompatibilidade dada por $(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})$ entre as similaridades do par de pontos e dos pontos mapeados (MAATEN; HINTON, G., 2008).

O gradiente descendente é inicializado amostrando os pontos dos mapa aleatoriamente por uma Gaussiana isotrópica com uma variância pequena centrada em torno da origem. Em sequência, para agilizar a otimização e evitar um local mínimo improdutivo, um termo de momento relativamente grande é adicionado ao gradiente. Em outras palavras, o gradiente atual é adicionado a uma soma exponencial decadente do gradiente anterior, para determinar a mudança nas coordenadas do mapa de pontos para cada iteração de busca do gradiente. Em sequência, tem-se a Equação 8 do gradiente atualizado com o termo de momento:

$$\gamma^{(t)} = \gamma^{(t-1)} + \eta\frac{\delta C}{\delta \gamma} + \alpha(t)(\gamma^{(t-1)} - \gamma^{(t-2)}) \tag{8}$$

em que $\gamma^{(t)}$ indica a solução na iteração (t), η indica a taxa de aprendizado e $\alpha(t)$ representa o momento da iteração t. Nos estágios iniciais da otimização, é adicionado um ruído Gaussiano nos pontos mapeados após cada iteração, no entanto, reduzindo gradativamente a variância do ruído, reproduz um tipo de *simulated annealing* (um tipo de meta-heurística para otimização)

que ajuda a escapar de mínimos locais na função de custo (MAATEN; HINTON, G., 2008).

2.3.3 PROJEÇÃO DA VIZINHANÇA ESTOCÁSTICA DISTRIBUIÇÃO-t

Projeção da vizinhança estocástica distribuição-t (t-SNE) é um método de RD voltado para a visualização de dados e uma variação do SNE, visto anteriormente na seção Seção 2.3.2.2. Esta técnica de redução foi proposta por Maaten e Geofftrey Hinton (2008) para diminuir problemas existentes no SNE, como o problema de aglomeração (*crowding problem*) e o problema de construir uma boa visualização impedido por uma função de custo de difícil otimização (MAATEN; HINTON, G., 2008).

Com o intuito de melhorar a visualização, a função de custo do t-SNE distingue-se da função de custo do SNE de duas maneiras. Primeiro, utiliza-se uma versão simétrica da função de custo com gradiente simples introduzida brevemente por Cook *et al.* (2007). Segundo, utiliza-se a *Student-t distribution* (distribuição t de *Student*) no lugar da distribuição Gaussiana para calcular a similaridade entre dois pontos no espaço de baixa dimensão. Entretanto, este algoritmo estabelece uma distribuição de cauda pesada (*heavy-tailed distribution*) no espaço de baixa dimensão para lidar com os problemas existentes de aglomeração e otimização da função de custo do SNE (MAATEN; HINTON, G., 2008).

2.3.3.1 SNE SIMÉTRICO

Como uma alternativa para minimizar a soma da divergência de KL entre as probabilidades condicionais $p_{j|i}$ e $q_{j|i}$, também é possível minimizar uma única divergência de KL entre a distribuição probabilística conjunta P no espaço de alta dimensão e a distribuição probabilística conjunta Q no espaço de baixa dimensão, conforme a Equação 9 (MAATEN; HINTON, G., 2008).

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{9}$$

E novamente, p_{ii} e q_{ii} são definidos como zero. No entanto, esse tipo de SNE é conhecido como SNE simétrico, pois segue a propriedade em que $p_{ij} = p_{ji}$ e $q_{ij} = q_{ji}$, para todo i e j. No SNE simétrico, a similaridade entre os pares q_{ij} no mapa de baixa dimensão é dada por:

$$q_{ij} = \frac{\exp\left(-\left\|y_i - y_j\right\|^2\right)}{\sum_{k \neq l} \exp\left(-\left\|y_k - y_l\right\|^2\right)} \tag{10}$$

Da mesma forma, é possível definir a similaridade entre os pares p_{ij} no espaço de alta dimensão como:

$$p_{ij} = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\left\|x_k - x_l\right\|^2 / 2\sigma^2)} \tag{11}$$

Pela Equação 11, torna-se um problema quando o ponto x_i da alta dimensão é um ponto fora da curva , i.e., todas distância do par $\left\|x_i - x_j\right\|^2$ é distante para x_i. Para tal ponto fora da curva, o valor de p_{ij} é extremamente pequeno para j, portanto, para a localização do y_i no mapa de baixa dimensão tem um efeito pequeno na função de custo. Como resultado, a posição do ponto no mapa não é definido adequadamente pela posição de outros pontos no mapa. Este problema foi contornado definindo a probabilidade conjunta, p_{ij}, no espaço de alta dimensão para ser a probabilidade condicional simetrizada como $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$. Isto garante que $\sum_j p_{ij} > \frac{1}{2n}$ para todos os pontos x_i. Consequentemente cada ponto x_i produz uma contribuição significante para a função de custo. E no mapa de baixa dimensão, o SNE simétrico simplifica o uso da Equação 10. Entretanto, a principal vantagem da versão simétrica do SNE é a simplificação da forma do gradiente, tornando-o rápido para se calcular. O gradiente do SNE simétrico é similar ao do SNE assimétrico e é dado pela Equação 12 (MAATEN; HINTON, G., 2008).

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \tag{12}$$

2.3.3.2 PROBLEMA DE AGLOMERAÇÃO

Considere um conjunto de pontos que está sobre um objeto curvado 2D que é aproximadamente linear em pequena escala e contido no espaço de alta dimensão. É possível modelar e igualar pequenas distâncias entre pontos relativamente bem em um mapa 2D, em que é ilustrado no conjunto de dados artificial como "*Swiss Roll*". Agora, supondo que o objeto tem dez dimensões intrínsecas e que está contido em um espaço de alta dimensão maior. Existem diversas razões do porquê igualar distâncias em mapa 2D não pode-se modelar com exatidão as distâncias entre pontos de objeto com dez dimensões. Por exemplo, em dez dimensões, é possível ter 11 pontos que são mutualmente equidistante e não há como modelar fielmente em mapa 2D. Um problema relatado é a distribuição muito diferente de distâncias iguais em dois

espaços (MAATEN; HINTON, G., 2008).

Por exemplo, o volume da esfera centrado no ponto i mede um tamanho r^m em que r é o raio da esfera e m a dimensionalidade da esfera. Então, se os pontos são distribuídos aproximadamente uniformemente na região em torno de i em um objeto de dez dimensões e tentar modelar as distâncias a partir de i para os outros pontos no mapa 2D, ocorrerá o *crowding problem* (problema de aglomeração). A área do mapa 2D disponível para adaptar os pontos relativamente distantes não será grande o suficiente em comparação com a área disponível para acomodar os pontos próximos. Por isso, para modelar distâncias pequenas precisamente no mapa, a maioria dos pontos que estão moderadamente distante do ponto i terá de ser posicionada para longe no mapa 2D (MAATEN; HINTON, G., 2008).

2.3.3.3 DISPARIDADE DE CAUDAS PODEM SER COMPENSADO POR DISPARIDADE DE LAÇOS DIMENSIONAIS

No espaço de alta dimensão converteu-se distâncias em probabilidades usando distribuição Gaussiana. E no mapa de baixa dimensão utiliza-se uma distribuição probabilística com caudas maiores do que a Gaussiana para converter distâncias em probabilidades. Isto permite uma distância moderada no espaço de alta dimensionalidade para modelar precisamente distâncias maiores no mapa, consequentemente, elimina-se forças atrativas indesejadas entre pontos do mapa que representa pontos relativamente dissimilares (MAATEN; HINTON, G., 2008).

No t-SNE, utiliza-se a distribuição-t com um grau de liberdade como *heavy-tailed distribution* no mapa de baixa dimensão. Usando esta distribuição, o conjunto de probabilidades q_{ij} é definido na Equação 13.

$$q_{ij} = \frac{(1 + \left\| y_i - y_j \right\|^2)^{-1}}{\sum_{k \neq l} (1 + \| y_k - y_l \|^2)^{-1}} \tag{13}$$

Utiliza-se a distribuição-t com um único grau de liberdade, pois apresenta uma propriedade particularmente interessante: $(1 + \left| y_i - y_j \right|^2)^{-1}$ representa a inversa da ordem quadrática para o par de pontos $\left| y_i - y_j \right|$, gerando uma ampla distância no mapa de baixa dimensão. Isso faz com que as representações dos mapas do conjunto de probabilidades sejam quase invariantes a mudanças nas escalas de distância entre os pontos. Além disso, significa que os pontos de grupos grandes, mesmo estando distantes, interagem da mesma maneira que pontos individuais. Portanto, a otimização é igualmente efetiva em todas as escalas adequadas (MAATEN; HINTON, G.,

2008).

A escolha da distribuição t se justifica por sua estreita relação com a distribuição Gaussiana. A propriedade computacionalmente conveniente desta distribuição é dada pela rapidez do cálculo da densidade de pontos abaixo da distribuição-t, em comparação com a Gaussiana. Assim, evita-se o uso de uma exponencial, apesar da distribuição-t ser equivalente a mistura infinita de Gaussianas com diferentes variâncias (MAATEN; HINTON, G., 2008).

O gradiente da divergência de KL entre P e a distribuição-t, com base no conjunto de probabilidades Q, é dado pela Equação 14.

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \left\| y_i - y_j \right\|^2)^{-1} \tag{14}$$

2.3.3.4 MÉTODO DE OTIMIZAÇÃO PARA t-SNE

O método de otimização proposto para o t-SNE é um procedimento de gradiente descendente relativamente simples para a função de custo. Este procedimento utiliza um termo de momento para reduzir o número de iterações necessárias e funciona bem quando o termo de momento é pequeno, até que os pontos do mapa se tornem relativamente organizados. Um pseudocódigo para este algoritmo é apresentado no Algoritmo 1. Este algoritmo pode ser acelerado utilizando um esquema adaptativo de taxa de aprendizagem descrito por Jacobs (1988), no qual a taxa de aprendizagem aumenta gradualmente à medida que o gradiente se estabiliza. No entanto, este algoritmo produz visualizações melhores do que outras técnicas não paramétricas de RD, e os resultados podem ser aprimorados utilizando algumas estratégias. Uma dessas estratégias é chamada de *early compression*. Essa estratégia faz com que os pontos do mapa fiquem próximos no início da otimização, facilitando o movimento dos agrupamentos quando as distâncias entre os pontos do mapa são pequenas. Consequentemente, isso possibilita a exploração de uma possível organização global dos dados (MAATEN; HINTON, G., 2008).

A estratégia *early compression* é implementada com uma penalidade adicional *L2-penalty*, proporcional à soma das distâncias quadráticas dos pontos do mapa a partir da origem (MAATEN; HINTON, G., 2008).

Outra maneira de aprimorar a otimização é conhecida como *early exaggeration*. Essa estratégia multiplica todos os p_{ij}*'s*, por exemplo, por 4 no estágio inicial da otimização. Isso significa que praticamente todos os q_{ij}*'s*, cujos valores são somados até 1, são pequenos demais para modelar seus correspondentes q_{ij}*'s*. Como resultado, a otimização é incentivada a destacar

Data: data set $X = \{x_1,x_2,...,x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$
Result: low-dimensional data representation $\gamma^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
compute paiwise affinities $p_{j|i}$ with perlexity $Perp$ (using equation 2)
set $p_{ji} = \frac{p_{j|i}+p_{i|j}}{2n}$
sample initial solution $\gamma^{(0)} = \{y_1,y_2,...,y_n\}$ from $N(0,10^{-4}I)$
for $t = 1{:}T$ **do**
    compute low-dimensional affinities q_{ij} (using equation 13)
    compute gradient $\frac{\delta C}{\delta \gamma}$ (using equation 14)
    set $\gamma^{(t)} = \gamma^{(t-1)} + \eta\frac{\delta C}{\delta \gamma} + \alpha(t)(\gamma^{(t-1)} - \gamma^{(t-2)})$
end for
end

algorithm 1 – Versão simplificada de t-Distributed Stochastic Neighbor Embedding

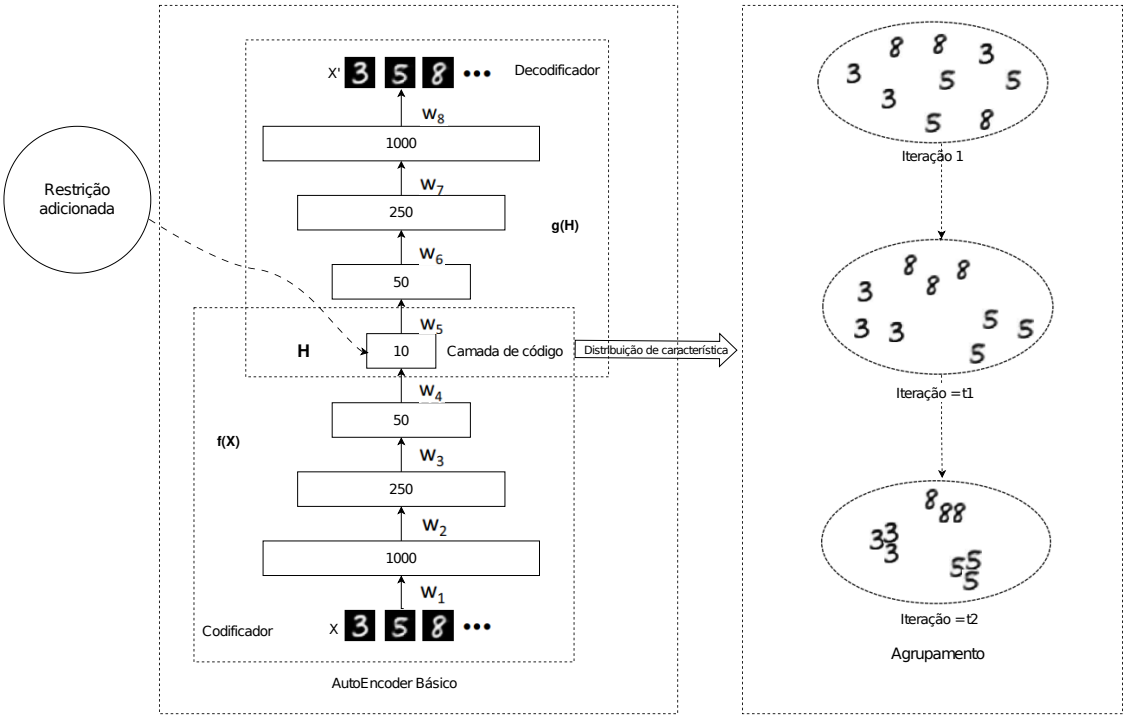
a modelagem de valores grandes de p_{ij}*'s* de forma precisa, gerando valores grandes de q_{ij}*'s*. Consequentemente, os agrupamentos tendem a ser densos e amplamente separados no mapa. Isso cria espaços relativamente vazios no mapa, facilitando o movimento dos grupos em torno de outros grupos para encontrar uma organização global adequada (MAATEN; HINTON, G., 2008).

2.3.4 AUTOENCODER

Os Autoencoders, introduzidos por Rumelhart, McClelland e PDP Research Group (1986), têm o objetivo de reconstruir uma saída com o menor erro possível dadas as entradas observadas. Seu princípio é treinar as redes de forma não supervisionada como uma representação informativa dos dados, que pode ser utilizada para diversas aplicações, como por exemplo, agrupamento (BANK; KOENIGSTEIN; GIRYES, 2020). Entre os Autoencoders, existem diversos tipos, como o Autoencoder regularizado, o Autoencoder esparso, o Autoencoder eliminador de ruído (*Denoising*), o Autoencoder variacional (VAE), entre outros.

Para um melhor entendimento dos Autoencoders, é necessário compreender sua arquitetura típica. Composto por três componentes principais, o Autoencoder possui um codificador, uma representação de características latentes e um decodificador. O codificador e o decodificador são funções simples, enquanto a representação latente das características é entendida como as características importantes dos dados. Na maioria das arquiteturas, o codificador e o decodificador são redes neurais, em que a saída do codificador é a representação latente, ou seja, informações significativas dos dados, e a entrada do decodificador é a representação latente (BANK; KOENIGSTEIN; GIRYES, 2020). Abaixo, na Figura 17, há uma ilustração da arquitetura de um Autoencoder.

Figura 17 – Arquitetura base de um Autoencoder com modificação



Fonte: Adaptado de Song *et al.* (2013).

2.3.4.1 USO DO AUTOENCODER PARA AGRUPAMENTO E LIMITAÇÕES

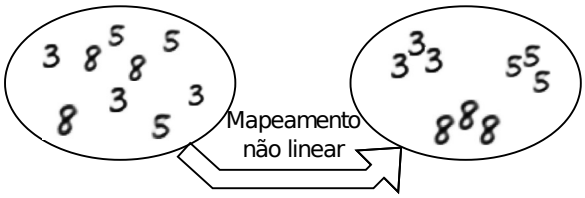
Agrupamento é um problema não supervisionado em que o objetivo é separar os dados em grupos de modo que as amostras em cada grupo sejam similares às outras amostras do grupo e diferentes das amostras de outros grupos. A maioria dos algoritmos de agrupamento são sensíveis a dimensionalidade de um dado e sofre do problema conhecido como a maldição da dimensionalidade (RUMELHART; MCCLELLAND; PDP RESEARCH GROUP, 1986).

Assumindo que os dados possuem alguma representação latente de baixa dimensionalidade pode-se utilizar o Autoencoder para calcular tal representação dos dados. Primeiro treina-se o Autoencoder de acordo com sua estrutura original. Porém, a etapa de decodificação é deixada de lado de forma similar realizada para a classificação. Assim, a representação latente (saída do codificador) é mantida e serve como entrada para qualquer algoritmo de agrupamento, por exemplo, K-means (RUMELHART; MCCLELLAND; PDP RESEARCH GROUP, 1986).

A principal desvantagem do uso do *vanilla encoder* é a etapa da camada de código da Figura 17 que é treinada exclusivamente para reconstrução e não para aplicações de agrupamentos. Em Song *et al.* (2013) é proposto uma forma similar ao algoritmo do *K-means* para agrupamento, mas as incorporações são retreinadas a cada iteração. No treinamento, um argumento foi adicionado na função de perda do Autoencoder que penaliza a distância entre os dados incorporados e

os centros dos grupos (BANK; KOENIGSTEIN; GIRYES, 2020). Na Figura 18, é apresentada uma ilustração de como um mapeamento não linear seria representado.

Figura 18 – Ilustração do mapeamento não linear dos dados



Fonte: Adaptado de Song *et al.* (2013).

2.4 DISTORÇÕES EM ALGORITMOS DE REDUÇÃO DE DIMENSIONALIDADE

Embora a proposta do uso de t-SNE seja representar fielmente estruturas globais e locais de dados de alta dimensionalidade em um mapa de duas ou três dimensões, há evidências de que ela falha nesse quesito (CHARI; PACHTER, 2022). No estudo realizado aplicando t-SNE para análise de expressões genômicas de célula única em Chari e Pachter (2022) *The Specious Art of Single-Cell Genomics*, encontraram-se distorções nas estruturas globais e locais das propriedades biológicas.

Para entender os efeitos de técnicas de redução bidimensional em dados sequenciais de RNA de célula única, em Chari e Pachter (2022) iniciou-se examinando a extensão da distorção em células incorporadas. Um resultado chave em transformações lineares de baixa distorção de pontos no espaço Euclidiano é o trabalho de Johnson e Lindenstrauss (1984), que fornece uma condição de suficiência para redução de dimensionalidade com baixa distorção. No artigo, é mencionado que a preservação de um par de distância de m pontos com um fator de distorção de $1 \pm \epsilon$ pode ser realizada na ordem de $\log(m)/\epsilon^2$ dimensões. Embora demonstre que a redução de dimensionalidade possa preservar a estrutura de dados de alta dimensionalidade, o número de dimensões requeridas ultrapassa duas ou três dimensões. Por exemplo, utilizando a constante de distorção de Dasgupta e Gupta (2003), a distorção entre os pares de pontos dentro de 20% de uma base de dados de 10.000 pontos pode chegar a pelo menos 1.842 dimensões com a redução (CHARI; PACHTER, 2022).

Para compreender melhor as limitações das técnicas de redução de dimensionalidade, considera-se um caso particularmente desafiador que envolve a transformação de pontos equidistantes. É impossível transformar $n + 1$ pontos equidistantes em um espaço $\mathbf{R}^k$ para $k \leq n$ sem distorção. Mesmo que se diminua a limitação de equidistância para pontos próximos, onde

quaisquer três pontos em $\mathbf{R}^n$ têm um par a uma distância unitária, só é possível acomodar sete pontos em $\mathbf{R}^2$ ou dez em $\mathbf{R}^3$ (CHARI; PACHTER, 2022) (BALKO *et al.*, 2020). Além disso, a distorção de pontos equidistantes pode ser particularmente mais aguda com o PCA, muito utilizado em dados pré-condicionados. O PCA de pontos equidistantes é equivalente à aplicação de uma projeção randômica, e como resultado, os pontos projetados mostram diversas miragens da estrutura em duas dimensões (CHARI; PACHTER, 2022).

2.5 MÉTRICAS DE AVALIAÇÃO DE ESTRUTURA

Na abordagem de avaliação de algoritmos de redução de dimensionalidade, existem métodos disponíveis para avaliar a conservação da estrutura global e das estruturas locais. Dentre esses métodos, segundo Gracia *et al.* (2014) em uma listagem de métodos, são mencionados o Diagrama de Sheppard, a Medição de Estresse Kruskal, o Estresse de Sammon, a Variância Residual, a Medição Global, o Erro Relativo, entre outros, que são do tipo que medem como a transformação do espaço de alta dimensão para o espaço de baixa dimensão afeta a estrutura global. Já para a medição da conservação das estruturas locais, são citados métodos como o *Spearman's Rho*, o Produto topológico, a Função topológica, a Medição de König, entre outros listados por Gracia *et al.* (2014).

3 MATERIAIS E MÉTODOS

Neste capítulo, abordaram-se os materiais e métodos utilizados neste trabalho, que envolveram a aplicação de algoritmos de redução de dimensionalidade em dados sintéticos gerados manualmente, bem como em dados reais. O objetivo foi reduzir a dimensionalidade desses dados para níveis observáveis e avaliar a redução por meio do Estresse de Kruskal, possibilitando uma comparação dos resultados. Um critério para a seleção das bases de dados foi que elas possuíssem estruturas internas, seja no formato de objetos ou que houvesse uma propriedade que permitisse relacionar a visualização obtida após a redução de dimensionalidade com os dados. Além disso, um critério importante a ser considerado foi a utilização dessas bases em pesquisas no âmbito de *Manifold Learning*. As técnicas de algoritmos de redução de dimensionalidade foram escolhidas por serem utilizadas em pesquisas e trabalhos e a métrica de Kruskal por ter relação com similaridades e dissimilaridades dos pontos, retornando uma porcentagem em termos de distorção.

Portanto, o processo foi organizado da seguinte forma:

1. Foi selecionada uma base de dados para ser aplicada aos algoritmos de redução de dimensionalidade, escolhendo-se bases que possuíssem estruturas visualizáveis, como a base do Rolo Suíço e a Curva S, ou que apresentassem alguma relação;
2. Foram selecionados algoritmos de redução de dimensionalidade disponibilizados pela biblioteca *scikit-learn*, que foram de fácil utilização;
3. As bases de dados foram aplicadas aos algoritmos de redução de dimensionalidade;
4. Foi aplicada a métrica de Kruskal para obter resultados quantitativos;
5. Por fim, foram geradas imagens para a comparação visual dos algoritmos de redução de dimensionalidade. Para melhorar a visualização, os gráficos das saídas foram normalizados para valores entre 0 e 1;

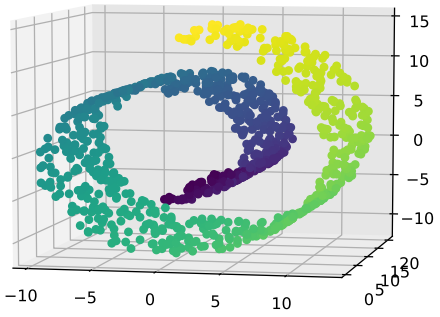
O desenvolvimento foi realizado em um notebook virtual disponibilizado pela *Google* conhecido como *Google Colab* em que a linguagem de programação escolhida para o desenvolvimento foi Python.

3.1 BASES DE DADOS UTILIZADAS

3.1.1 BASE DE DADOS SINTÉTICA

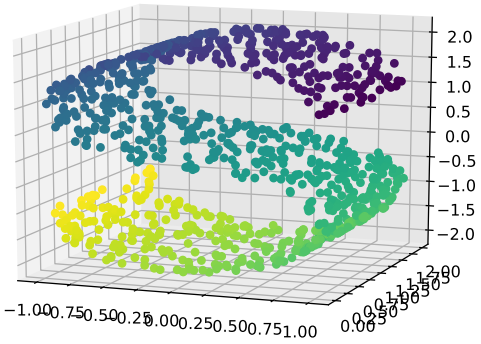
Entre as bases sintéticas, gerou-se as bases por meio de uso de funções disponibilizadas pela biblioteca da *scikit-learn* (PEDREGOSA *et al.*, 2011). Utilizou-se o Rolo Suíço mostrado na Figura 19 e a Curva S mostrada na Figura 20, que são utilizados para estudos de *Manifold* e é possível entender como propriedades intrínsecas existentes dentro de uma base de dados. Nos dois casos existe uma propriedade na base que demonstra o formato dos dados. Outra base gerada foi a *Hello* da Figura 21, um exemplo de VanderPlas (2017).

Figura 19 – Visualização em 3D da base Rolo Suíço



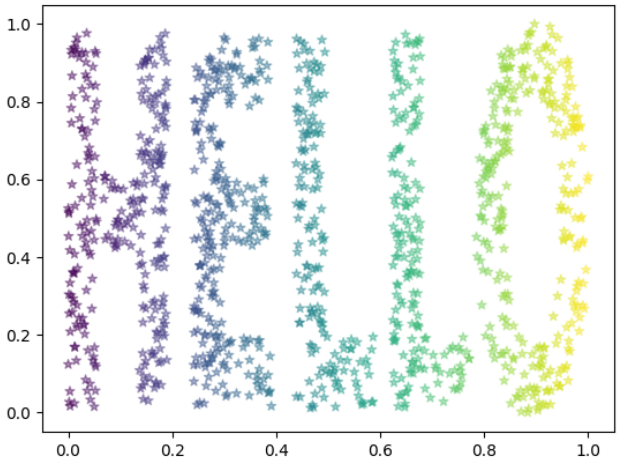
Fonte: Autoria própria.

Figura 20 – Visualização em 3D da base Curva S



Fonte: Autoria própria.

Figura 21 – Visualização em 2D da base Hello



Fonte: Autoria própria.

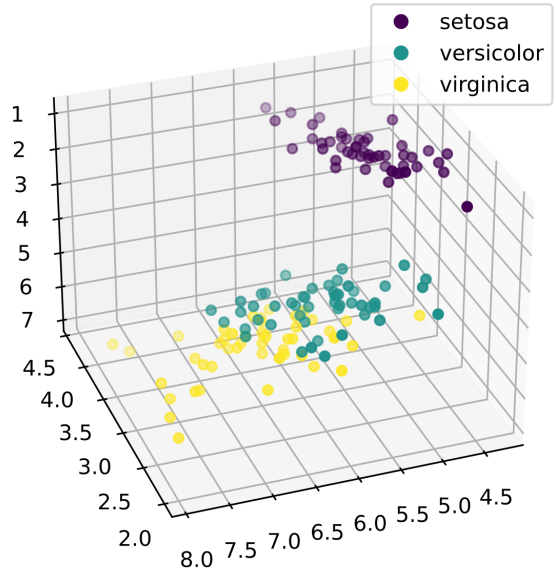
3.1.2 BASE DE DADOS REAIS

Pôde-se entender uma base de dados reais como um conjunto de medições por meio de sensores, imagens e textos que formaram um conjunto de informações que puderam ser tratadas, analisadas e utilizadas, por exemplo, para treinar modelos de aprendizado de máquina. Por exemplo, foi utilizada a base MNIST da Figura 22, que consistiu em imagens de dígitos formados por 28x28 pixels, totalizando 784 pixels como características da imagem. Também foi utilizada a base Iris, disponibilizada pela biblioteca *scikit-learn* (PEDREGOSA *et al.*, 2011), com 3 características da planta Íris.

Figura 22 – Base de dados de imagens de dígitos de 784 dimensões

Fonte: Autoria própria.

Figura 23 – Base de dados Iris em projeção 3D



Fonte: Autoria própria.

3.2 ALGORITMOS DE REDUÇÃO DE DIMENSIONALIDADE

Os algoritmos de redução de dimensionalidade utilizados foram os disponibilizados pela biblioteca scikit-learn (PEDREGOSA *et al.*, 2011): Isomap, Projeção linear local padrão (LLE), MDS, Árvores Randômicas (Random Trees), t-SNE, PCA e Autoencoder importado da

interface de programação de aplicação conhecida como Keras (CHOLLET *et al.*, 2015), que faz parte do TensorFlow (MARTIN ABADI *et al.*, 2015). O Autoencoder restringiu-se a saída do codificador a uma dimensão projetável 2D ou 3D.

Para o desenvolvimento deste trabalho, foram consideradas algumas configurações para o uso dos algoritmos de redução de dimensionalidade. Para todos os algoritmos, o número de componentes ou eixos de projeção foi configurado como dois, a fim de obter a visualização em uma projeção 2D. Especificamente para Isomap e LLE padrão, o parâmetro *n_neighbors* foi configurado como 30 pontos de vizinhança, o qual foi calculado como 1% do total de pontos gerados para cada base, uma vez que esses algoritmos trabalham com a busca dos vizinhos mais próximos. Para MDS, o valor de *n_init*, responsável por determinar quantas vezes o algoritmo SMACOF (*Scaling by Majorizing a Complicated Function*) será executado, foi inicializado para ter apenas uma execução. Segundo Pedregosa *et al.* (2011), o SMACOF é utilizado para minimizar a função objetivo, garantindo sua convergência monótona e sendo considerado uma técnica mais poderosa do que, por exemplo, o gradiente descendente. Para Árvores Randômicas, utilizou-se os mesmos parâmetros do exemplo disponibilizado pela Pedregosa *et al.* (2011) devido à difícil configuração do algoritmo. Por fim, para t-SNE, *n_iter*, que representa o número de iterações do cálculo interno do algoritmo, foi configurado para executar 750 iterações, com um critério de parada de 150 iterações em *n_iter_without_progress*, caso não haja mudança nos valores calculados. Também foi inicializada a variável *random_state* com o valor 1, com o intuito de obter resultados diferentes para a visualização, apesar do t-SNE possuir propriedade estocástica na implementação, retornando visualizações distintas.

3.3 MEDIÇÃO DE ESTRESSE DE KRUSKAL

Dentre as métricas que medem estruturas globais, existe a medição de Estresse de Kruskal, que é uma técnica de escala multidimensional cujo conceito é similar à técnica de *Shepard*. No entanto, a técnica *Shepard* busca encontrar uma configuração espacial que melhor se ajuste às medidas de dissimilaridade ou similaridade iniciais (KRUSKAL, 1964), sendo o único requisito que a relação seja monótona entre essas medidas e as distâncias na configuração (SHEPARD, 1957). Por outro lado, o Estresse de Kruskal adota uma abordagem diferente, pois seu foco principal não está apenas na preservação da relação monotônica. O Estresse de Kruskal introduz uma medida quantitativa chamada “estresse” para avaliar o ajuste da configuração às medidas de dissimilaridade ou similaridade. Essa medida leva em consideração tanto a relação

monotônica quanto a variância residual (KRUSKAL, 1964).

No trabalho de Kruskal (1964), foi apresentada uma técnica de escala multidimensional similar à técnica de *Shepard*, que surgiu a partir de tentativas de melhorar e aperfeiçoar suas ideias. A técnica adota a relação monotônica entre dissimilaridade e distância como objetivo principal. Assim, como um acréscimo à medição de Estresse de Kruskal, é fornecida uma medida quantitativa natural de não monotonicidade. Para cada configuração dada, realiza-se uma regressão monotônica da distância em relação à dissimilaridade, e a variância residual é utilizada de maneira adequada e normalizada como uma medida quantitativa chamada Estresse (KRUSKAL, 1964).

O Estresse de Kruskal pode ser classificado de acordo com os resultados obtidos no cálculo, conforme demonstrado na Tabela 3.

Tabela 3 – Tabela de qualidade de Estresse de Kruskal

Estresse	Qualidade
20%	Muito ruim
10%	Ruim
5%	Bom
$2\frac{1}{2}\%$	Excelente
0%	“Perfeito"

Fonte: Adaptado de Kruskal (1964).

Isto é, por “Perfeito”, entende-se que existe uma relação perfeita de monotonicidade entre a dissimilaridade dos dados e a distância. A equação do Estresse de Kruskal pode ser dada da seguinte forma:

$$Estresse = S = \sqrt{\frac{S*}{T*}} = \sqrt{\frac{\sum_{i<j}^{n}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j}^{n} d_{ij}^2}} \tag{15}$$

em que d_{ij} é distância entre os pontos i e j na dimensão original e $\hat{d_{ij}}$ é a distância entre os pontos i e j na dimensão projetada. Dado a equação 15, tem-se as melhores configurações de projeção, sendo as que retornam o valor mínimo possível para o estresse (KRUSKAL, 1964).

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 RESULTADOS COM BASES SINTÉTICAS

Primeiramente, iniciou-se a aplicação dos algoritmos de redução à base “Hello” e passou-se pelo processo de transformação. No entanto, por tratar-se de uma transformação de $R^2 \rightarrow R^2$, foi possível apenas observar distorções sem que ocorresse efetivamente a redução de dimensionalidade para análise dos resultados. Na Figura 24, foi possível notar a distorção da informação intrínseca dos dados presentes na base de acordo com o algoritmo.

Neste ponto, tornou-se notável que os algoritmos com propriedades de transformação linear, como PCA e MDS, apresentaram resultados melhores sem alterar sua estrutura. Portanto, os algoritmos com propriedade de transformação não linear, como Isomap, LLE padrão, Árvores Randômicas e t-SNE, distorceram as saídas em comparação com a entrada.

Na Tabela 4, pôde-se observar o valor do Estresse de Kruskal calculado para cada algoritmo em ordem crescente. Embora a visualização do t-SNE apresentasse uma melhor preservação da estrutura em comparação, por exemplo, com as Árvores Randômicas e a projeção do LLE Padrão, em termos do valor quantitativo calculado pelo Estresse de Kruskal, obteve-se um valor pior. Com base nisso, observou-se que, por meio da visualização do t-SNE, havia a presença de pequenos espaços densos e espaços em branco, nos quais os pontos similares se tornaram ainda mais próximos e os dissimilares ainda mais distantes. No entanto, acreditava-se que o fator mais impactante a ser considerado para obter o pior valor calculado do Estresse de Kruskal era a natureza de agrupamento do algoritmo.

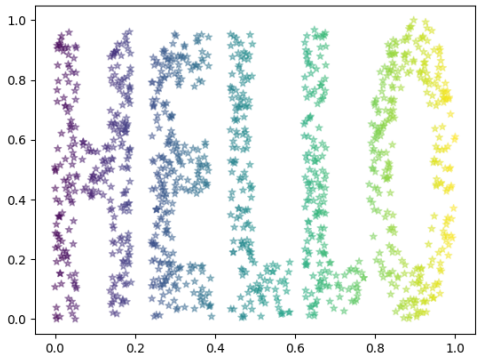
Tabela 4 – Tabela de Stress para a base Hello

Algoritmo	KruskalStress(%)
PCA	0.00
MDS	0.19
Isomap	17.25
LLE padrão	93.64
Árvores randômicas	706.12
t-SNE	5692.01

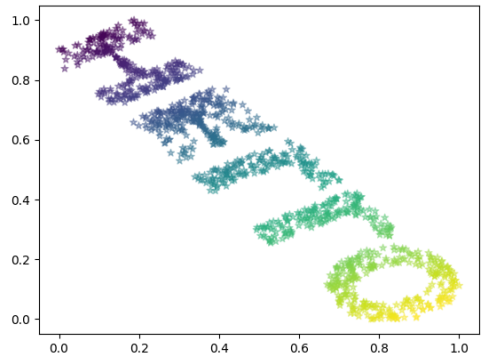
Fonte: Autoria própria.

Figura 24 – Projeção das transformação para a base sintética Hello

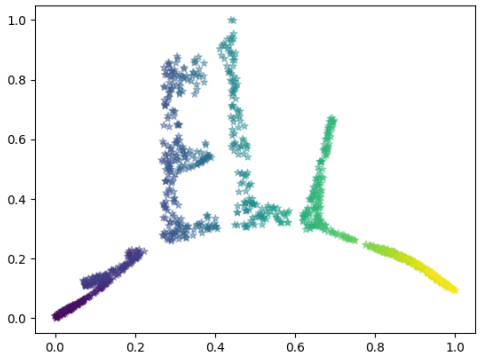
(a) projeção PCA



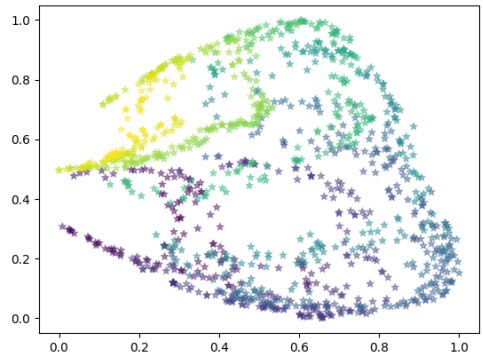
(b) projeção MDS



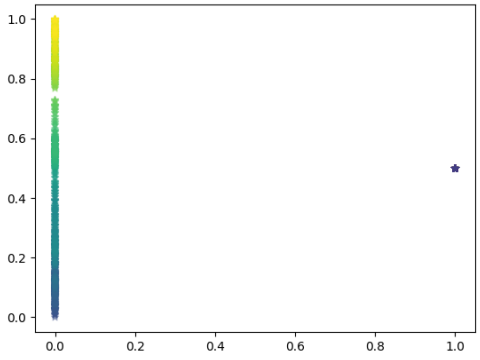
(c) projeção Isomap



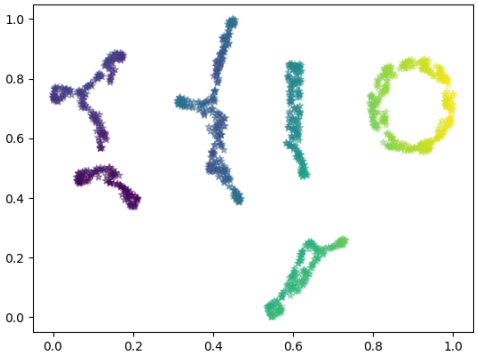
(d) projeção Árvores Randômicas



(e) projeção LLE Padrão



(f) projeção t-SNE



Fonte: Autoria própria.

A base da Curva S, mostrada na Figura 25, e a base do Rolo Suíço, na Figura 26, sendo uma conversão de $R^3 \rightarrow R^2$, apresentaram um comportamento semelhante tanto para o PCA quanto para o MDS.

Ao analisar as reduções para a base da Curva S, tanto o PCA quanto o MDS apresentaram projeções visualmente semelhantes, com poucas alterações. No entanto, ao observar os outros algoritmos, notou-se que o Isomap conseguiu preservar a distância entre as extremidades, como se estivesse "esticando" a projeção. Da mesma forma, o LLE padrão e o t-SNE apresentaram resultados semelhantes ao Isomap, mantendo as duas extremidades distantes. No entanto, ao verificar a Tabela 5, constatou-se que o t-SNE era a projeção com a maior distorção, segundo o valor do Estresse de Kruskal, apesar de estar visualmente próximo ao PCA e ao MDS. Acredita-se que a natureza de agrupamento do t-SNE tenha impactado consideravelmente na relação de distância entre os pontos, resultando no pior valor calculado para o Estresse de Kruskal, assim como na base Hello.

Tabela 5 – Tabela de Stress para a base da curva S

Algoritmo	KruskalStress(%)
MDS	11.75
PCA	13.99
Isomap	78.52
LLE padrão	98.51
Árvores randômicas	159.87
t-SNE	2241.90

Fonte: Autoria própria.

Para a base do Rolo Suíço na Figura 26, pôde-se verificar visualmente que o PCA e o MDS obtiveram melhores resultados na preservação da estrutura em espiral do rolo. Além disso, visualmente, tinha-se a impressão de que o Isomap e o t-SNE conservaram bem a estrutura, mas ao consultar a Tabela 6, constatou-se que apresentaram maior distorção quando o estresse foi calculado. No entanto, a redução utilizando Árvores Randômicas manteve um comportamento em espiral, enquanto o LLE padrão apresentou uma projeção semelhante à Curva S.

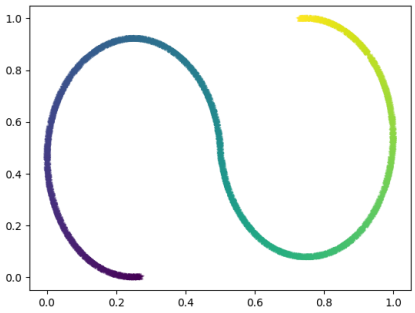
Tabela 6 – Tabela de Stress para a Base do rolo suíço

Algoritmo	KruskalStress(%)
MDS	24.12
PCA	26.13
Árvores randômicas	71.81
LLE padrão	99.80
Isomap	162.29
t-SNE	233.19

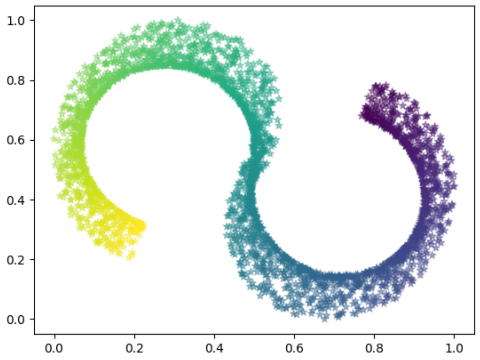
Fonte: Autoria própria.

Figura 25 – Projeção das transformação para a base sintética da Curva S

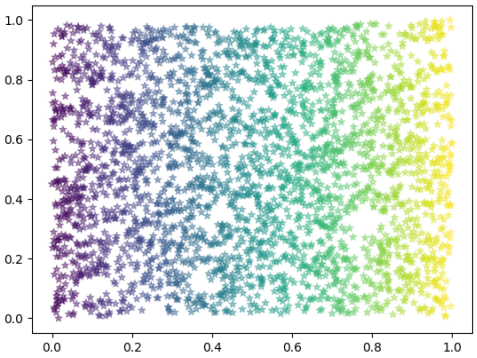
(a) projeção PCA



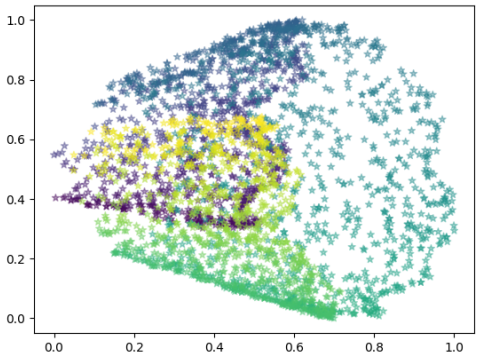
(b) projeção MDS



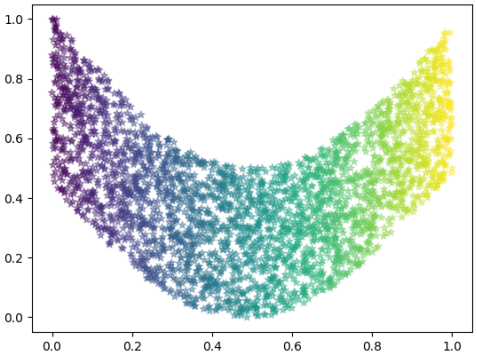
(c) projeção Isomap



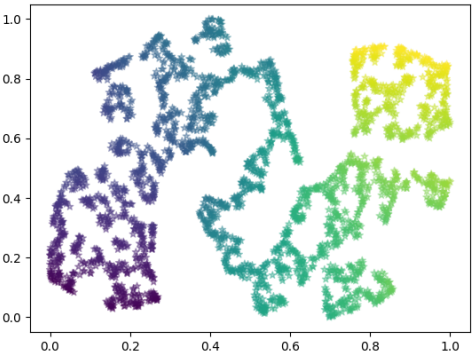
(d) projeção Árvores Randômicas



(e) projeção LLE Padrão



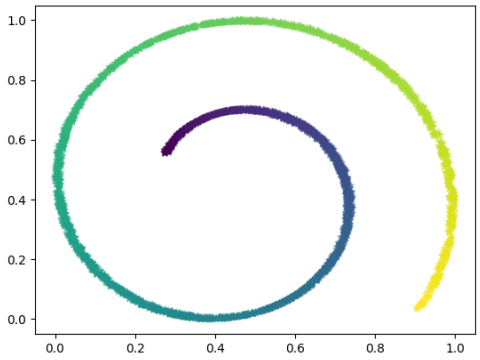
(f) projeção t-SNE



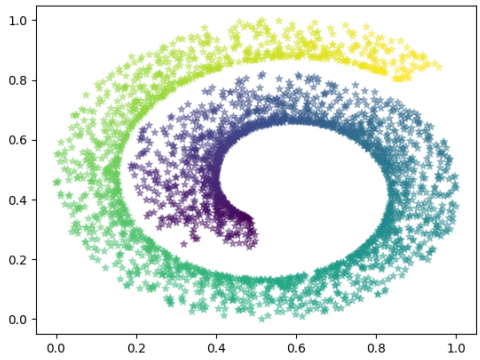
Fonte: Autoria própria.

Figura 26 – Projeção das transformação para a base sintética Rolo Suíço

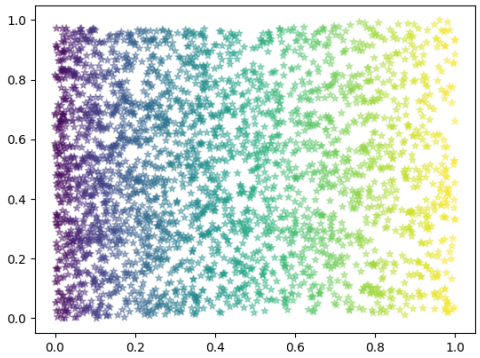
(a) projeção PCA



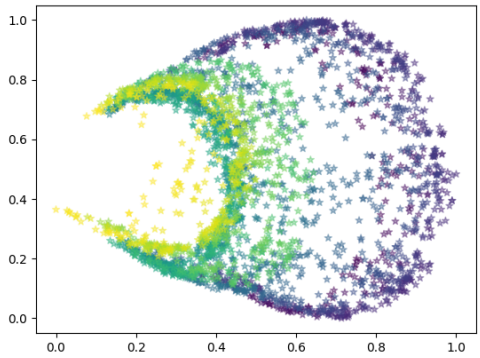
(b) projeção MDS



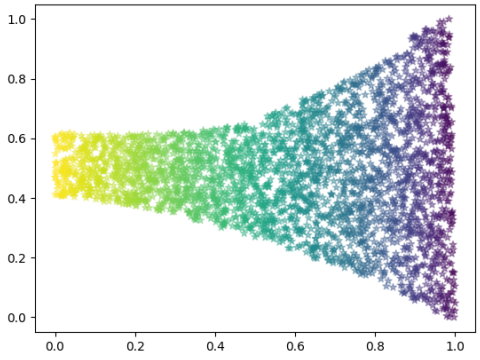
(c) projeção Isomap



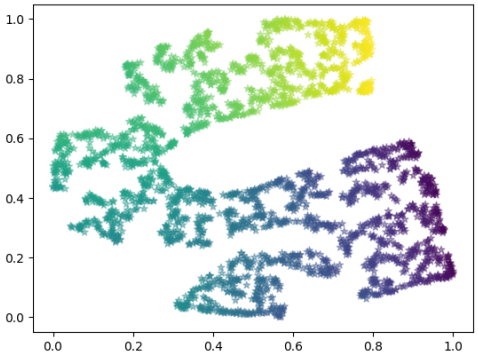
(d) projeção Árvores Randômicas



(e) projeção LLE Padrão



(f) projeção t-SNE



Fonte: Autoria própria.

4.2 RESULTADOS BASES REAIS

Ao analisar a base de dados MNIST, observou-se na Figura 27 que a maioria dos algoritmos apresentou uma visualização dos dados bastante aglomerada, dificultando a análise. Um exemplo claro disso foi as imagens geradas pelo PCA e MDS, que proporcionaram uma boa visualização para as bases de dados sintéticas, mas resultaram em uma visualização aglomerada quando aplicados a dados de maior dimensionalidade. Por outro lado, é evidente que a projeção do t-SNE obteve a melhor separação das informações presentes na estrutura de dados, por meio de agrupamentos.

No entanto, acredita-se que, no meio do processo de redução de dimensionalidade dos algoritmos, exceto o t-SNE, as técnicas que não possuíam um meio de manter a relação estrutural interna e sofreram com o problema conhecido como problema de aglomeração após um determinado ponto. Esse ponto foi mencionado por (JOHNSON; LINDENSTRAUSS, 1984), que abordaram o limite que poderia ser alcançado com baixa distorção no processo de redução de dimensionalidade. Apesar do t-SNE fornecer a melhor visualização dos dados por meio de agrupamentos, pôde-se verificar na Tabela 7 que era o algoritmo com a pior porcentagem de estresse.

Na Figura 28, que representa a base de dados Iris, pôde-se observar que todos os métodos apresentaram uma boa visualização após a redução dimensional. Isso se deve à natureza da base de dados ser de baixa dimensionalidade e possuir agrupamentos bem definidos. Nas imagens, foi perceptível que t-SNE não apenas agrupou bem o grupo mais afastado, mas também aumentou a distância entre os grupos.

Tabela 7 – Tabela de Stress para a base MNIST

Algoritmo	KruskalStress(%)
MDS	38.76
Isomap	57.16
PCA	64.42
Árvores randômicas	68.66
LLE padrão	99.76
t-SNE	402.19

Fonte: Autoria própria.

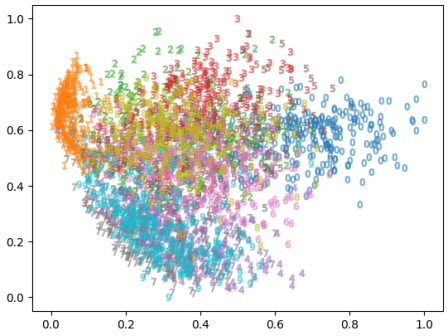
Tabela 8 – Tabela de Stress para a base da Iris

Algoritmo	KruskalStress(%)
PCA	4.18
MDS	5.96
Isomap	15.23
LLE padrão	95.06
Árvores randômicas	183.89
t-SNE	602.62

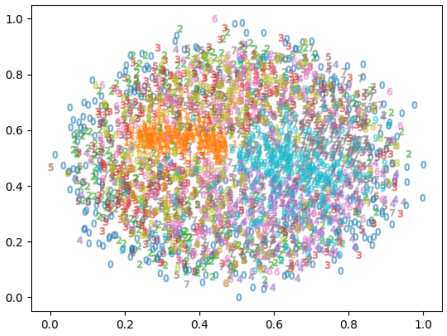
Fonte: Autoria própria.

Figura 27 – Projeção das transformação para a base MNIST

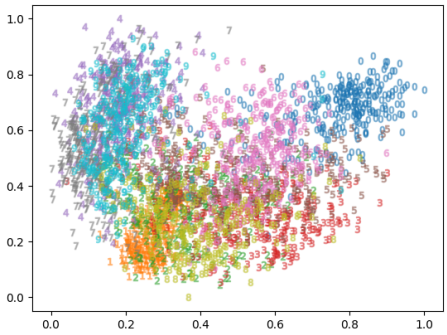
(a) projeção PCA



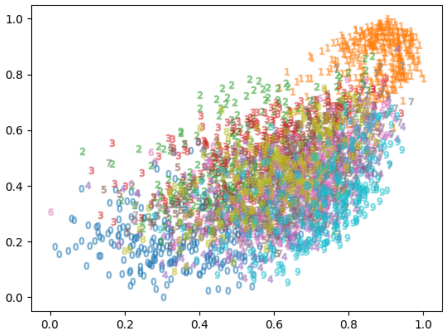
(b) projeção MDS



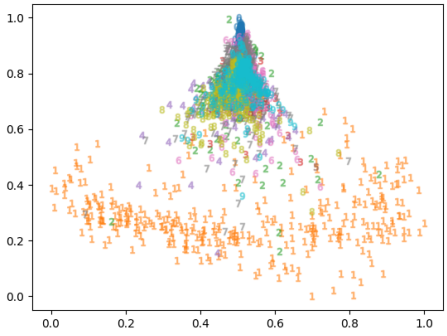
(c) projeção Isomap



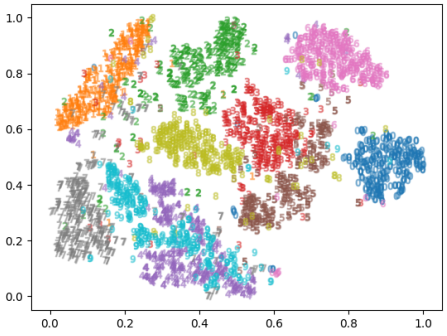
(d) projeção Árvores Randômicas



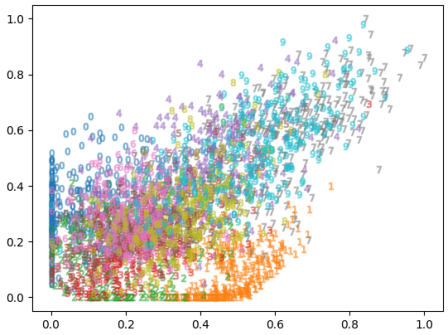
(e) projeção LLE Padrão



(f) projeção t-SNE



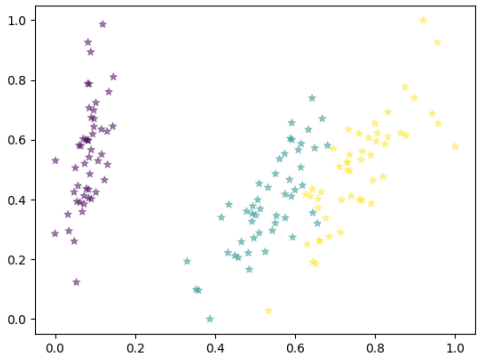
(g) projeção Autoencoder



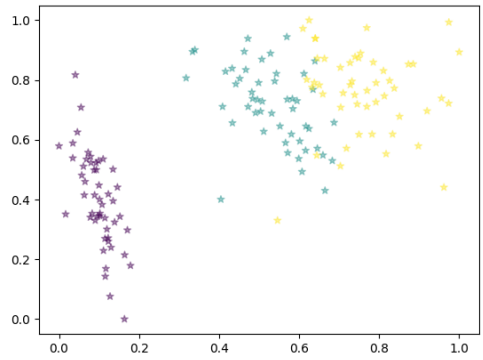
Fonte: Autoria própria.

Figura 28 – Projeção das transformação para a base planta Iris

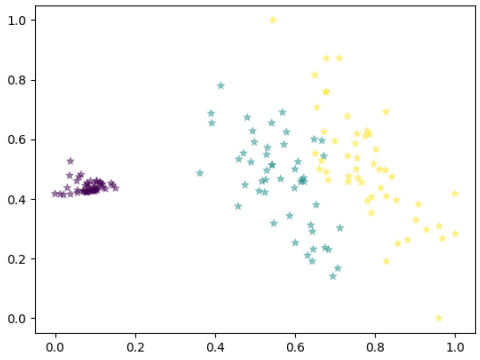
(a) projeção PCA



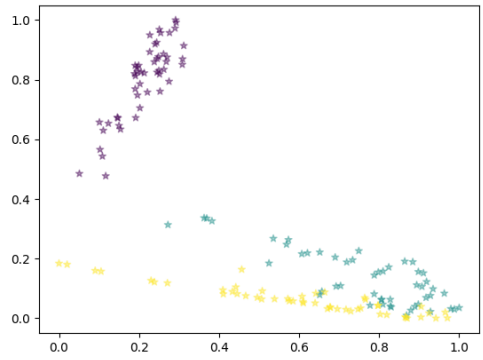
(b) projeção MDS



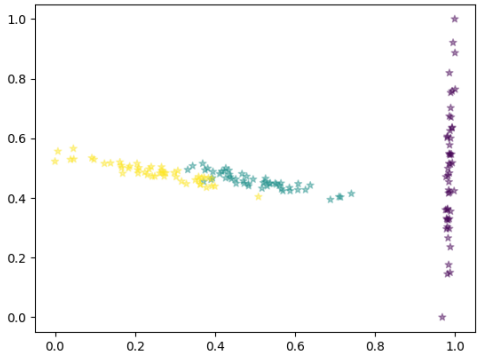
(c) projeção Isomap



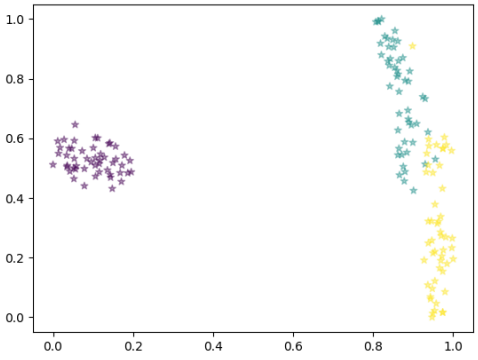
(d) projeção Árvores Randômicas



(e) projeção LLE Padrão



(f) projeção t-SNE



Fonte: Autoria própria.

5 CONCLUSÃO

Notou-se que, ao aplicarem as reduções em dados de baixa dimensionalidade, os algoritmos lineares (PCA e MDS) se destacaram com resultados positivos, apresentaram distorções mínimas. No entanto, ao observarem os algoritmos não lineares, apesar das distorções, percebe-se, por meio da visualização, que existe uma preservação da estrutura dos dados que não pode ser representada pela métrica de Kruskal, que busca calcular a distorção da estrutura global. Isso fica mais evidente ao aplicar a métrica na redução de t-SNE na base de dados reais MNIST, onde, apesar de oferecer a melhor visualização das informações dos dados, apresentou o pior estresse calculado.

Portanto, com base nos testes realizados, constatou-se que o Estresse de Kruskal não tem sido a abordagem mais adequada para a comparação quantitativa e qualitativa dos algoritmos do trabalho. Um exemplo disso é o caso do t-SNE, no qual a limitação reside no fato de que o Estresse de Kruskal, baseado em Distâncias Euclidianas, não tem conseguido representar adequadamente a estrutura intrínseca de conjuntos de dados complexos, como no caso da base MNIST. Os algoritmos de redução de dimensionalidade não lineares, como o t-SNE, têm conseguido preservar de forma mais eficaz as relações de vizinhança local, mas essa característica não tem sido adequadamente refletida pelo Estresse de Kruskal. O mesmo tem ocorrido para o Isomap, que, apesar de ter preservado a relação de distância entre os pontos por meio de Distâncias Geodésicas, não tem sido uma forma adequada de realizar comparações utilizando o Estresse de Kruskal, mesmo tendo proporcionado uma boa visualização.

Por fim, entende-se que seria ideal utilizar métricas distintas de distorção para algoritmos lineares e não lineares. Além disso, recomenda-se realizar uma análise segregada dos algoritmos, considerando aqueles que tendem a preservar as distâncias entre os dados e aqueles que possuem propriedades de conservação topológica, ou seja, algoritmos que preservam as estruturas internas em relação à dimensionalidade intrínseca dos dados. Essas abordagens permitem uma avaliação mais precisa e adequada dos algoritmos de redução de dimensionalidade, considerando suas características específicas. Tais considerações são importantes para direcionar futuras pesquisas e aplicações na área.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros, recomenda-se a aplicação de métricas distintas para algoritmos específicos, como o t-SNE, que possui uma natureza de agrupamento dos dados. Exemplos dessas métricas são o *Dunn Index* e a *Silhouette*, que podem ser utilizadas para avaliar os agrupamentos gerados. Essa abordagem permitirá obter uma visão mais completa dos efeitos desses algoritmos na estrutura dos dados. Outro aspecto relevante será utilizar mais bases de dados reais de alta dimensionalidade disponíveis na biblioteca da *scikit-learn*, explorando diferentes resultados. Será interessante também aplicar métricas que utilizem outras formas de análise de distância, como a Distância de Manhattan, a Distância de Chebyshev e a Distância de Minkowski, para obter perspectivas adicionais sobre a qualidade da redução de dimensionalidade. Esses pontos ajudarão a avançar o conhecimento sobre os algoritmos de redução de dimensionalidade, explorar diferentes abordagens e métricas, e aprimorar sua aplicação em diversos conjuntos de dados no futuro.

REFERÊNCIAS

BALKO, M. *et al.* **Almost-equidistant sets**. [*S. l.: s. n.*], 2020. arXiv: 1706.06375 [math.MG].

BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders, mar. 2020.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. **Artificial Intelligence**, v. 97, n. 1, p. 245–271, 1997. Relevance. Disponível em: http://www.sciencedirect.com/science/article/pii/S0004370297000635.

BOLÓN-CANEDO, V.; SÁNCHEZ-MAROÑO, N.; ALONSO-BETANZOS, A. A review of feature selection methods on synthetic data. **Knowledge and Information Systems**, v. 34, mar. 2012. DOI: 10.1007/s10115-012-0487-8.

CHARI, T.; PACHTER, L. The Specious Art of Single-Cell Genomics. **bioRxiv**, Cold Spring Harbor Laboratory, 2022. DOI: 10.1101/2021.08.25.457696. eprint: https://www.biorxiv.org/content/early/2022/12/22/2021.08.25.457696.full.pdf. Disponível em: https://www.biorxiv.org/content/early/2022/12/22/2021.08.25.457696.

CHOLLET, F. *et al.* **Keras**. 2015. Disponível em: https://github.com/fchollet/keras.

CLIM, A.; ZOTA, R. D.; TINICĂ, G. The Kullback-Leibler Divergence Used in Machine Learning Algorithms for Health Care Applications and Hypertension Prediction: A Literature Review. **Procedia Computer Science**, v. 141, p. 448–453, 2018. Disponível em: http://www.sciencedirect.com/science/article/pii/S1877050918317939.

COOK, J. *et al.* Visualizing Similarity Data with a Mixture of Maps. **Journal of Machine Learning Research - Proceedings Track**, v. 2, p. 67–74, jan. 2007.

DASGUPTA, S.; GUPTA, A. An Elementary Proof of a Theorem of Johnson and Lindenstrauss. **Random Struct. Algorithms**, v. 22, p. 60–65, jan. 2003. DOI: 10.1002/rsa.10073.

FUKUNAGA, K. **Introduction to Statistical Pattern Recognition (2Nd Ed.)** San Diego, CA, USA: Academic Press Professional, Inc., 1990.

GRACIA, A. *et al.* A methodology to compare Dimensionality Reduction algorithms in terms of loss of quality. **Information Sciences**, v. 270, p. 1–27, 2014. ISSN 0020-0255. DOI: https://doi.org/10.1016/j.ins.2014.02.068. Disponível em: https://www.sciencedirect.com/science/article/pii/S0020025514001741.

GRINSTEIN, G.; TRUTSCHL, M.; CVEK, U. High-Dimensional Visualizations. In.

GUYON, I. *et al.* (Ed.). **Feature Extraction - Foundations and Applications**. [*S. l.*]: Springer, 2006. v. 207. (Studies in Fuzziness and Soft Computing). ISBN 978-3-540-35488-8. Disponível em: http://dblp.uni-trier.de/db/books/collections/GNGZ2006.html.

HALL, M. A. **Correlation-based Feature Selection for Machine Learning**. [*S. l.*], 1999.

HINTON, G. E.; ROWEIS, S. T. Stochastic Neighbor Embedding. In: BECKER, S.; THRUN, S.; OBERMAYER, K. (Ed.). **Advances in Neural Information Processing Systems 15**. [*S. l.*]: MIT Press, 2003. P. 857–864. Disponível em: http://papers.nips.cc/paper/2276-stochastic-neighbor-embedding.pdf.

HOFFMAN, P.; GRINSTEIN, G. A survey of visualizations for high-dimensional data mining. In: p. 47–82.

JACOBS, R. A. Increased rates of convergence through learning rate adaptation. **Neural Networks**, v. 1, n. 4, p. 295–307, 1988. Disponível em: http://www.sciencedirect.com/science/article/pii/0893608088900032.

JINDAL, P.; KUMAR, D. A Review on Dimensionality Reduction Techniques. **International Journal of Computer Applications**, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 173, n. 2, p. 42–46, set. 2017. Disponível em: http://www.ijcaonline.org/archives/volume173/number2/28311-2017915260.

JOHNSON, W.; LINDENSTRAUSS, J. Extensions of Lipschitz maps into a Hilbert space. **Contemporary Mathematics**, v. 26, p. 189–206, jan. 1984. DOI: 10.1090/conm/026/737400.

KASKI, S.; PELTONEN, J. Dimensionality Reduction for Data Visualization. English. **IEEE - Signal Processing Magazine**, Institute of Electrical e Electronics Engineers Inc., v. 28, n. 2, p. 100–104, 2011.

KRUSKAL, J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. **Psychometrika**, Springer, v. 29, n. 1, p. 1–27, 1964.

KRUSKAL, J.; WISH, M. **Multidimensional Scaling**. [*S. l.*]: Sage Publications, 1978.

LIU, H.; YU, L. Yu, L.: Toward Integrating Feature Selection Algorithm for Classification and Clustering. IEEE Transaction on Knowledge and Data Engineering 17(4), 491-502. **IEEE Transactions on Knowledge and Data Engineering - TKDE**, v. 17, p. 491–502, abr. 2005. DOI: 10.1109/TKDE.2005.66.

MAATEN, L. van der; HINTON, G. Visualizing Data using t-SNE. **Journal of Machine Learning Research 9**, 2008.

MAATEN, L. van der; POSTMA, E.; HERIK, H. Dimensionality Reduction: A Comparative Review. **Journal of Machine Learning Research - JMLR**, v. 10, jan. 2007.

MAKLIN, C. **KL Divergence Python Example**. 2019. Disponível em: https://towardsdatascience.com/kl-divergence-python-example-b87069e4b810. Acesso em: 19 nov. 2019.

MARSLAND, S. **Machine Learning: An Algorithmic Perspective, Second Edition**. 2nd. [*S. l.*]: Chapman & Hall/CRC, 2014.

MARTIN ABADI *et al.* **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. [*S. l.: s. n.*], 2015. Software available from tensorflow.org. Disponível em: https://www.tensorflow.org/.

MIAO, J.; NIU, L. A Survey on Feature Selection. **Procedia Computer Science**, v. 91, p. 919–926, dez. 2016. DOI: 10.1016/j.procs.2016.07.111.

NASREEN, S. A Survey Of Feature Selection And Feature Extraction Techniques In Machine Learning,SAI,2014. In.

PEARSON, K. LIII. On lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin philosophical magazine and journal of science**, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PIROLLA, F. R. **Redução de dimensionalidade usando agrupamento e discretização ponderada para a recuperação de imagens por conteúdo**. Out. 2012. Diss. (Mestrado) – Universidade Federal de São Carlos.

PREETI SHARMA, R. P. S. A Review on Non Linear Dimensionality Reduction Techniques for Face Recognition. **International Journal on Recent and Innovation Trends in Computing and Communication**, v. 5, n. 7, p. 195–200, jul. 2017.

REMESH, R.; V, P. A survey on the cures for the curse of dimensionality in big data. **Asian Journal of Pharmaceutical and Clinical Research**, v. 10, p. 355, jul. 2017.

RUMELHART, D. E.; MCCLELLAND, J. L.; PDP RESEARCH GROUP, C. (Ed.). **Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations**. Cambridge, MA, USA: MIT Press, 1986. ISBN 026268053X.

SAMMON, J. W. A Nonlinear Mapping for Data Structure Analysis. **IEEE Transactions on Computers**, v. C-18, n. 5, 1969.

SHEPARD, R. N. Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. **Psychometrika**, v. 22, p. 325–345, 1957.

SILVA, V. D.; TENENBAUM, J. B. Global Versus Local Methods in Nonlinear Dimensionality Reduction. In: BECKER, S.; THRUN, S.; OBERMAYER, K. (Ed.). **Advances in Neural Information Processing Systems 15**. [*S. l.*]: MIT Press, 2003. P. 721–728. Disponível em: http://papers.nips.cc/paper/2141-global-versus-local-methods-in-nonlinear-dimensionality-reduction.pdf.

SONG, C. *et al.* Auto-encoder Based Data Clustering. In: IBEROAMERICAN Congress on Pattern Recognition. [*S. l.: s. n.*], 2013.

TORGERSON, W. S. Multidimensional scaling: I. Theory and method. **Psychometrika**, v. 17, n. 4, p. 401–419, dez. 1952. Disponível em: https://doi.org/10.1007/BF02288916.

VANDERPLAS, J. T. **Python Data Science Handbook: Essential Tools for working with data**. [*S. l.*]: O'Reilly Media, Inc., 2017.

WOLD, S.; ESBENSEN, K.; GELADI, P. Principal component analysis. **Chemometrics and Intelligent Laboratory Systems**, v. 2, n. 1, p. 37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists. ISSN 0169-7439. DOI: https://doi.org/10.1016/0169-7439(87)80084-9. Disponível em: https://www.sciencedirect.com/science/article/pii/0169743987800849.