

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO

Matheus Kirstus

**Mapeamento isométrico de atributos baseado
em geometria diferencial para aprendizado de
métricas não supervisionado**

São Carlos - SP

2021

Matheus Kirstus

**Mapeamento isométrico de atributos baseado em
geometria diferencial para aprendizado de métricas não
supervisionado**

Trabalho de conclusão de curso apresentado
para obtenção do título de Engenheiro de
Computação.

Orientação Prof. Dr. Alexandre L. M. Levada

São Carlos - SP

2021

Resumo

O ato de representar um conjunto de dados de maneira mais compacta e significativa é denominado redução de dimensionalidade. A capacidade de construir funções de distância adaptativas para cada conjunto de dados antes do estágio de classificação é chamada de aprendizado de métricas. Algoritmos de aprendizado de variedades têm se mostrado como poderosos métodos para aprendizado de métricas baseado em redução de dimensionalidade por extrair das amostras atributos não-lineares relevantes para classificação.

Este trabalho propõe o K-ISOMAP, um método que utiliza conceitos de geometria diferencial para construir uma função de distância intrínseca aos dados para aproximar as distâncias geodésicas no grafo KNN usando a noção de curvatura. Ao substituir a distância Euclidiana extrínseca entre os amostras vizinhas por uma medida da variação dos espaços tangentes em cada vizinhança, temos indícios de que o método proposto seja mais robusto em relação à ruído presente nos dados. Resultados experimentais com diversos conjuntos de dados reais demonstram que o método proposto é capaz de produzir uma performance de classificação melhor do que outros métodos de redução de dimensionalidade existentes na literatura.

Palavras-chave: Aprendizado de variedades. Aprendizado de métricas. ISOMAP. Curvatura.

Abstract

The act of representing a dataset in a way that's more compact and significant is denominated dimensionality reduction. The capacity of building adaptive distance functions to each dataset before classification is known as metric learning. Manifold learning algorithms have been shown to be powerful methods for dimensionality reduction based metric learning, as they extract from the samples non-linear attributes that are relevant for classification.

This work proposes K-ISOMAP, a method that uses differential geometry concepts to build a intrinsic distance function to approximate the geodesic distances in the KNN graph using the notion of curvature. By replacing the extrinsic Euclidean distance between neighboring samples by a measure of the variation in the tangent space at each neighborhood, there are signs that the proposed method is more robust against the presence of noise in data. Experimental results with several real world datasets show that the proposed method is capable of producing better classification performance than other dimensionality reduction methods that exist in the literature.

Keywords: Machine learning. Manifold learning. ISOMAP. Curvature.

Lista de ilustrações

Figura 1	–	Aproximação do vetor tangente entre os pontos r e s em uma curva $\vec{\alpha}(t)$	6
Figura 2	–	Espaço tangente Tp passando por p no manifold M	7
Figura 3	–	Ilustração de uma carta do <i>manifold</i> M mapeando a vizinhança U_0 em um espaço euclidiano \mathbb{R}^m	7
Figura 4	–	Duas vizinhanças U e V sobrepostas no <i>manifold</i> M , com uma transformação de coordenadas definida entre as vizinhanças em \mathbb{R}^m	8
Figura 5	–	Comparação da acurácia de classificação entre o ISOMAP e o K-ISOMAP em 58 datasets após a redução de dimensionalidade.	23

Lista de tabelas

Tabela 1 – p -valores resultantes do teste de Nemenyi para coeficientes de silhueta.	22
Tabela 2 – p -valores resultantes do teste de Nemenyi para acurácias médias de classificação.	23
Tabela 3 – Quantidade de amostras, atributos e classes para todos os conjuntos de dados utilizados neste trabalho, disponíveis em openML.org.	37
Tabela 4 – Acurácia média obtida por 8 classificadores após utilizar o K-ISOMAP (nas variantes A e B) e outros métodos de redução de dimensionalidade, para diversos conjuntos de dados disponíveis em openML.org. (Caso 2-D)	38
Tabela 5 – Coeficientes de silhueta do agrupamento de dados produzido pelo K-ISOMAP (na variantes A e B) e outros métodos de redução de dimensionalidade, para diversos conjuntos de dados disponíveis em openML.org. (Caso 2-D)	39

Lista de abreviaturas

DT	Decision Trees
GPC	Gaussian Process Classifier
ISOMAP	Isometric Feature Mapping
KNN	k Nearest Neighbors
KPCA	Kernel Principal Component Analysis
LLE	Locally Linear Embedding
LPP	Locality Preserving Projections
MDS	Multidimensional Scaling
MLP	Multilayer Perceptron
NB	Naive Bayes
PCA	Principal Component Analysis
QDA	Quadratic Discriminant Analysis
RFC	Random Forest Classifier
SVM	Support vector machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection

Sumário

1	INTRODUÇÃO	1
2	EMBASAMENTO TEÓRICO	3
2.1	Redução de dimensionalidade	3
2.1.1	Aprendizado de métricas	4
2.1.2	Manifold Learning	4
2.2	Geometria Diferencial	5
2.2.1	Curvas	5
2.2.1.1	Comprimento de Arco	5
2.2.1.2	Curvatura	6
2.2.1.3	Espaço tangente	6
2.2.2	Manifolds	7
2.2.3	Equações de Frenet-Serret	8
2.2.3.1	Equações de Frenet-Serret em n dimensões	9
2.3	Algoritmos de redução de dimensionalidade	9
2.3.1	Principal Component Analysis	9
2.3.1.1	Estimação do espaço tangente local por meio do PCA	10
2.3.1.2	Kernel Principal Component Analysis	10
2.3.2	Multidimensional Scaling	11
2.3.3	Isometric Feature Mapping	13
2.3.3.1	Relação entre ISOMAP e Kernel PCA	14
2.4	Trabalhos relacionados	15
3	DESENVOLVIMENTO DO TRABALHO	17
3.1	K-ISOMAP: mapeamento isométrico baseado em curvatura	17
3.1.1	Aproximação da curvatura por diferença finita	18
3.1.2	Aproximação da curvatura por geometria diferencial	18
4	EXPERIMENTOS E RESULTADOS	20
4.1	Materiais e métodos	20
4.1.1	Dados	20
4.1.2	Software	20
4.2	Descrição dos Experimentos	20
4.3	Resultados obtidos	21
5	CONCLUSÕES	25

	REFERÊNCIAS	26
	APÊNDICE A – DETALHES MATEMÁTICOS	28
A.1	Curvas, comprimento de arco e curvatura	28
A.2	Equações de Frenet-Serret	31
A.2.1	Varição no vetor tangente	31
A.2.2	Variações no vetor binormal	32
A.2.3	Variações no vetor normal	32
A.3	Desenvolvimento matemático do MDS	33
	APÊNDICE B – TABELAS DE RESULTADOS	37

1 Introdução

Atualmente aprendizado de máquina é utilizado em inúmeras aplicações, como reconhecimento de imagens, sistemas de recomendação, análises preditivas e processamento de linguagem natural. O reconhecimento de padrões é um dos componentes mais cruciais do aprendizado de máquina, sendo necessário em sistemas de classificação, predição, interpretação e recomendação.

É intuitivo que quanto maior o número de atributos analisados, mais fácil será de chegar à uma conclusão sobre os dados, porém quaisquer dados multidimensionais extraídos do mundo real apresentam algum nível de ruído. Nesse cenário, classificadores simples apresentam resultados subótimos, e muitas vezes falham em aprender os padrões presentes nos dados.

A área de redução de dimensionalidade existe dentro da área de aprendizado de máquina, com objetivo de pré-processar os dados e alimentar o sistema de classificação com conjuntos de dados de dimensões reduzidas e estatisticamente relevantes para a classificação, onde cada um dos atributos aprendidos pode ser uma combinação linear ou não-linear dos atributos originais da amostra.

A disposição das amostras dentro de um conjunto de dados pode seguir algum padrão não-linear, de modo que os dados se encontram em um plano, porém este plano se encontra esticado, rotacionado, enrolado, torcido e curvado em diferentes direções no conjunto de dados original. Muitas vezes a utilização da distância Euclidiana como medida de similaridade entre as amostras pode não resultar em bons resultados. Para extrair atributos relevantes desses dados, é de extrema importância um método adaptativo que produza funções de distância para cada conjunto de dados. Esse modelo adaptativo é denominado aprendizado de métricas.

Este trabalho tem como objetivo apresentar uma contribuição no aprendizado de métricas de um dos algoritmos de redução de dimensionalidade não-lineares mais utilizados na atualidade.

O restante deste trabalho será dividido em 3 partes: embasamento teórico; desenvolvimento; e experimentos e resultados.

A primeira parte apresenta os conceitos de geometria diferencial, redução de dimensionalidade e aprendizado de variedades, explicações breves das principais técnicas na qual esse trabalho se baseia, e traça paralelos com trabalhos relacionados.

Na segunda parte é detalhado o método de aprendizagem métrica baseado em geometria diferencial proposto por este trabalho.

Na terceira parte são apresentados os experimentos realizados com diversos conjuntos de dados, assim como a comparação de resultados obtidos com outros algoritmos semelhantes.

Ao final do documento encontram-se as conclusões com base nos resultados obtidos e possíveis contribuições posteriores ou trabalhos futuros.

2 Embasamento teórico

Esse capítulo está dividido em quatro seções. Inicialmente será feita uma breve apresentação da área geral de *machine learning*, e dos tópicos específicos aos quais esse trabalho está relacionado, como reconhecimento de padrões, redução de dimensionalidade e *manifold learning*. Seguidamente apresenta-se uma breve revisão de conceitos de curvas e geometria diferencial necessários para a compreensão do desenvolvimento deste trabalho. A terceira parte será dedicada a explicar em detalhes o funcionamento dos algoritmos que esse trabalho utiliza e se baseia. Na última seção desse capítulo se encontra um resumo de trabalhos recentes relacionados.

2.1 Redução de dimensionalidade

Redução de dimensionalidade consiste em encontrar uma representação mais compacta dos dados observados, em uma etapa que precede o reconhecimento de padrões ou qualquer tipo de processamento do conjunto de dados.

Intuitivamente, um dos motivos para reduzir a dimensionalidade de um conjunto de dados sem que haja perdas é a eficiência computacional, se as mesmas conclusões podem ser obtidas à partir de um conjunto de dados conciso então processar um conjunto de dados de maiores dimensões irá utilizar recursos computacionais desnecessariamente.

Outro motivo mais importante e menos intuitivo para a reduzir a dimensão de um conjunto de dados é a chamada “maldição da dimensionalidade” (*dimensionality curse*), onde quanto maior é a diferença entre a dimensionalidade real e a dimensionalidade artificial dos dados, menor será a acurácia obtida na classificação. Portanto a existência de atributos desnecessários e irrelevantes no conjunto de dados resulta não somente em ineficiência, mas também em ineficácia (REMESH; V, 2017).

Os métodos de redução de dimensionalidade tentam encontrar a dimensionalidade intrínseca do conjunto de dados, e transformar esses dados em uma forma mais compacta com o mínimo de perdas de informações. Esse objetivo pode ser atingido por diversos caminhos diferentes com vários níveis de complexidade e diferentes acurácias em cada tipo de conjunto de dados. Métodos para reduzir dimensionalidade podem ser lineares ou não lineares, considerar vizinhanças locais ou todo o conjunto de forma global, e utilizar diversas ferramentas matemáticas e diferentes hipóteses sobre os dados.

2.1.1 Aprendizado de métricas

O objetivo do aprendizado de métricas é inferir uma métrica de similaridade intrínseca ao conjunto de dados, de modo a otimizar os resultados obtidos na classificação. Uma das métricas de similaridades mais comumente utilizadas é a distância, como a distância Euclideana, similaridade por cosseno, ou distância de Mahalanobis.

Qualquer distância aprendida pode ser vista como a combinação de uma função que mapeia cada amostra a um espaço Euclidiano, seguido de um cálculo realizado entre a função aplicada em dois pontos (WANG; SUN, 2014).

A ideia básica da aprendizagem métrica de distâncias é reduzir a distância intra-classe e maximizar a distância inter-classe, ou em outras palavras, deixar cada ponto próximo a outros pontos de mesmo rótulo e distante de pontos que possuem rótulos diferentes. Por hipótese, a definição da métrica de distância que atinge esses objetivos depende intrinsecamente do próprio conjunto de dados, portanto é de extrema dificuldade obter uma métrica de distância universal que performe bem em diferentes tipos de dados (MCINNES; HEALY; MELVILLE, 2020).

2.1.2 Manifold Learning

Aprendizado de variedades, aprendizagem múltipla ou *manifold learning* é um tipo de redução de dimensionalidade não linear, com a premissa de que a alta dimensão do conjunto de dados é artificial, e que existe uma imersão de baixa dimensionalidade na qual todos os dados estão localizados. Assim, a geometria intrínseca dos dados existe em um espaço de menos dimensões, mas que no espaço dimensional dos dados de entrada essa geometria pode estar curvada, dobrada ou torcida. Ou seja, os pontos de dados são, na verdade, amostras de uma variedade de baixa dimensão que está embutida/imersa em um espaço de alta dimensão.

A hipótese múltipla (*manifold hypothesis*) afirma que os dados de altas dimensões do mundo real estão em *manifolds* de baixa dimensão imersos nesse espaço de alta dimensão (BENGIO; COURVILLE; VINCENT, 2013). Com isso, sabemos que ao reduzir a dimensionalidade dos nossos dados de modo que não haja distorções ou perdas, podemos fazer o processamento desses dados de forma mais eficiente, economizando recursos computacionais ao utilizar os dados em sua forma mais concisa.

A definição formal de um *manifold* ou variedade será introduzida posteriormente neste trabalho, mas em termos simples podemos definir um *manifold* como uma hiper-superfície que é localmente semelhante a um espaço Euclidiano. Isto é, dado um ponto p qualquer nessa hiper-superfície, existe uma função que mapeia cada um dos pontos da vizinhança de p em um espaço Euclidiano com dimensão menor que a dimensão da hiper-superfície, para todos os pontos desse *manifold*.

A terminologia padrão de *manifolds* se utiliza de palavras como cartas, mapas e atlas, o que ajuda a perceber intuitivamente que a superfície de uma esfera em 3 dimensões é um *manifold* de 2 dimensões, dado que grande parte das descrições geográficas e sistemas de navegação do nosso mundo real são exibidas em 2D sem perda de informações.

2.2 Geometria Diferencial

Nesta seção será apresentada uma revisão dos principais conceitos de geometria diferencial, vista em detalhes em [O’Neill \(2006\)](#), [Shifrin \(2006\)](#), [Carmo \(1976\)](#). Especificamente, será apresentada uma breve revisão de curvas em \mathbb{R}^3 e definição de curvatura e, em seguida, serão apresentadas as equações de Frenet-Serret no caso de 3 dimensões e sua generalização em n dimensões.

2.2.1 Curvas

Seja $\alpha : [a, b] \rightarrow \mathbb{R}^3$ uma curva parametrizada diferenciável. Se as coordenadas de α são dadas por $\alpha = (x(t), y(t), z(t))$, então para cada ponto t temos que o vetor tangente à curva neste ponto é:

$$\alpha'(t) = (x'(t), y'(t), z'(t))$$

Os itens a seguir são indicados somente como revisão dos conceitos sobre curvas, os detalhes e explicações matemáticas estão disponíveis para consulta no Apêndice [A.1](#).

2.2.1.1 Comprimento de Arco

Um dos conceitos fundamentais sobre curvas é o comprimento de arco de uma curva parametrizada. O comprimento de arco entre os pontos $r, s \in [a, b]$ de uma função $\vec{\alpha}(t)$ pode ser aproximado pela diferença dos valores da função aplicados nesses dois pontos.

No caso em que os pontos r e s forem muito próximos ($a - b = \Delta t \approx 0$), é possível aplicar o conceito de limite e portanto o comprimento de arco é dado por:

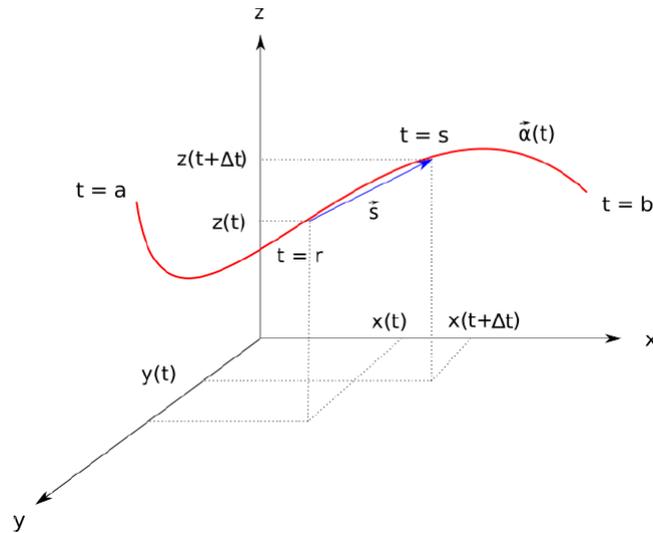
$$s(t) = \int_r^s \|\vec{\alpha}'(t)\| dt$$

Em alguns casos, é necessário parametrizar uma curva com base no seu comprimento de arco, de modo que sua primeira derivada esteja sempre no intervalo $[0, 1]$, como no caso das equações de Frenet Serret. Isso implica que os vetores tangente e normal (assim como binormal, trinormal e assim em diante) terão norma unitária.

O vetor tangente é dado por:

$$\vec{T}(t) = \alpha'(s) = \frac{\vec{\alpha}'(t)}{\|\vec{\alpha}'(t)\|}$$

Figura 1 – Aproximação do vetor tangente entre os pontos r e s em uma curva $\vec{\alpha}(t)$



Fonte: Lima (2007)

A segunda derivada de $\alpha(t)$ representa o vetor normal, que é ortogonal ao vetor tangente. Para obter o vetor normal com norma unitária, temos:

$$\vec{N}(t) = \frac{T'(t)}{\|T'(t)\|} = \frac{\vec{\alpha}''(s)}{\|\vec{\alpha}''(s)\|}$$

2.2.1.2 Curvatura

A curvatura principal de $\vec{\alpha}(s)$ em um ponto qualquer é exatamente a taxa de variação instantânea da direção da curva:

$$K(s) = \|\vec{\alpha}''(s)\| = \|\vec{T}'(t)\|$$

Como pode ser visto no Apêndice A.1, com algum esforço matemático o mesmo valor de curvatura em um ponto pode ser obtida à partir da expressão abaixo, utilizando valores conhecidos de $\vec{\alpha}(t)$:

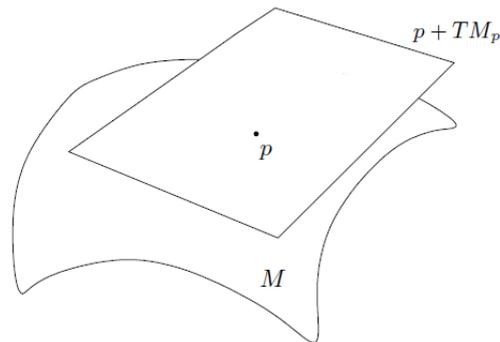
$$K(s) = \|\vec{\alpha}''(s)\| = \frac{\|\vec{\alpha}'(t) \times \vec{\alpha}''(t)\|}{\|\vec{\alpha}'(t)\|^3} \tag{1}$$

2.2.1.3 Espaço tangente

Seja p um ponto de uma superfície em \mathbb{R}^3 . Então o vetor tangente v em \mathbb{R}^3 no ponto p é tangente à superfície representa a velocidade de alguma curva nessa superfície. O conjunto de todos os vetores tangentes ao ponto p da superfície é chamado de espaço tangente T_p .

O espaço tangente é diferente para cada ponto p da curva, e seus vetores são calculados por meio de derivações aplicadas no ponto. Como todos os vetores tangentes em um espaço tangente são aplicados no mesmo ponto e, por definição, são ortogonais entre si, as operações de soma e multiplicação por escalar válidas para um espaço vetorial também podem ser realizadas no espaço tangente.

Figura 2 – Espaço tangente Tp passando por p no manifold M



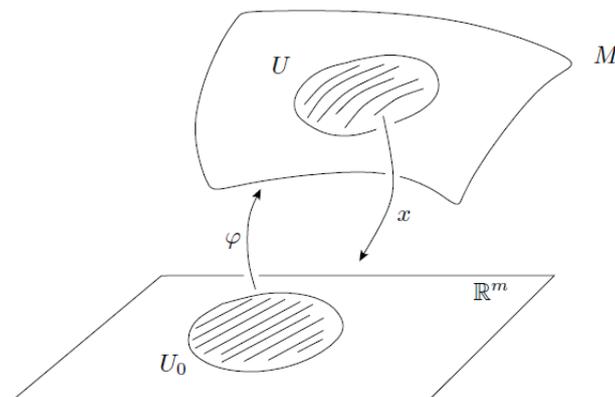
Fonte: Lima (2007)

Para uma superfície em \mathbb{R}^3 , o espaço tangente é comumente chamado de plano tangente. No caso genérico para uma hipersuperfície em \mathbb{R}^n o espaço tangente será um espaço vetorial de n dimensões.

2.2.2 Manifolds

Em termos simples, uma Variedade ou *Manifold* é uma generalização de curvas e superfícies para dimensões arbitrárias (TU, 2011). Por questões de simplicidade, iremos considerar manifolds que podem ser incorporados em um espaço \mathbb{R}^n .

Figura 3 – Ilustração de uma carta do *manifold* M mapeando a vizinhança U_0 em um espaço euclidiano \mathbb{R}^m

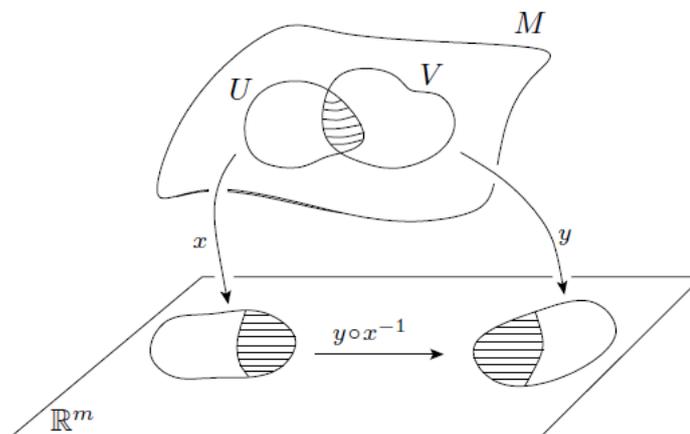


Fonte: Lima (2007)

Considerando o *manifold* M , um sistema de coordenadas locais ou carta local é um homeomorfismo $x : U \rightarrow x(U)$ de um conjunto aberto $U \subset M$ sobre um aberto $x(U) \subset \mathbb{R}^m$. Em outras palavras, uma carta local mapeia a vizinhança de um ponto de M em um espaço euclidiano \mathbb{R}^m . Um conjunto de cartas cujos domínios cobrem M é chamado de atlas.

Um *manifold* diferenciável pode ser definido como um conjunto de pontos com vizinhanças homeomórficas à um espaço Euclidiano, em que as coordenadas de vizinhanças que se sobrepõem são relacionadas por uma transformação diferenciável (WHITNEY, 1996).

Figura 4 – Duas vizinhanças U e V sobrepostas no *manifold* M , com uma transformação de coordenadas definida entre as vizinhanças em \mathbb{R}^m



Fonte: Lima (2007)

2.2.3 Equações de Frenet-Serret

Seja $\vec{B}(t) = \vec{T}(t) \times \vec{N}(t)$ o vetor binormal unitário. O conjunto de vetores unitários $\{\vec{T}(t), \vec{N}(t), \vec{B}(t)\}$ formam uma base ortonormal em cada ponto t que a curva está definida. Esse referencial móvel dependente da curva parametrizada α é chamado de Triedro de Frenet. As fórmulas de Frenet-Serret permitem calcular as variações dessa base ortonormal na curva à partir dos próprios vetores da base, e da definição de curvatura e torção.

As equações de Frenet-Serret em 3 dimensões são:

$$\begin{aligned} \vec{T}'(t) &= K(t)\vec{N}(t) \\ \vec{N}'(t) &= -K(t)\vec{T}(t) + \tau\vec{B}(t) \\ \vec{B}'(t) &= -\tau\vec{N}(t) \end{aligned} \tag{2}$$

As operações matemáticas para se chegar nas equações de Frenet-Serret e as variações de cada um dos vetores do Triedro de Frenet se encontram no Apêndice A.2.

2.2.3.1 Equações de Frenet-Serret em n dimensões

No caso n -dimensional a base ortonormal é composta pelos vetores $\{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$, e as equações de Frenet Serret são generalizadas da seguinte forma (ITO, 1987)

$$\begin{aligned} e'_i(t) &= -K_{i-1}(t)e_{i-1}(t) + -K_i(t)e_{i+1}(t) \quad i = 1, \dots, n \\ K_0(t) &= K_n(t) = 0 \end{aligned} \quad (3)$$

De forma mais visual, as equações de Frenet-Serret podem ser obtidas pela matriz abaixo, expandindo o mesmo padrão para quantas dimensões forem necessárias. Denominando $T = e_1, N = e_2, B = e_3, W = e_4$, temos:

$$\begin{pmatrix} T' \\ N' \\ B' \\ W' \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 & K_1 & 0 & 0 & \dots \\ -K_1 & 0 & K_2 & 0 & \dots \\ 0 & -K_2 & 0 & K_3 & \dots \\ 0 & 0 & -K_3 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} T \\ N \\ B \\ W \\ \vdots \end{pmatrix} \quad (4)$$

Com isso, se torna evidente que o triedro de Frenet pode ser generalizado para dimensões mais altas, e as curvaturas principais de uma curva podem ser calculadas mesmo em dimensões de tamanho arbitrário.

2.3 Algoritmos de redução de dimensionalidade

2.3.1 Principal Component Analysis

O Principal Component Analysis (PCA), ou Análise dos Componentes Principais, é um método estatístico multivariável para converter observações de dados que podem estar correlacionados em variáveis linearmente independentes, chamadas de componentes principais. Amplamente utilizado em diversas áreas acadêmicas para exploração de conjuntos de dados, e é um dos algoritmos de redução de dimensionalidade linear mais conhecidos.

O PCA clássico consiste na diagonalizar a matriz de covariância dos dados para criar uma matriz de transformação das novas coordenadas. Seja o conjunto de dados $X_r, r = 1, \dots, n, x_r \in \mathbb{R}^p$, se a média dos dados for zero, a matriz de covariância é definida como:

$$C = \frac{1}{n} \sum_{j=1}^n x_j x_j^T$$

Como a matriz C é simétrica e positiva semi-definida, é possível realizar a decomposição espectral da forma:

$$C = V \Lambda V^T \quad (5)$$

onde V é a matriz com os autovetores de C e Λ é a matriz diagonal dos autovalores correspondentes. Para reduzir a dimensão do conjunto de dados, são escolhidos os d autovetores associados aos maiores autovalores da matriz de covariância.

As novas coordenadas dos dados são obtidas fazendo sua projeção os autovetores associados aos d maiores autovalores da matriz de covariância. As variáveis independentes obtidas dessa transformação são chamadas de componentes principais.

Seja \tilde{V} a matriz dos d maiores autovetores, os componentes principais são obtidos pela seguinte transformação linear:

$$Y = \tilde{V}X$$

2.3.1.1 Estimação do espaço tangente local por meio do PCA

Na diagonalização da matriz de covariância obtém-se p autovetores. Por definição, esses vetores de V são unitários e linearmente independentes, portanto formam uma base ortogonal em \mathbb{R}^p . Similarmente, os d autovetores associados aos maiores autovalores obtidos em \tilde{V} também formam uma base ortonormal em \mathbb{R}^d .

Com isso, ao aplicar [PCA](#) localmente na vizinhança de um ponto \vec{x}_i , é possível obter uma base ortogonal em cada ponto. Como as vizinhanças entre pontos próximos são semelhantes, o resultado obtido pelo PCA nesses pontos exibe pequenas diferenças devido à curvatura. Essa base ortogonal serve como uma estimação do espaço tangente neste ponto, conforme mostrado em detalhes em [Singer e Wu \(2012\)](#).

2.3.1.2 Kernel Principal Component Analysis

O Kernel PCA é uma adaptação do [PCA](#) que utiliza o conceito de funções de kernel para obter relações não lineares durante a redução de dimensionalidade apresentado por [Scholkopf, Smola e Müller \(1999\)](#). O conceito do Kernel PCA é mapear os dados em um espaço de maiores dimensões que está relacionado com o espaço de dados original de forma não linear.

Utilizando o truque do kernel (*kernel trick*), é possível trabalhar somente com o produto interno $\Phi(x_i) \cdot \Phi(x_j)$ sem conhecer explicitamente o mapeamento nos pontos i e j e mesmo sem conhecer a função $\Phi(x)$. Portanto todo o procedimento é realizado com base na matriz de kernel, que define os produtos internos no espaço de maiores dimensões:

$$K_{ij} = (\Phi(x_i) \cdot \Phi(x_j))$$

Caso os dados não estejam centralizados na origem é necessário a seguinte transformação linear:

$$\tilde{K}_{ij} = K - 1_n K - K 1_n + 1_n K 1_n \quad (6)$$

onde 1_n é uma matriz $n \times n$ onde todos os elementos são iguais a $1/n$.

Após a diagonalização da matriz \tilde{K} é necessário resolver o seguinte problema de autovalores:

$$n\lambda\tilde{K}\alpha = \tilde{K}^2\alpha$$

sujeito a

$$\lambda_k(\alpha^k \cdot \alpha^k) = 1$$

Para extrair as componentes principais ao kernel K de um ponto x_r em relação ao autovetor V^k é necessário calcular seu produto interno com todos os outros pontos do conjunto de dados:

$$(V^k \cdot \Phi(x)) = \sum_{i=1}^n \alpha_i^k (\Phi(x_i) \cdot \Phi(x_r)) = \sum_{i=1}^n \alpha_i^k (\tilde{K}_{ir})$$

2.3.2 Multidimensional Scaling

Multidimensional Scaling (**MDS**), ou escalonamento multidimensional, é uma técnica de redução de dimensionalidade linear que encontra uma imersão dos dados em um espaço de dimensionalidade menor que o original, mantendo a distância entre os pontos.

MDS, ou Escalonamento Multidimensional, é uma técnica de redução de dimensionalidade linear utilizada para encontrar uma imersão dos dados em um espaço de dimensionalidade menor que o original, e recuperar as coordenadas nesse espaço.

Seu funcionamento consiste em mapear os pontos em um espaço de baixa dimensão, de modo que a distância Euclidiana entre dois pontos nesse espaço corresponda ao valor da função de dissimilaridade entre os pontos no espaço original de alta dimensão, para todos os pares de pontos (**COX; COX, 2000**). No MDS clássico, a função de dissimilaridade é exatamente a distância Euclidiana entre os pontos, e o processo consiste em encontrar coordenadas em outras dimensões de modo que a distância entre os pontos se mantenha igual (**BORG; GROENEN, 2005**).

Sendo δ_{rs} a função de dissimilaridade entre os pontos r e s em alta dimensão, e d_{rs} a distância Euclidiana em baixa dimensão, temos:

$$\delta_{rs} = d_{rs}$$

Para recuperar as coordenadas dos pontos em baixa dimensão, o MDS possui dois passos principais:

1. Calcular a matriz de produtos internos à partir da matriz de distâncias.
2. Decompor a matriz de produtos internos em autovalores e autovetores e obter as coordenadas.

Considere que a matriz de coordenadas de n pontos em um espaço Euclidiano de p dimensões seja dada por $X = \{\vec{x}_r\}$, onde $r = 1, 2, \dots, n$ e $\vec{x}_r = (x_{r1}, \dots, x_{rp})^T$.

Seja $D = \{d_{rs}^2\}$ a matriz de distâncias entre esses pontos, e $B = \{b_{rs}\}$ a matriz de produtos internos, para $r, s = 1, 2, \dots, n$. Para dois pontos arbitrários \vec{x}_r e \vec{x}_s , temos:

$$d_{rs}^2 = \|\vec{x}_r - \vec{x}_s\|^2 = (\vec{x}_r - \vec{x}_s)^T (\vec{x}_r - \vec{x}_s) \quad (7)$$

$$b_{rs} = \vec{x}_r^T \vec{x}_s \quad (8)$$

A literatura nos mostra que a matriz de produtos internos $B = \{b_{rs}\}$ pode ser obtida pela equação abaixo. Para manter essa seção breve, todos os detalhes e explicações matemáticas se encontram no Apêndice A.3.

Temos que a matriz de produtos internos é computada por:

$$B = -\frac{1}{2}HDH \quad (9)$$

onde H é a matriz centralizadora definida como:

$$H = I - \frac{1}{n}U, \quad \text{onde } U_{n \times n} = \{1\} \quad (10)$$

Após obter os valores da matriz B , o passo seguinte é recuperar as coordenadas dos pontos em R^d , assim encontrando a incorporação dos dados. A matriz de produtos internos B é dada por:

$$B_{n \times n} = X_{n \times p}^T X_{n \times p} \quad (11)$$

onde n é o número de amostras e p é a dimensionalidade original dos dados de entrada. Considerando isso, sabemos de três características importantes sobre essa matriz (COX; COX, 2000): é simétrica, positiva semi-definida e possui posto p . Em outras palavras, a matriz B possui p autovalores não-negativos e $n - p$ autovalores iguais a zero.

Escrevendo a matriz B em termos da decomposição espectral, temos:

$$B = V\Lambda V^T \quad (12)$$

onde $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ é a matriz diagonal dos autovalores de B , e $V = [v_1, v_2, \dots, v_n]$ é uma matriz na qual as colunas são os autovetores correspondentes aos autovalores, normalizados.

Desconsiderando os $n - p$ autovalores iguais a zero e seus autovetores correspondentes, reescrevemos B como:

$$B = \tilde{V}\tilde{\Lambda}\tilde{V}^T \quad (13)$$

onde $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ e $V = [v_1, v_2, \dots, v_p]$.

Finalmente, sabendo que a matriz B é dada por $B = X^T X$ conforme a Equação (11), podemos recuperar as coordenadas de X :

$$X = \tilde{\Lambda}^{\frac{1}{2}} \tilde{V}^T \quad (14)$$

onde $\tilde{\Lambda}^{\frac{1}{2}} = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_p})$ é a matriz diagonal da raiz dos p autovalores diferentes de zero, e \tilde{V} é a matriz cujas colunas são os autovetores normalizados associados à esses p autovalores.

2.3.3 Isometric Feature Mapping

Isometric Feature Mapping (**ISOMAP**), ou mapeamento isométrico de características, é um método de redução de dimensionalidade não linear baseado no **MDS**, buscando preservar a geometria intrínseca dos dados ao computar as distâncias geodésicas entre cada par de amostras, apresentado em **Tenenbaum, Silva e Langford (2000)**.

Partindo do princípio que a superfície de *manifold* é localmente semelhante à um espaço Euclidiano, a distância Euclidiana entre pontos vizinhos são calculadas como distâncias intrínsecas. Para estimar a distância geodésica entre pontos distantes, são utilizados os caminhos já conhecidos pelas vizinhanças, e a distância de cada trecho é somada, de modo que a distância entre dois pontos seja o menor caminho.

Em suma, o **ISOMAP** consiste em três passos:

1. Formar um grafo com as distâncias locais entre os pontos vizinhos utilizando k Nearest Neighbors (**KNN**) ou ϵ -neighborhood;
2. Formar uma matriz com a aproximação das distâncias geodésicas encontrando os menores caminhos no grafo entre cada par de pontos;
3. Aplicar o algoritmo **MDS** na matriz global de distâncias e obter as novas coordenadas dados em dimensão reduzida preservando as distâncias entre os pontos.

Para um número suficiente de amostras, os caminhos mínimos no grafo de pontos vizinhos se aproxima da distância geodésica real assintoticamente (**BERNSTEIN et al., 2000**), desde que o grafo seja conexo e que o número de vizinhos não seja maior que a curvatura intrínseca do *manifold*.

O Algoritmo 1 abaixo exhibe a sequência de passos necessários para o **ISOMAP**,

considerando os passos necessários para a execução do MDS.

Algoritmo 1: ISOMAP

Entrada: $X_{m \times n}$ - Matriz de dados com m dimensões

Resultado: $X_{d \times n}$ - Matriz de dados com d dimensões

1 **início**

2 Construir um grafo KNN com as distâncias locais

3 Aplicar Dijkstra e obter a matriz de distâncias globais $D_{n \times n}$

4 Construir a matriz $H = I - \frac{1}{n}U$, onde $U = [1]_{n \times n}$

5 Calcular a matriz de produtos internos $B = -\frac{1}{2}HDH$

6 Encontrar os autovalores e autovetores de B

7 Selecionar os d maiores autovalores e definir $\tilde{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$

8 Definir a matriz $\tilde{V}_{n \times d} = \langle \vec{v}_1, \vec{v}_2, \dots, \vec{v}_d \rangle$ com os autovetores correspondentes à $\tilde{\Lambda}$

9 Recuperar as coordenadas $\tilde{X}_{d \times n} = \tilde{\Lambda}^{\frac{1}{2}} \tilde{V}^T$

10 **fin**

Algumas das limitações conhecidas da implementação original do ISOMAP são a projeção de dados fora da amostra original e a presença de duas regiões do manifold topologicamente distantes na mesma vizinhança, conhecidas como *out-of-sample problem* e *short-circuit*, respectivamente.

O primeiro desses problemas deriva do fato do ISOMAP ser um método global, o que demanda que o método seja aplicado novamente para todas as amostras sempre que uma única nova amostra for adicionada ao conjunto de dados. Esse custo adicional para cada amostra torna a projeção imediata de novos dados inviável, e geralmente o processamento em lote de novas amostras é uma alternativa mais utilizada.

Dependendo do tamanho da vizinhança escolhida, da topologia inerente dos dados e do nível de ruído, a presença de *outliers* no conjunto de dados pode causar um “curto-circuito” na saída gerada pelo ISOMAP, causando áreas que são distantes na superfície do *manifold* porém são próximas no espaço de dimensão reduzida. Nesses casos o *outlier* age como um atalho dentro da superfície, encurtando a distância estimada entre os pontos e resultando uma estimacão distorcida dos dados em baixa dimensão.

2.3.3.1 Relação entre ISOMAP e Kernel PCA

No Kernel PCA, após computar a matriz de kernel K é necessário centralizar os dados, removendo a média conforme a equação (6). Por conveniência, a equação está reescrita abaixo.

$$\tilde{K} = K - 1_n K - K 1_n + 1_n K 1_n$$

onde 1_n é uma matriz $n \times n$ onde todos os elementos são $1/n$. Podemos definir o vetor \vec{e} como:

$$\vec{e} = \frac{1}{\sqrt{n}}[1, 1, \dots, 1]^T$$

de modo que $1_n = \vec{e}\vec{e}^T$. Substituindo na equação (6), temos:

$$\begin{aligned} \tilde{K} &= K - \vec{e}\vec{e}^T K - K \vec{e}\vec{e}^T + \vec{e}\vec{e}^T K \vec{e}\vec{e}^T \\ &= [(I - \vec{e}\vec{e}^T)K] - [(I - \vec{e}\vec{e}^T)K]\vec{e}\vec{e}^T \\ &= (I - \vec{e}\vec{e}^T)K(I - \vec{e}\vec{e}^T) \end{aligned} \quad (15)$$

No **ISOMAP**, após computar os menores caminhos no grafo e obter a matriz de distâncias geodésicas aproximadas D , é necessário calcular a matriz de produtos internos B , conforme a equação (A.3.14), reescrita abaixo.

$$B = -\frac{1}{2}HDH$$

onde a matriz de centralização é $H = (I - \vec{e}\vec{e}^T)$, conforme a equação (A.3.13). Portanto:

$$K_{Isomap} = B = -\frac{1}{2}(I - \vec{e}\vec{e}^T)D(I - \vec{e}\vec{e}^T) \quad (16)$$

Se torna evidente a relação entre os dois métodos, e podemos dizer que o Kernel PCA se torna o **ISOMAP** quando a matriz de kernel é exatamente $K = -\frac{1}{2}D$, onde D é a matriz de distâncias geodésicas (HAM et al., 2004).

2.4 Trabalhos relacionados

Nesta seção são apresentados os principais artigos e contribuições que incorporaram diferentes técnicas e propuseram melhorias na aplicação do **ISOMAP**, que é objeto de estudo para este trabalho.

Em 2017 o artigo *Robust kernel Isomap* (CHOI; CHOI, 2007) foi publicado, sugerindo melhorias em duas áreas que faltam no **ISOMAP**: projeção de novos dados e tratamento de *outliers*. Para obter a propriedade de generalização, é feito um estudo detalhado da relação entre **ISOMAP** e kernel PCA. Para garantir que seja possível de decompor a matriz de kernel em autovalores e autovetores, é adicionado um problema de adição de constante à matriz de distâncias de modo que a matriz de kernel resultante seja sempre positiva e semi-definida.

Para garantir estabilidade topológica e impedir a ocorrência de curto-circuitos durante o processo, os autores propõem uma análise do fluxo de cada amostra para identificar *outlier*. A análise ocorre durante o cálculo dos caminhos mínimos no grafo, em uma etapa prévia à estimação de distância geodésica. O fluxo é definido pelo número

de vezes que um certo nó do grafo está presente no caminho mínimo entre dois outros pontos. A intuição é que, considerando a superfície em que os dados se encontram, o fluxo de pontos vizinhos deve ser semelhante. Se algum ponto apresentar um fluxo muito alto, é mais provável que isso seja indicativo de um curto-circuito e que esse ponto seja um *outlier*.

Para aumentar a robustez contra curto-circuitos, os autores de *Non-linear Dimensionality Reduction by Locally Linear Isomaps* (SAXENA; GUPTA; MUKERJEE, 2004) apresentam uma variante do ISOMAP que se utiliza de propriedades locais de forma semelhante ao de outros métodos como Locally Linear Embedding (LLE) e Laplacian Eigenmaps. À partir do fato de que localmente a superfície do *manifold* é semelhante à um espaço Euclidiano, as vizinhanças são determinadas representando cada ponto por meio da combinação linear de pontos próximos, em contraste aos meios tradicionais de KNN ou ϵ -neighborhood.

Outra contribuição que aplica métodos locais no ISOMAP é apresentado no artigo *Global versus local methods in nonlinear dimensionality reduction* (SILVA; TENENBAUM, 2002). Os autores propõe duas variações do algoritmo Conformal ISOMAP (C-ISOMAP) e Landmark ISOMAP (L-ISOMAP). Fazendo uma suposição de que o *manifold* foi amostrado uniformemente, o C-ISOMAP amplifica a contribuição de áreas com maior densidade e atenua a contribuição de áreas com baixa densidade de pontos, extendendo o algoritmo inicial com o reconhecimento da estrutura de certos *manifolds* curvados.

O L-ISOMAP substitui as computações globais por um conjunto de cálculos locais em pontos de referência, denominados *landmarks*, o que resulta em uma variação do algoritmo mais eficiente computacionalmente. Uma melhoria desse algoritmo escolhe automaticamente o número e a posição dos pontos de referência, de modo a reduzir erros de curto-circuito, como apresentado no artigo *An Extended Isomap for Manifold Topology Learning with SOINN Landmarks* (GAN; SHEN; ZHAO, 2014).

Dos três passos do ISOMAP, o MDS é o passo com maior complexidade espacial e computacional. No *paper Efficient Inter-Geodesic Distance Computation and Fast Classical Scaling* é apresentado uma variação mais rápida do MDS utilizando informações locais e globais, reduzindo a complexidade de quadrática para quase-linear ao interpolar as distâncias geodésicas de algumas amostras e pressupor que as amostras em um intervalo variam de forma suave.

Em contraste com a distância ponto a ponto do método clássico, o artigo *ISOMAP-KL: a parametric approach for unsupervised metric learning* Neto e Levada (2020) apresenta uma abordagem robusta na presença de *outliers*, baseada na distância entre *patches*, ou regiões. Mais especificamente, a métrica utilizada é uma distância não Euclidiana utilizando a divergência de Kullback-leibler entre distribuições Gaussianas obtidas das vizinhanças do grafo k NN.

3 Desenvolvimento do trabalho

Este capítulo apresenta o método proposto para calcular a distância geodésica intrínseca do conjunto de dados em baixa dimensão utilizando uma estimativa da curvatura extrínseca do manifold imerso em um espaço de altas dimensões dos dados originais.

3.1 K-ISOMAP: mapeamento isométrico baseado em curvatura

No algoritmo [ISOMAP](#) original, a distância geodésica entre os pontos é aproximada localmente pela distância Euclidiana, e as curvas no manifold são aproximadas pelos menores caminhos no grafo de distâncias. A mudança proposta no K-ISOMAP é utilizar a curvatura como distância entre pontos da mesma vizinhança ao invés da distância Euclidiana.

Em \mathbb{R}^3 , os vetores tangente, normal e binormal definem uma base ortogonal em um ponto da curva. O Triedro de Frenet utiliza as equações de Frenet-Serret para definir essa base ortogonal em todos os pontos da curva, com as variações de cada um de seus vetores expressas inteiramente em função dos outros vetores e da curvatura. A mesma noção pode ser generalizada para espaços euclidianos de dimensões arbitrárias, com espaço tangente e m curvaturas principais.

Estendendo essa ideia, é proposto que essa base ortonormal que define o espaço tangente em um ponto do *manifold* pode ser utilizada para extrair informações geodésicas. Vimos que a variação no campo tangente pode ser expressa com base na curvatura principal $K(t)$. É proposto que ao analisar a variação no espaço tangente entre dois pontos e calcular a curvatura principal nessa região é possível obter uma função de distância intrínseca. Com as distâncias locais levando em consideração a curvatura, é esperado que os caminhos mínimos no grafo serão uma aproximação mais acurada da distância geodésica no *manifold*.

Seja $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n]$ a matriz de dados, com $\vec{x}_i \in \mathbb{R}^m$. Assim como no método clássico, o primeiro passo consiste em construir o grafo KNN com as distâncias entre pontos em X . Em primeiro momento, para definir os vizinhos mais próximos de \vec{x}_i é utilizada a distância Euclidiana extrínseca. A região de \vec{x}_i contém esse ponto e todos os seus pontos vizinhos.

$$P_i = \{\vec{x}_i, \vec{x}_{i1}, \vec{x}_{i2}, \dots, \vec{x}_{ik}\}$$

Calculando a matriz de covariância na região P_i , é possível aplicar [PCA](#) e obter o espaço tangente nesse ponto, conforme visto em [2.3.1.1](#). Sendo Σ_i a matriz de covariância

de P_i , a decomposição em autovalores e autovetores da Equação (5) se torna:

$$\Sigma_i = U_i \Lambda_i U_i^T$$

onde $U_i = \{\vec{u}_{i1}, \vec{u}_{i2}, \dots, \vec{u}_{im}\}$ é uma matriz $m \times m$ onde as colunas são autovetores e Λ_i é uma matriz diagonal $m \times m$ com os autovalores de Σ_i .

Com isso, o espaço tangente no ponto \vec{x}_i é representado por V_i . O objetivo é obter a variação no espaço tangente entre pontos próximos, calcular a curvatura principal, e utilizar esses valores para atualizar as distâncias geodésicas.

São propostos dois métodos para estimar a curvatura principal entre pontos no *manifold*: 1) Aproximação por diferença finita e 2) Aproximação por geometria diferencial.

3.1.1 Aproximação da curvatura por diferença finita

A definição de curvatura é taxa de variação do vetor tangente ao longo da curva:

$$K(t) = \|\vec{T}'(t)\|$$

Então o jeito mais simples de aproximar a curvatura é utilizando um modelo de diferença finita. Seja $C = a_a, a_1, a_2, \dots, a_b$ o menor caminho entre amostras \vec{x}_a e \vec{x}_b no grafo KNN e seja P_i a região da i -ésima amostra em C . Como visto anteriormente, os espaços tangentes das regiões P_i e P_j são obtidos pela aplicação do PCA nas matrizes de covariância Σ_i e Σ_j . Sejam $U_i = [\vec{u}_{i1}, \vec{u}_{i2}, \dots, \vec{u}_{im}]$ e $U_j = [\vec{u}_{j1}, \vec{u}_{j2}, \dots, \vec{u}_{jm}]$ as bases ortogonais que definem os espaços tangentes dos pontos \vec{x}_i e \vec{x}_j , respectivamente. Então uma aproximação ingênua para as curvaturas principais no vértice (v_i, v_j) é dado por:

$$K_{ij}^{(l)} = \|\vec{u}_{il} - \vec{u}_{jl}\| \quad \text{para } l = 1, 2, \dots, m$$

Este trabalho propõe a utilização de $\|K_{ij}\|$ como peso da aresta (v_i, v_j) no grafo de distâncias, trocar a distância Euclidiana extrínseca entre \vec{x}_i e \vec{x}_j pela norma do vetor de curvaturas principais.

3.1.2 Aproximação da curvatura por geometria diferencial

Foi mostrado anteriormente em 2.2.1.2 que a curvatura principal pode ser computada exatamente por:

$$K(s) = \frac{\|\vec{\alpha}'(t) \times \vec{\alpha}''(t)\|}{\|\vec{\alpha}'(t)\|^3} \quad (17)$$

Usando a mesma notação do método anterior, sejam $U_i = [\vec{u}_{i1}, \vec{u}_{i2}, \dots, \vec{u}_{im}]$ e $U_j = [\vec{u}_{j1}, \vec{u}_{j2}, \dots, \vec{u}_{jm}]$ as bases ortogonais que definem os espaços tangentes dos pontos \vec{x}_i e \vec{x}_j . Podemos aproximar as variações de cada um dos vetores no espaço tangente por:

$$N = [\vec{n}_1, \vec{n}_2, \dots, \vec{n}_m] = [\vec{u}_{i1} - \vec{u}_{j1}, \vec{u}_{i2} - \vec{u}_{j2}, \dots, \vec{u}_{im} - \vec{u}_{jm}]$$

O numerador de (17) para o l -ésimo vetor tangente é calculado por:

$$\|\vec{u}_l \times \vec{n}_l\| = \|\vec{u}_l\| \|\vec{n}_l\| \text{sen}(\theta_l) \quad \text{para } l = 1, 2, \dots, m$$

onde o ângulo θ é obtido da seguinte forma:

$$\theta_l = \arccos \left(\frac{\vec{u}_l^T \vec{n}_l}{\|\vec{u}_l\| \|\vec{n}_l\|} \right) \quad \text{para } l = 1, 2, \dots, m$$

O denominador de (17) pode ser calculado diretamente por:

$$\|\vec{u}_l\|^3 = \left(\sqrt{\vec{u}_l(1)^2 + \vec{u}_l(2)^2 + \dots + \vec{u}_l(m)^2} \right)^3$$

Finalmente, os m componentes do vetor de curvaturas principais é computado por:

$$K_{ij}^{(l)} = \frac{\|\vec{u}_l\| \|\vec{n}_l\| \text{sen}(\theta_l)}{\|\vec{u}_l\|^3} \quad \text{para } l = 1, 2, \dots, m \quad (18)$$

Este trabalho propõe substituir a distância Euclidiana entre os pontos pela norma do vetor de curvaturas principais na aresta (v_i, v_j) do grafo de distâncias.

4 Experimentos e Resultados

4.1 Materiais e métodos

Esta seção são apresentadas as principais fontes de dados utilizadas para validar o método proposto, e informações sobre as ferramentas e bibliotecas utilizadas no desenvolvimento deste trabalho.

4.1.1 Dados

Foram utilizados uma gama de conjuntos de dados disponíveis disponibilizados publicamente no OpenML ([VANSCHOREN et al., 2013](#)), de modo que exista uma grande variedade no número de amostras, atributos e classes dos diferentes conjuntos de dados utilizados. Os detalhes sobre cada um dos conjuntos de dados utilizados podem ser vistos na Tabela 3, no Apêndice B.

4.1.2 Software

A linguagem escolhida para a implementação do algoritmo proposto foi Python devido à existência de diversas bibliotecas de *machine learning* e programação científica disponíveis para uso, além da popularidade da linguagem, especificamente na comunidade de *machine learning*.

As bibliotecas SciPy e NumPy foram utilizadas para operações matemáticas de alto nível e cálculos matriciais necessários na implementação deste trabalho. A biblioteca Scikit-Learn disponibiliza diversos algoritmos conhecidos de classificação e redução de dimensionalidade, estes foram utilizados para validar o modelo desenvolvido e realizar experimentos comparativos de análise de performance, respectivamente.

4.2 Descrição dos Experimentos

Para avaliar a eficácia do método proposto por este trabalho, foram realizados testes comparativos com outros métodos de redução de dimensionalidade existentes na literatura. As métricas de eficácia escolhidas foram: 1) a qualidade da representação dos dados em baixas dimensões após a redução de dimensionalidade; e 2) a acurácia média obtida por diferentes classificadores supervisionados treinados e aplicados nesses dados em baixas dimensões. Todos os experimentos realizados nesse trabalho foram para o caso 2-D, ou seja, todos os conjuntos de dados foram reduzidos à duas dimensões por diferentes métodos de redução de dimensionalidade.

O coeficiente de silhueta é uma medida de quão similares são as amostras pertencentes à um mesmo agrupamento e quão separadas estão de objetos pertencentes à outros agrupamentos. Após a aplicação do K-ISOMAP em cada conjunto de dados, o coeficiente de silhueta foi utilizado para mensurar a qualidade da representação dos dados em 2-D resultante. Os agrupamentos foram gerados à partir do *ground truth*, utilizando os rótulos dos dados para criar agrupamentos artificialmente, de modo que cada classe seja considerada como um agrupamento distinto, e coeficiente de silhueta foi calculado utilizando uma métrica Euclidiana de distância.

A acurácia média de classificação foi medida utilizando 8 diferentes classificadores supervisionados após reduzir a dimensionalidade dos dados para 2-D. Os classificadores utilizados foram: KNN, Support vector machine (SVM), Decision Trees (DT), Naive Bayes (NB), Quadratic Discriminant Analysis (QDA), Multilayer Perceptron (MLP), Random Forest Classifier (RFC) e Gaussian Process Classifier (GPC).

É esperado que essas duas medidas de eficácia exibirão um aumento significativo caso a distância intrínseca dos dados seja aprendida corretamente e, considerando a aplicação do método em conjuntos de dados distintos, é esperado que a média dessas medidas também irá apresentar um aumento significativo.

Os resultados de coeficiente de silhueta e da acurácia média de classificação obtidos pelo K-ISOMAP para cada um dos *datasets* foi comparado com os resultados obtidos utilizando outros métodos de redução de dimensionalidade conhecidos, nomeadamente: PCA, Kernel Principal Component Analysis (KPCA), ISOMAP, LLE, Laplacian Eigenmaps, t-Distributed Stochastic Neighbor Embedding (t-SNE) e Uniform Manifold Approximation and Projection (UMAP).

4.3 Resultados obtidos

Os resultados completos do primeiro conjunto de experimentos estão dispostos na Tabela 5, no Apêndice B. Dos 58 conjuntos de dados utilizados, uma das variações do método proposto neste trabalho obteve o melhor resultado para coeficiente de silhueta em 32 deles, representando aproximadamente 55% dos casos. Observando os valores da média e mediana, pode-se concluir que o método proposto obteve resultados melhores do que métodos atuais como t-SNE e UMAP nestes *datasets*.

Para examinar se os resultados obtidos pelo método proposto são estatisticamente superiores foi aplicado o teste de Kruskal-Wallis. Com um nível de significância $\alpha = 0.01$, concluímos que existem evidências claras de que a hipótese nula de que todos os grupos são idênticos pode ser rejeitada ($p = 0.0003$). No entanto, o teste de Kruskal não nos permite saber quais grupos são significativamente diferentes. Para isso foi aplicado o teste de Nemenyi, um teste *post-hoc* de comparação múltipla onde são realizados comparações

par a par.

Segundo o teste de Nemenyi, para um nível de significância $\alpha = 0.01$, existem evidências fortes de que ambas as variantes A e B do método proposto neste trabalho produziram coeficientes de silhueta significativamente mais altos do que os outros métodos utilizados para comparação, como visto na Tabela 1 abaixo. No entanto, não existem evidências de que os métodos K-ISOMAP-A e K-ISOMAP-B sejam diferentes com base nos coeficientes de silhueta obtidos nestes conjuntos de dados ($p = 0.812$).

Tabela 1 – p -valores resultantes do teste de Nemenyi para coeficientes de silhueta.

	K-ISOMAP-A	K-ISOMAP-B
PCA	9.45×10^{-12}	4.79×10^{-11}
KPCA	2.05×10^{-10}	9.38×10^{-10}
ISOMAP	9.92×10^{-14}	2.29×10^{-10}
LLE	4.79×10^{-11}	5.82×10^{-13}
Lap. Eig.	2.17×10^{-09}	9.10×10^{-9}
t-SNE	6.08×10^{-9}	2.44×10^{-8}
UMAP	4.36×10^{-8}	1.62×10^{-7}

Para o segundo conjunto de experimentos, foram separados 50% de cada conjunto de dados para utilizar no treinamento dos 8 classificadores supervisionados distintos: **KNN** ($K = 7$), **SVM** (linear), **DT**, **NB**, **QDA** (hipótese Gaussiana), **MLP**, **RFC** e **GPC**. Após isso, esses modelos treinados foram utilizados para classificar os outros 50% das amostras, e a média dos *scores* obtidos por cada classificador foi utilizada como parâmetro de avaliação da redução de dimensionalidade para classificação supervisionada. Os resultados completos desses experimentos podem ser vistos na Tabela 4, disposta no Apêndice B. Nestes experimentos, as variantes do K-ISOMAP apresentaram o maior valor de acurácia média em 48 dos 58 *datasets*, o que corresponde à aproximadamente 82% dos casos.

Novamente foi utilizado o teste de Kruskal-Wallis, para testar se as acurácias médias na classificação supervisionada obtidas pelo método proposto por este trabalho são estatisticamente superiores àquelas obtidas por outros métodos da literatura. Para um nível de significância de $\alpha = 0.01$, existem evidências fortes contra a hipótese nula de que todos os grupos são idênticos ($p = 3.31 \times 10^{-6}$).

Para verificar quais grupos são significativamente distintos, foi realizado o teste *post-hoc* de Nemenyi. Para um nível de significância de $\alpha = 0.01$, o teste nos diz que existem fortes evidências de que as duas variantes do K-ISOMAP produziram acurácias de classificação significativamente mais altas do que outros métodos de redução de dimensionalidade, conforme visto na Tabela 2 abaixo. Concluímos que não existem diferenças significativas entre o K-ISOMAP-A e o K-ISOMAP-B em termos de acurácia de classificação, visto que seus resultados foram semelhantes ($p = 0.812$).

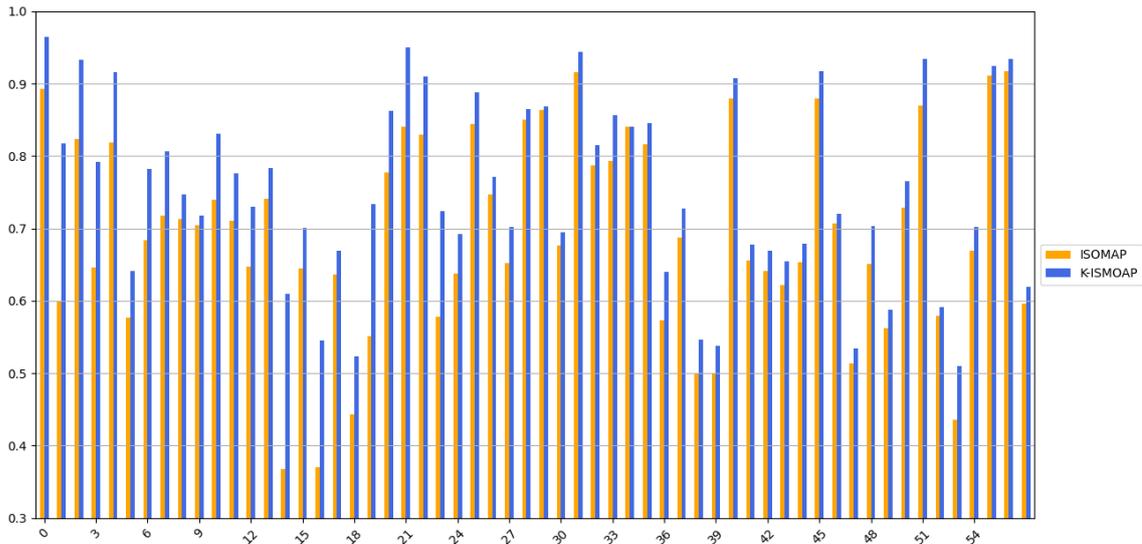
Uma comparação entre o método tradicional ISOMAP e o método proposto K-ISOMAP revela que, para estes conjuntos de dados, a utilização de uma métrica que

Tabela 2 – p -valores resultantes do teste de Nemenyi para acurácias médias de classificação.

	K-ISOMAP-A	K-ISOMAP-B
PCA	8.40×10^{-12}	1.56×10^{-12}
KPCA	1.45×10^{-13}	2.37×10^{-14}
ISOMAP	2.42×10^{-13}	4.01×10^{-14}
LLE	$< 10^{-17}$	$< 10^{-17}$
Lap. Eig.	$< 10^{-17}$	$< 10^{-17}$
t-SNE	7.82×10^{-05}	2.82×10^{-05}
UMAP	3.40×10^{-06}	1.04×10^{-06}

incorpora o cálculo de curvatura proporcionou um aumento médio de 12% na performance de classificação. As maiores diferenças nos resultados foram observadas nos *datasets* parity5, aids, e prnn_crabs, com melhorias de 65%, 36%, e 47%, respectivamente. A Figura 5 exibe a comparação da acurácia média de classificação obtida em cada um dos conjuntos de dados após a aplicação dos métodos de redução de dimensionalidade.

Figura 5 – Comparação da acurácia de classificação entre o ISOMAP e o K-ISOMAP em 58 datasets após a redução de dimensionalidade.



Como mencionado na seção 2.3.3, é de conhecimento que uma das limitações dos métodos globais de *manifold learning* é o *out-of-sample problem*, que exige que o método seja aplicado novamente na totalidade dos dados à cada nova amostra inserida. Foi mostrado brevemente na seção 2.3.3.1 deste trabalho e discutido em detalhes por Choi e Choi (2007) que o método Kernel PCA se torna ISOMAP quando a matriz de kernel é definida como $-\frac{1}{2}D$, onde D é a matriz de distâncias geodésicas. Portanto, considerando essa relação, é possível contornar o problema de *out-of-sample* ao aplicar o K-ISOMAP, utilizando a mesma estratégia de projeção do Kernel PCA.

Uma limitação do método proposto é escolha do melhor valor para o parâmetro K , que é o número de vizinhos. Como o número de vizinhos define a região que será utilizada para estimar o espaço tangente em cada amostra, os resultados de acurácia e coeficiente

de silhueta exibiram uma alta sensibilidade em relação à esse parâmetro. Os experimentos mostraram que certos valores de K podem resultar na estimação errônea da curvatura, e em casos extremos pode resultar em uma performance pior que a do ISOMAP tradicional.

Neste trabalho, a seguinte estratégia foi utilizada: para cada conjunto de dados, foi construído um gráfico KNN para cada valor de K no intervalo $[2,30]$. O modelo que apresentasse na maior acurácia de classificação dentre todos os valores foi escolhido como melhor. Os rótulos das classes foram utilizados para selecionar o melhor modelo, porém a redução de dimensionalidade é realizada de forma completamente não supervisionada. Este método de seleção do valor de K poderia ser substituído por outros métodos da literatura para estimação de um número ótimo de vizinhos.

5 Conclusões

Este trabalho abordou o desafio da aprendizagem de métricas não supervisionada para a extração de atributos não-lineares de dados. Foi proposto um método baseado no ISOMAP que incorpora a curvatura intrínseca dos dados no grafo KNN. Em outras palavras, as arestas do grafo KNN são distorcidas conforme a curvatura, para que as distâncias no grafo se aproximem da distância geodésica do *manifold* em que os dados se encontram.

Resultados preliminares obtidos experimentalmente mostram que o método proposto exibe uma performance melhor do que o ISOMAP, e pode se mostrar uma alternativa viável à diversos outros métodos de redução de dimensionalidade devido aos resultados experimentais, onde vimos que os atributos não-lineares produzidos pelo K-ISOMAP podem ser mais discriminativos para classificadores supervisionados do que atributos produzidos por outros métodos.

Apesar dos resultados de coeficiente de silhueta sugerirem que a qualidade da representação dos dados em baixas dimensões gerados pelo K-ISOMAP é superior àquela gerada por outros métodos de redução de dimensionalidade, vale salientar que os agrupamentos foram feitos de forma sintética utilizando as classes dos dados. Resultados mais relevantes requerem uma abordagem semelhante ao que fizemos para a acurácia de classificação: aplicar diversos algoritmos de *clustering*, computar o coeficiente de silhueta em cada um deles e calcular a média desse valor.

Existem indícios de que a performance do K-ISOMAP pode ser atribuída à abordagem utilizada de estimação de similaridade entre amostras, visto que métodos baseados *patches* tendem a ser mais robustas ao se deparar com *outliers* e ruído nos conjunto de dados, quando comparados com métodos que utilizam distância Euclidiana entre pontos. Trabalhos futuros podem buscar evidências definitivas analisando os resultados do ISOMAP original e do método proposto em *datasets* com altos níveis de ruído.

Por ser um método de aprendizado de métrica não supervisionado, o K-ISOMAP pode ser incorporado em diferentes algoritmos de *manifold learning* baseados em grafos como, por exemplo, LLE, Laplacian Eigenmaps e Locality Preserving Projections (LPP), incorporando a estimativa da curvatura dos dados nas funções de distância intrínseca. Trabalhos futuros podem explorar o uso de outras variantes do PCA na estimação dos espaços tangentes, como Robust PCA, Sparse PCA ou Parametric PCA. Ainda outra possibilidade é a utilização de outros métodos de estimação de curvatura baseados em espaços tangente, como o operador de Weingarten.

Referências

- BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, v. 35, p. 1798–1828, 08 2013. Citado na página 4.
- BERNSTEIN, M. et al. *Graph Approximations to Geodesics on Embedded Manifolds*. 2000. Citado na página 13.
- BORG, I.; GROENEN, P. *Modern Multidimensional Scaling: Theory and Applications*. [S.l.]: Springer, 2005. Citado na página 11.
- CAO, Y. et al. Efficient weingarten map and curvature estimation on manifolds. *Machine Learning*, Springer Science and Business Media LLC, v. 110, n. 6, p. 1319–1344, May 2021. ISSN 1573-0565. Disponível em: <<http://dx.doi.org/10.1007/s10994-021-05953-4>>. Nenhuma citação no texto.
- CARMO, M. P. do. *Differential geometry of curves and surfaces*. [S.l.]: Prentice Hall, 1976. I-VIII, 1-503 p. ISBN 978-0-13-212589-5. Citado na página 5.
- CHOI, H. H.; CHOI, S. Robust kernel isomap. *Pattern Recognition*, v. 40, p. 853–862, 03 2007. Citado 2 vezes nas páginas 15 e 23.
- COX, T.; COX, M. *Multidimensional Scaling*. Chapman and Hall/CRC, 2000. Disponível em: <<https://doi.org/10.1201/9780367801700>>. Citado 2 vezes nas páginas 11 e 12.
- GAN, Q.; SHEN, F.; ZHAO, J. An extended isomap for manifold topology learning with soinn landmarks. In: *2014 22nd International Conference on Pattern Recognition*. [S.l.: s.n.], 2014. p. 1579–1584. Citado na página 16.
- HAM, J. et al. A kernel view of the dimensionality reduction of manifolds. In: . [S.l.: s.n.], 2004. Citado na página 15.
- ITO, K. *Encyclopedic dictionary of mathematics*. Cambridge, Mass: MIT Press, 1987. ISBN 978-0262090261. Citado na página 9.
- LIMA, E. *Variiedades diferenciáveis*. Rio de Janeiro: IMPA, 2007. ISBN 9788524402678. Citado 3 vezes nas páginas 6, 7 e 8.
- MCINNES, L.; HEALY, J.; MELVILLE, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. Citado na página 4.
- NETO, A. C.; LEVADA, A. Isomap-kl: a parametric approach for unsupervised metric learning. In: *Anais do XXXIII Conference on Graphics, Patterns and Images*. Porto Alegre, RS, Brasil: SBC, 2020. p. 57–64. Disponível em: <<https://sol.sbc.org.br/index.php/sibgrapi/article/view/14108>>. Citado na página 16.
- O’NEILL, B. Chapter 2 - frame fields. In: O’NEILL, B. (Ed.). *Elementary Differential Geometry (Second Edition)*. Second edition. Boston: Academic Press, 2006. p. 43–99. ISBN 978-0-12-088735-4. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780120887354500067>>. Citado na página 5.

- 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, 07 2017. Citado na página 3.
- SAXENA, A.; GUPTA, A.; MUKERJEE, A. Non-linear dimensionality reduction by locally linear isomaps. In: . [S.l.: s.n.], 2004. v. 3316, p. 1038–1043. ISBN 978-3-540-23931-4. Citado na página 16.
- SCHOLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Kernel principal component analysis. In: *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*. [S.l.]: MIT Press, 1999. p. 327–352. Citado na página 10.
- SHIFRIN, T. *Differential Geometry: A First Course in Curves and Surfaces*. [S.l.]: University of Georgia, 2006. Citado na página 5.
- SILVA, V. de; TENENBAUM, J. B. Global versus local methods in nonlinear dimensionality reduction. In: *NIPS*. [S.l.: s.n.], 2002. Citado na página 16.
- SINGER, A.; WU, H.-T. Vector diffusion maps and the connection laplacian. *Communications on pure and applied mathematics*, v. 65 8, 2012. Citado na página 10.
- TENENBAUM, J. B.; SILVA, V. de; LANGFORD, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, v. 290, n. 5500, p. 2319–2323, 2000. Citado na página 13.
- TU, L. *An introduction to manifolds*. New York: Springer, 2011. ISBN 978-1-4419-7400-6. Citado na página 7.
- VANSCHOREN, J. et al. Openml: Networked science in machine learning. *SIGKDD Explorations*, ACM, New York, NY, USA, v. 15, n. 2, p. 49–60, 2013. Disponível em: <<http://doi.acm.org/10.1145/2641190.2641198>>. Citado na página 20.
- WANG, F.; SUN, J. Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, v. 29, p. 534–564, 03 2014. Citado na página 4.
- WHITNEY, H. Differentiable manifolds. *Annals of Mathematics, Second Series, Vol. 37, No. 3 (Jul., 1936), pp. 645-680*, 1936. Citado na página 8.

APÊNDICE A – Detalhes matemáticos

Neste apêndice se encontram definições e explicações detalhadas dos tópicos matemáticos decorridos no texto principal, para aqueles que estiverem interessados.

A.1 Curvas, comprimento de arco e curvatura

Definição 1. *Uma curva paramétrica em \mathbb{R}^3 é uma função diferenciável $\vec{\alpha} : [a, b] \rightarrow \mathbb{R}^3$ definida em um intervalo $[a, b]$ e cuja imagem existe em \mathbb{R}^3 . Então uma curva paramétrica é escrita como*

$$\vec{\alpha}(t) = (x(t), y(t), z(t))$$

onde $x(t), y(t)$ e $z(t)$ são funções diferenciáveis $f : [a, b] \rightarrow \mathbb{R}$ denominadas funções de coordenada Euclidiana.

Definição 2. *O vetor tangente de uma curva paramétrica $\vec{\alpha} : t \in [a, b] \rightarrow \mathbb{R}^3$ é definida como:*

$$\vec{\alpha}'(t) = (x'(t), y'(t), z'(t))$$

onde $f'(t)$ é a primeira derivada de $f(t)$.

Sejam r e s dois pontos no qual $\alpha(t)$ está definida, $r, s \in [a, b]$. Então o comprimento de arco entre r e s pode ser aproximado pela diferença entre os valores da curva nesses dois pontos, um vetor $\vec{S} = \alpha(s) - \alpha(r)$, como visto na Figura 1.

$$S(t) = \|\alpha(s) - \alpha(r)\|$$

$$S(t) = \sqrt{(x(s) - x(r))^2 + (y(s) - y(r))^2 + (z(s) - z(r))^2}$$

Quanto menor for o intervalo $[r, s]$, melhor será a aproximação do comprimento de arco. Igualmente, se aumentarmos o número de subdivisões nesse intervalo e calcular o comprimento de arco para cada uma das subdivisões, a soma de todos esses resultados será uma melhor aproximação do comprimento de arco no intervalo $[r, s]$.

Denominando Δt como a espessura de cada subdivisão, a curvatura por ser aproximada por:

$$S(t) \approx \sum_{t=r}^s \sqrt{(x(t + \Delta t) - x(t))^2 + (y(t + \Delta t) - y(t))^2 + (z(t + \Delta t) - z(t))^2}$$

A equação acima pode ser reescrita como:

$$S(t) \approx \sum_{t=r}^s \sqrt{\left(\frac{x(t+\Delta t) - x(t)}{\Delta t}\right)^2 + \left(\frac{y(t+\Delta t) - y(t)}{\Delta t}\right)^2 + \left(\frac{z(t+\Delta t) - z(t)}{\Delta t}\right)^2} \Delta t \quad (\text{A.1.1})$$

Da definição de derivada e uma função, sabemos que:

$$f'(t) = \lim_{\Delta t \rightarrow 0} \frac{f(t+\Delta t) - f(t)}{\Delta t}$$

Da definição de integral de uma função, sabemos que:

$$\int_a^b f(t) dt = \lim_{\Delta t \rightarrow 0} \sum_{t=a}^b f(t+\Delta t) \Delta t$$

Portanto, no caso limite onde $\Delta t \rightarrow 0$, a aproximação da Equação (A.1.1) se torna exata:

$$\begin{aligned} S(t) &= \int_r^s \sqrt{x'(t)^2 + t'(t)^2 + z'(t)^2} dt \\ &= \int_r^s \sqrt{\vec{\alpha}'(t)^T \vec{\alpha}'(t)} dt \\ &= \int_r^s \|\vec{\alpha}'(t)\| dt \end{aligned} \quad (\text{A.1.2})$$

onde $\|\vec{\alpha}'(t)\|$ é a norma do vetor tangente em t .

Portanto o comprimento de arco de uma curva depende do vetor tangente. O vetor tangente em um ponto da curva também é chamado de vetor de velocidade de α em t . Como a velocidade de uma curva é definida pelas derivadas de suas funções de coordenada, parametrizações diferentes de uma mesma curva podem exibir vetores de velocidade diferentes em um mesmo ponto.

Por isso em diversas aplicações é preferível utilizar a parametrização de uma curva pelo seu comprimento de arco, de modo que a velocidade em que nos movemos ao longo da curva seja sempre constante e unitária.

$$\vec{\alpha}'(s) = \frac{\vec{\alpha}'(t)}{\|\vec{\alpha}'(t)\|} = \vec{T}(t) \quad (\text{A.1.3})$$

Utilizando a parametrização por comprimento de arco, os vetores tangente, normal e binormal são normalizados. Com isso, esses vetores formam não somente uma base ortogonal, mas uma base ortonormal em \mathbb{R}^3 .

Definição 3. *O vetor normal, que é a segunda derivada de uma curva $\vec{\alpha}$ em um ponto t , é ortogonal ao vetor tangente.*

$$\vec{N}(t) \perp \vec{T}(t)$$

Note que, segundo a Equação (A.1.3), $\vec{\alpha}'(s)$ é um vetor normalizado, o que nos leva à:

$$\|\vec{\alpha}'(s)\|^2 = x'(s)^2 + y'(s)^2 + z'(s)^2 = 1$$

Derivando os dois lados em relação à s , obtemos:

$$x'(s)x''(s) + y'(s)y''(s) + z'(s)z''(s) = 0$$

que pode ser reescrito como:

$$\vec{\alpha}'(s)^T \vec{\alpha}''(s) = 0$$

$$\vec{T}'(s)^T \vec{N}''(s) = 0$$

Portanto os vetores tangente e normal são ortogonais, e sabendo que o vetor tangente é dado por $\vec{\alpha}'(s) = \vec{T}(t)$, a sua variação é dada por $\vec{\alpha}''(s) = \vec{T}'(t)$, que é o vetor normal. Para normalizar o vetor normal, definimos:

$$\vec{N}(t) = \frac{\vec{T}'(t)}{\|\vec{T}'(t)\|} = \frac{\vec{\alpha}''(s)}{\|\vec{\alpha}''(s)\|} \quad (\text{A.1.4})$$

Definição 4. A curvatura de uma curva $\vec{\alpha}$ em um ponto s é a variação instantânea do vetor tangente normalizado neste ponto:

$$K(s)\|\vec{\alpha}''(s)\| = \|\vec{T}'(t)\|$$

Para expressar a curvatura em função de termos conhecidos, note que podemos escrever a Equação (A.1.3) como:

$$\vec{\alpha}'(t) = \|\vec{\alpha}'(t)\|\vec{\alpha}'(s) \quad (\text{A.1.5})$$

e aplicando o teorema fundamental do cálculo na definição de comprimento de arco da Equação (A.1.2), obtemos:

$$s(t) = \int_r^s \|\vec{\alpha}'(t)\| dt \quad \implies \quad \frac{ds}{dt} = \|\vec{\alpha}'(t)\| \quad (\text{A.1.6})$$

Substituindo a Equação (A.1.6) em (A.1.5):

$$\vec{\alpha}'(t) = \frac{ds}{dt} \vec{\alpha}'(s)$$

Derivando ambos os lados em relação à t obtemos, pela regra do produto:

$$\begin{aligned} \vec{\alpha}''(t) &= \frac{d^2s}{dt^2} \vec{\alpha}'(s) + \frac{ds}{dt} \frac{d}{ds} \vec{\alpha}'(s) \frac{ds}{dt} \\ &= \frac{d^2s}{dt^2} \vec{\alpha}'(s) + \left(\frac{ds}{dt}\right)^2 \vec{\alpha}''(s) \end{aligned} \quad (\text{A.1.7})$$

Sabendo que os vetores $\vec{\alpha}'(s)$ e $\vec{\alpha}'(t)$ são paralelos, e portanto o produto vetorial entre eles é zero, calculamos o produto vetorial com $\vec{\alpha}'(t)$ nos dois lados da equação anterior, e obtemos:

$$\begin{aligned} \vec{\alpha}'(t) \times \vec{\alpha}''(t) &= \vec{\alpha}'(t) \times \left[\frac{d^2s}{dt^2} \vec{\alpha}'(s) + \left(\frac{ds}{dt}\right)^2 \vec{\alpha}''(s) \right] \\ &= \left(\frac{ds}{dt}\right)^2 [\vec{\alpha}'(t) \times \vec{\alpha}''(s)] \end{aligned} \quad (\text{A.1.8})$$

Usando a equação (A.1.6), podemos escrever:

$$\vec{\alpha}'(t) \times \vec{\alpha}''(t) = \|\vec{\alpha}'(t)\|^2 [\vec{\alpha}'(t) \times \vec{\alpha}''(s)] \quad (\text{A.1.9})$$

Tirando a norma de ambos os lados da equação, temos:

$$\|\vec{\alpha}'(t) \times \vec{\alpha}''(t)\| = \|\vec{\alpha}'(t)\|^2 \|\vec{\alpha}''(s)\| \text{sen}(\theta)$$

onde θ é o ângulo entre $\vec{\alpha}'(t)$ e $\vec{\alpha}''(s)$. Como visto anteriormente, esses vetores são ortogonais e portanto $\text{sen}(\theta) = 1$.

Finalmente chegamos na expressão para calcular a curvatura $K(t)$ em um ponto da curva à partir de informações conhecidas:

$$K(s) = \|\vec{\alpha}''(s)\| = \frac{\|\vec{\alpha}'(t) \times \vec{\alpha}''(t)\|}{\|\vec{\alpha}'(t)\|^3} \quad (\text{A.1.10})$$

A.2 Equações de Frenet-Serret

Sejam $\vec{T}(t), \vec{N}(t)$ os vetores tangente e normal, respectivamente, ambos com norma unitária. O vetor $\vec{B}(t) = \vec{T}(t) \times \vec{N}(t)$ é denominado vetor binormal. O conjunto formado por esses três vetores define uma base ortonormal em \mathbb{R}^3 para cada ponto onde a curva $\vec{\alpha}(t)$ está definida. As equações de Frenet-Serret nos permitem expressar cada um desses vetores apenas em função dos dois outros vetores que compõem a base, como veremos à seguir.

A.2.1 Variação no vetor tangente

A variação no vetor tangente entre dois pontos próximos já foi vista de forma indireta na seção A.1 desse apêndice. Reescrevendo a equação (A.1.4), temos:

$$\vec{T}'(t) = \|\vec{T}'(t)\| \vec{N}(t) = K(t) \vec{N}(t) \quad (\text{A.2.1})$$

o que nos mostra que a variação no vetor tangente se dá unicamente em função do vetor normal, pela magnitude da curvatura $K(t)$. Portanto para analisar as mudanças do vetor tangente ao longo do

A.2.2 Variações no vetor binormal

A variação no vetor binormal deve ser obtida em função dos três componentes do Triedro de Frenet, portanto escrevemos:

$$\vec{B}'(t) = c_1\vec{T}(t) + c_2\vec{N}(t) + c_3\vec{B}(t)$$

Como o vetor binormal possui norma unitária, sabemos que $\vec{B}(t)^T\vec{B}(t) = 1$. Derivando em relação a t , podemos escrever:

$$\vec{B}'(t)^T\vec{B}(t) + \vec{B}(t)^T\vec{B}'(t) = 0$$

$$2\vec{B}(t)^T\vec{B}'(t) = 0$$

e obtemos que $\vec{B}'(t)$ é ortogonal à $\vec{B}(t)$, e portanto $c_3 = 0$. Por definição, sabemos que os vetores tangente e binormal são ortogonais, portanto $\vec{B}(t)^T\vec{T}(t) = 0$. Derivando ambos os lados, temos:

$$\vec{B}'(t)^T\vec{T}(t) + \vec{B}(t)^T\vec{T}'(t) = 0$$

Substituindo o valor de $\vec{T}'(t)$ obtido na equação (A.2.1), chegamos em:

$$\vec{B}'(t)^T\vec{T}(t) + K(t)\vec{B}(t)^T\vec{N}(t) = 0$$

Como os vetores $\vec{B}(t)$ e $\vec{N}(t)$ são ortogonais, sabemos que $\vec{B}(t)^T\vec{N}(t) = 0$. Com isso, obtemos o resultado:

$$\vec{B}'(t)^T\vec{T}(t) = 0$$

o que nos mostra que $c_1 = 0$. Como $\vec{B}'(t)$ é ortogonal aos vetores $\vec{T}(t)$ e $\vec{B}(t)$, então deve ser paralelo ao vetor normal $\vec{N}(t)$:

$$\vec{B}'(t) = c_2\vec{N}(t) \tag{A.2.2}$$

Por convenção, $c_2 = -\tau(t)$ é o negativo da torção da curva, que é uma medida do quão rápido a curva se desloca do plano definido pelos vetores $\vec{T}(t)$ e $\vec{N}(t)$.

A.2.3 Variações no vetor normal

Similarmente ao vetor binormal, podemos expressar a variação do vetor normal $\vec{N}'(t)$ em função dos componentes do Triedro de Frenet:

$$\vec{N}'(t) = c_1\vec{T}(t) + c_2\vec{N}(t) + c_3\vec{B}(t)$$

e à partir usando o mesmo raciocínio podemos ver que $\vec{N}'(t)^T \vec{N}(t) = 0$, e portanto $c_2 = 0$.

Sabendo que os vetores tangente, normal e binormal formam uma base ortogonal, temos:

$$\vec{N}(t)^T \vec{T}(t) = 0 \quad (\text{A.2.3})$$

$$\vec{N}(t)^T \vec{B}(t) = 0 \quad (\text{A.2.4})$$

Derivando a equação (A.2.3) em relação à t nos leva a:

$$\begin{aligned} \vec{N}'(t)^T \vec{T}(t) + \vec{N}(t)^T \vec{T}'(t) &= 0 \\ \vec{N}'(t)^T \vec{T}(t) + K(t) \vec{N}(t)^T \vec{N}(t) &= 0 \\ \vec{N}'(t)^T \vec{T}(t) &= -K(t) \end{aligned} \quad (\text{A.2.5})$$

Portanto obtemos o resultado de que a variação do vetor normal na direção do vetor tangente é dado pelo oposto da curvatura, o que implica que $c_1 = -K(t)$.

Realizando o mesmo procedimento de derivada na equação (A.2.4) e substituindo o valor obtido da equação (A.2.2), temos:

$$\begin{aligned} \vec{N}'(t)^T \vec{B}(t) + \vec{N}(t)^T \vec{B}'(t) &= 0 \\ \vec{N}'(t)^T \vec{B}(t) - \tau(t) \vec{N}(t)^T \vec{N}(t) &= 0 \\ \vec{N}'(t)^T \vec{B}(t) &= \tau(t) \end{aligned} \quad (\text{A.2.6})$$

que nos diz que a projeção da variação do vetor normal no vetor binormal é a torção, e que $c_3 = \tau(t)$.

Portanto, as equações de Frenet-Serret são definidas por:

$$\begin{aligned} \vec{T}'(t) &= K(t) \vec{N}(t) \\ \vec{N}'(t) &= -K(t) \vec{N}(t) + \tau \vec{B}(t) \\ \vec{B}'(t) &= -\tau \vec{N}(t) \end{aligned} \quad (\text{A.2.7})$$

A.3 Desenvolvimento matemático do MDS

Considere que a matriz de coordenadas de n pontos em um espaço Euclidiano de p dimensões seja dada por $X = \{\vec{x}_r\}$, onde $r = 1, 2, \dots, n$ e $\vec{x}_r = (x_{r1}, \dots, x_{rp})^T$.

Seja $D = \{d_{rs}^2\}$ a matriz de distâncias entre esses pontos, e $B = \{b_{rs}\}$ a matriz de produtos internos, para $r, s = 1, 2, \dots, n$. Para dois pontos arbitrários \vec{x}_r e \vec{x}_s , temos:

$$d_{rs}^2 = \|\vec{x}_r - \vec{x}_s\|^2 = (\vec{x}_r - \vec{x}_s)^T (\vec{x}_r - \vec{x}_s) \quad (\text{A.3.1})$$

$$b_{rs} = \vec{x}_r^T \vec{x}_s \quad (\text{A.3.2})$$

Como o MDS preserva a distância entre os pontos, existem infinitas soluções possíveis para o problema, diferentes entre si apenas por uma translação arbitrária. Para obter uma solução única, iremos supor que a nossa matriz de coordenadas X possui média zero.

$$\sum_{r=1}^n \vec{x}_{ri} = 0 \quad (i = 1, \dots, p) \quad (\text{A.3.3})$$

Expandindo a equação (A.3.1), temos:

$$d_{rs}^2 = \vec{x}_r^T \vec{x}_r + \vec{x}_s^T \vec{x}_s - 2\vec{x}_r^T \vec{x}_s \quad (\text{A.3.4})$$

A partir disso podemos calcular a média de uma coluna arbitrária de D :

$$\begin{aligned} \frac{1}{n} \sum_{r=1}^n d_{rs}^2 &= \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r + \frac{1}{n} \sum_{r=1}^n \vec{x}_s^T \vec{x}_s - \frac{1}{n} \sum_{r=1}^n 2\vec{x}_r^T \vec{x}_s \\ &= \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r + \frac{n}{n} \vec{x}_s^T \vec{x}_s - 2\vec{x}_s \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \xrightarrow{0} \\ &= \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r + \vec{x}_s^T \vec{x}_s \end{aligned} \quad (\text{A.3.5})$$

Analogamente para a média de uma linha arbitrária de D :

$$\frac{1}{n} \sum_{s=1}^n d_{rs}^2 = \vec{x}_r^T \vec{x}_r + \frac{1}{n} \sum_{s=1}^n \vec{x}_s^T \vec{x}_s \quad (\text{A.3.6})$$

Calculando a média de todos os elementos da matriz D , temos:

$$\begin{aligned} \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n d_{rs}^2 &= \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n \vec{x}_r^T \vec{x}_r + \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n \vec{x}_s^T \vec{x}_s - \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n 2\vec{x}_r^T \vec{x}_s \\ &= \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r + \frac{1}{n} \sum_{s=1}^n \vec{x}_s^T \vec{x}_s - \frac{2}{n^2} \sum_{r=1}^n \left(\vec{x}_r^T \sum_{s=1}^n \vec{x}_s \right) \xrightarrow{0} \\ &= \frac{2}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r \end{aligned} \quad (\text{A.3.7})$$

Podemos reescrever a equação (A.3.2) como:

$$b_{rs} = \vec{x}_r^T \vec{x}_s = -\frac{1}{2} (d_{rs}^2 - \vec{x}_r^T \vec{x}_r - \vec{x}_s^T \vec{x}_s) \quad (\text{A.3.8})$$

A partir das equações (A.3.5), (A.3.6) e (A.3.7), podemos isolar os termos necessários para obter o valor de b_{rs} :

$$\begin{aligned}
 -\vec{x}_s^T \vec{x}_s &= -\frac{1}{n} \sum_{r=1}^n d_{rs}^2 + \frac{1}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r \\
 -\vec{x}_r^T \vec{x}_r &= -\frac{1}{n} \sum_{s=1}^n d_{rs}^2 + \frac{1}{n} \sum_{s=1}^n \vec{x}_s^T \vec{x}_s \\
 -\vec{x}_r^T \vec{x}_r - \vec{x}_s^T \vec{x}_s &= -\frac{1}{n} \sum_{r=1}^n d_{rs}^2 - \frac{1}{n} \sum_{s=1}^n d_{rs}^2 + \frac{2}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r \\
 \frac{2}{n} \sum_{r=1}^n \vec{x}_r^T \vec{x}_r &= \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n d_{rs}^2
 \end{aligned} \tag{A.3.9}$$

Substituindo os valores acima em (A.3.8), finalmente escrevemos b_{rs} em função das distâncias entre os pares:

$$\begin{aligned}
 b_{rs} &= -\frac{1}{2} \left(d_{rs}^2 - \frac{1}{n} \sum_{r=1}^n d_{rs}^2 - \frac{1}{n} \sum_{s=1}^n d_{rs}^2 + \frac{1}{n^2} \sum_{r=1}^n \sum_{s=1}^n d_{rs}^2 \right) \\
 &= a_{rs} - a_{r.} - a_{.s} + a_{..}
 \end{aligned} \tag{A.3.10}$$

Seja $a_{rs} = -\frac{1}{2}d_{rs}^2$. Podemos escrever

$$a_{r.} = n^{-1} \sum_{s=1}^n a_{rs}, \quad a_{.s} = n^{-1} \sum_{r=1}^n a_{rs}, \quad a_{..} = n^{-2} \sum_{r=1}^n \sum_{s=1}^n a_{rs}$$

e expressar b_{rs} em termos de a_{rs} , por conveniência:

$$b_{rs} = a_{rs} - a_{r.} - a_{.s} + a_{..} \tag{A.3.11}$$

Considerando a matriz $A = -\frac{1}{2}D$, temos $A = \{a_{rs}\}$ para $rs = 1, 2, \dots, n$. Definindo a matriz unitária $U_{n \times n} = \{1\}$, podemos finalmente computar todos os elementos da matriz de produtos internos B :

$$B = \left(I - \frac{1}{n}U \right) A \left(I - \frac{1}{n}U \right) = A - A \frac{U}{n} - \frac{U}{n} A - \frac{1}{n^2} U A U \tag{A.3.12}$$

Definindo a matriz centralizadora H como:

$$H = I - \frac{1}{n}U \tag{A.3.13}$$

Temos que a matriz B é computada por:

$$B = H A H \tag{A.3.14}$$

Após obter os valores da matriz B , o passo seguinte é recuperar as coordenadas dos pontos em R^d , assim encontrando a incorporação dos dados. A matriz de produtos internos B é dada por:

$$B_{n \times n} = X_{n \times p}^T X_{n \times p} \quad (\text{A.3.15})$$

onde n é o número de amostras e p é a dimensionalidade original dos dados de entrada. A literatura nos garante que essa matriz é simétrica, positiva semi-definida e possui posto p . Em outras palavras, a matriz B possui p autovalores não-negativos e $n - p$ autovalores iguais a zero.

Escrevendo a matriz B em termos da decomposição espectral, temos:

$$B = V \Lambda V^T \quad (\text{A.3.16})$$

onde $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ é a matriz diagonal dos autovalores de B , e $V = [v_1, v_2, \dots, v_n]$ é uma matriz na qual as colunas são os autovetores correspondentes aos autovalores, normalizados.

Desconsiderando os $n - p$ autovalores iguais a zero e seus autovetores correspondentes, reescrevemos B como:

$$B = \tilde{V} \tilde{\Lambda} \tilde{V}^T \quad (\text{A.3.17})$$

onde $\tilde{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ e $\tilde{V} = [v_1, v_2, \dots, v_p]$.

Finalmente, sabendo que a matriz B é dada por $B = X^T X$, podemos recuperar as coordenadas de X :

$$X = \tilde{\Lambda}^{\frac{1}{2}} \tilde{V}^T \quad (\text{A.3.18})$$

onde $\tilde{\Lambda}^{\frac{1}{2}} = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_p})$ é a matriz diagonal da raiz dos p autovalores diferentes de zero, e \tilde{V} é a matriz cujas colunas são os autovetores normalizados associados à esses p autovalores.

APÊNDICE B – Tabelas de resultados

Tabela 3 – Quantidade de amostras, atributos e classes para todos os conjuntos de dados utilizados neste trabalho, disponíveis em openML.org.

Nome completo do <i>dataset</i>	nº de amostras	nº de atributos	nº de classes
iris	150	4	3
prnn_crabs	200	7	2
servo	167	4	2
tic-tac-toe	958	9	2
visualizing_galaxy	323	4	2
sleuth_ex1605	62	5	2
car-evaluation	1728	21	4
Smartphone-Based_Recognition_of_Human_Activities	180	66	6
blogger	100	5	2
sa-heart	462	9	2
pyrim	74	27	2
SPECTF	349	44	2
veteran	137	7	2
balance-scale	625	4	3
parity5	32	5	2
kidney	76	6	2
aids	50	4	2
hayes-roth	132	4	2
breast-tissue	106	9	4
diabetes_numeric	43	2	2
parkinsons	195	22	2
rabe_131	50	5	2
corral	160	6	2
baskball	96	4	2
grub-damage	155	8	2
bolts	40	7	2
haberman	306	3	2
TuningSVMs	156	80	2
backache	180	31	2
prnn_synth	250	2	2
visualizing_environmental	111	3	2
mu284	284	10	2
ar4	107	29	2
vineyard	52	2	2
confidence	72	3	2
lungcancer_GSE31210	226	23	2
user-knowledge	403	5	5
fri_c3_100_5	100	5	2
tae	151	5	3
disclosure_z	662	3	2
diggle_table_a2	310	8	2
threeOf9	512	9	2
cloud	108	7	2
fl2000	67	15	2
triazines	186	60	2
acute-inflammations	120	6	2
vertebra-column	310	6	3
pm10	500	7	2
planning-relax	182	12	2
sensory	576	11	2
xd6	973	9	2
segment	2310	19	2
fri_c4_500_10	500	10	2
vehicle	846	18	4
ilpd	583	10	2
seeds	210	7	3
auto_price	159	15	2
wisconsin	194	32	2

Tabela 4 – Acurácia média obtida por 8 classificadores após utilizar o K-ISOMAP (nas variantes A e B) e outros métodos de redução de dimensionalidade, para diversos conjuntos de dados disponíveis em openML.org. (Caso 2-D)

dataset	PCA	KPCA	ISOMAP	LLE	Lap. Eig.	t-SNE	UMAP	A	B
iris	0.932	0.825	0.893	0.798	0.667	0.908	0.973	0.965	0.965
prnn_crabs	0.594	0.583	0.599	0.583	0.561	0.752	0.754	0.818	0.821
servo	0.860	0.740	0.823	0.829	0.762	0.830	0.931	0.933	0.942
tic-tac-toe	0.657	0.720	0.646	0.638	0.726	0.778	0.791	0.792	0.788
visualizing_galaxy	0.862	0.824	0.819	0.727	0.706	0.899	0.905	0.916	0.907
sleuth_ex1605	0.556	0.544	0.577	0.609	0.504	0.609	0.593	0.641	0.661
car-evaluation	0.770	0.767	0.683	0.735	0.648	0.696	0.672	0.782	0.757
Smartphone	0.632	0.681	0.718	0.393	0.464	0.906	0.883	0.807	0.836
blogger	0.680	0.657	0.713	0.710	0.635	0.708	0.670	0.748	0.738
sa-heart	0.660	0.667	0.705	0.659	0.665	0.674	0.686	0.718	0.726
pyrim	0.841	0.770	0.740	0.777	0.713	0.784	0.750	0.831	0.845
SPECTF	0.757	0.751	0.710	0.686	0.759	0.751	0.767	0.776	0.769
veteran	0.663	0.698	0.647	0.658	0.652	0.685	0.681	0.730	0.730
balance-scale	0.668	0.726	0.741	0.617	0.565	0.760	0.740	0.783	0.867
parity5	0.445	0.383	0.4531	0.367	0.383	0.438	0.336	0.609	0.914
kidney	0.615	0.615	0.645	0.599	0.556	0.605	0.658	0.701	0.694
aids	0.375	0.325	0.370	0.355	0.370	0.430	0.425	0.545	0.520
hayes-roth	0.570	0.638	0.636	0.621	0.616	0.674	0.564	0.669	0.699
breast-tissue	0.422	0.467	0.443	0.495	0.476	0.531	0.424	0.524	0.524
diabetes_numeric	0.551	0.528	0.551	0.653	0.534	0.659	0.545	0.733	0.750
parkinsons	0.858	0.834	0.777	0.775	0.800	0.850	0.834	0.862	0.843
rabe_131	0.735	0.955	0.840	0.760	0.755	0.775	0.910	0.950	0.960
corral	0.877	0.845	0.830	0.694	0.780	0.908	0.762	0.909	0.895
basketball	0.617	0.646	0.578	0.596	0.539	0.654	0.630	0.724	0.708
grub-damage	0.692	0.609	0.638	0.651	0.644	0.665	0.684	0.692	0.699
bolts	0.781	0.838	0.844	0.662	0.750	0.812	0.825	0.887	0.938
haberman	0.751	0.723	0.747	0.730	0.720	0.726	0.744	0.771	0.765
TuningSVMs	0.633	0.611	0.652	0.628	0.625	0.625	0.673	0.702	0.707
backache	0.854	0.853	0.850	0.853	0.853	0.839	0.851	0.865	0.864
prnn_synth	0.860	0.828	0.863	0.765	0.706	0.859	0.846	0.868	0.857
environmental	0.676	0.683	0.676	0.583	0.578	0.661	0.627	0.694	0.681
mu284	0.938	0.932	0.916	0.796	0.775	0.933	0.908	0.944	0.946
ar4	0.803	0.810	0.787	0.782	0.778	0.787	0.789	0.815	0.819
vineyard	0.788	0.779	0.793	0.798	0.764	0.779	0.798	0.856	0.837
confidence	0.833	0.837	0.840	0.799	0.799	0.826	0.840	0.840	0.844
lungcancer	0.826	0.847	0.816	0.831	0.845	0.845	0.845	0.845	0.854
user-knowledge	0.522	0.500	0.574	0.447	0.455	0.657	0.627	0.640	0.673
fri_c3_100_5	0.662	0.665	0.688	0.623	0.660	0.670	0.670	0.728	0.733
tae	0.431	0.480	0.498	0.413	0.424	0.539	0.503	0.546	0.525
disclosure_z	0.502	0.527	0.500	0.498	0.503	0.515	0.512	0.538	0.550
diggle_table_a2	0.859	0.898	0.880	0.845	0.766	0.888	0.889	0.907	0.906
threeOf9	0.602	0.613	0.656	0.683	0.583	0.716	0.773	0.678	0.701
cloud	0.657	0.632	0.641	0.639	0.632	0.623	0.655	0.669	0.667
fl2000	0.640	0.607	0.621	0.625	0.577	0.629	0.621	0.654	0.669
triazines	0.644	0.638	0.653	0.625	0.629	0.681	0.690	0.679	0.675
inflammations	0.894	0.935	0.879	0.673	0.656	0.954	1.000	0.917	0.944
vertebra-column	0.671	0.663	0.707	0.652	0.631	0.781	0.777	0.720	0.746
pm10	0.510	0.533	0.513	0.492	0.509	0.548	0.529	0.534	0.553
planning-relax	0.669	0.661	0.651	0.687	0.665	0.651	0.690	0.703	0.692
sensory	0.559	0.568	0.562	0.573	0.579	0.585	0.584	0.588	0.576
xd6	0.682	0.677	0.729	0.658	0.686	0.800	0.716	0.765	0.769
segment	0.815	0.847	0.870	0.782	0.809	0.954	0.956	0.934	0.936
fri_c4_500_10	0.542	0.523	0.580	0.531	0.560	0.631	0.594	0.591	0.601
vehicle	0.450	0.391	0.436	0.302	0.363	0.582	0.576	0.509	0.507
ilpd	0.655	0.700	0.669	0.680	0.694	0.664	0.663	0.702	0.698
seeds	0.901	0.883	0.911	0.813	0.675	0.915	0.912	0.924	0.921
auto_price	0.925	0.948	0.917	0.773	0.834	0.948	0.941	0.934	0.950
wisconsin	0.610	0.568	0.597	0.561	0.564	0.607	0.602	0.620	0.624
Média	0.690	0.690	0.693	0.651	0.640	0.727	0.721	0.754	0.764
Mediana	0.668	0.679	0.685	0.658	0.650	0.712	0.703	0.740	0.753
Min.	0.375	0.325	0.367	0.302	0.363	0.430	0.336	0.509	0.507
Max.	0.938	0.955	0.917	0.853	0.853	0.954	1.000	0.965	0.965

Tabela 5 – Coeficientes de silhueta do agrupamento de dados produzido pelo K-ISOMAP (na variantes A e B) e outros métodos de redução de dimensionalidade, para diversos conjuntos de dados disponíveis em openML.org. (Caso 2-D)

dataset	PCA	KPCA	ISOMAP	LLE	Lap. Eig.	t-SNE	UMAP	A	B
iris	0.401	0.469	0.453	0.365	0.542	0.495	0.529	0.491	0.537
prnn_crabs	0.041	0.030	0.037	0.023	0.029	0.031	0.051	0.076	0.078
servo	0.122	0.106	0.114	0.104	0.085	0.056	0.396	0.471	0.470
tic-tac-toe	-0.023	-0.019	-0.020	0.007	-0.011	0.005	0.154	0.366	0.366
visualizing_galaxy	0.179	0.256	0.194	0.235	0.271	0.166	0.128	0.309	0.304
sleuth_ex1605	0.126	0.081	0.122	0.088	0.129	0.083	0.092	0.150	0.136
car-evaluation	-0.032	-0.157	-0.095	-0.062	-0.104	-0.093	-0.030	0.162	0.211
Smartphone	0.062	0.250	0.160	0.072	0.221	0.412	0.460	0.402	0.413
blogger	0.036	-0.011	0.053	0.029	0.003	0.034	-0.010	0.245	0.255
sa-heart	0.092	0.051	0.094	0.083	0.030	0.079	0.072	0.077	0.082
pyrim	0.193	0.140	0.164	0.210	0.199	0.158	0.158	0.212	0.203
SPECTF	-0.018	0.093	-0.036	-0.235	0.019	-0.017	0.110	0.084	0.090
veteran	0.001	-0.007	0.003	0.017	0.003	-0.001	0.000	0.030	0.028
balance-scale	0.025	0.065	0.111	0.052	0.057	0.071	0.051	0.095	0.115
parity5	-0.062	-0.052	-0.049	-0.051	-0.036	-0.034	-0.051	0.028	0.310
kidney	0.020	0.028	0.018	0.042	0.035	0.007	0.013	0.057	0.056
aids	-0.022	-0.027	-0.028	-0.038	-0.034	-0.014	-0.019	0.002	-0.000
hayes-roth	-0.023	0.038	-0.010	-0.014	-0.014	-0.012	-0.008	0.037	0.048
breast-tissue	-0.029	-0.030	-0.017	-0.082	-0.018	-0.025	-0.034	-0.015	-0.014
diabetes_numeric	0.092	0.082	0.093	0.085	0.089	0.077	0.063	0.124	0.112
parkinsons	0.131	0.156	0.115	0.002	0.242	0.204	0.202	0.181	0.176
rabe_131	0.147	0.291	0.193	0.255	0.305	0.233	0.304	0.325	0.318
corral	0.304	0.290	0.254	0.042	0.219	0.299	0.302	0.304	0.302
basketball	0.070	0.089	0.058	0.111	0.068	0.075	0.063	0.116	0.116
grub-damage	0.043	0.050	0.067	0.095	0.132	0.095	0.091	0.101	0.094
bolts	0.338	0.254	0.287	0.029	0.318	0.216	0.322	0.370	0.379
haberman	0.060	-0.024	0.062	-0.004	-0.033	0.006	-0.029	0.113	0.117
TuningSVMs	0.006	0.016	0.008	-0.006	0.033	0.007	-0.000	0.029	0.030
backache	0.124	-0.060	0.189	-0.016	0.172	0.083	0.039	0.120	0.155
prnn_synth	0.246	0.264	0.236	0.231	0.286	0.182	0.229	0.262	0.259
environmental	0.105	0.103	0.104	0.089	0.108	0.111	0.103	0.120	0.112
mu284	0.301	0.339	0.287	0.284	0.390	0.360	0.249	0.350	0.347
ar4	0.358	0.176	0.318	0.203	0.131	0.184	0.171	0.202	0.207
vineyard	0.278	0.263	0.281	0.192	0.252	0.294	0.385	0.480	0.369
confidence	0.173	0.214	0.123	-0.131	-0.082	0.129	0.056	0.217	0.244
lungcancer	0.087	0.048	0.090	0.070	0.048	0.033	0.033	0.083	0.086
user-knowledge	-0.036	-0.071	-0.043	-0.031	-0.045	0.003	-0.058	0.007	0.002
fri_c3_100_5	0.078	0.141	0.086	0.095	0.169	0.096	0.129	0.138	0.133
tae	-0.059	-0.005	-0.072	-0.023	-0.046	-0.021	-0.029	-0.010	-0.014
disclosure_z	-0.002	0.006	-0.002	-0.001	-0.002	0.001	0.001	0.012	0.013
diggle_table_a2	0.407	0.410	0.450	0.328	0.304	0.423	0.277	0.432	0.448
threeOf9	0.035	0.059	0.049	0.095	0.045	0.062	0.106	0.185	0.089
cloud	-0.009	-0.001	-0.015	-0.009	-0.011	-0.013	-0.023	-0.002	-0.008
fl2000	0.181	0.044	0.119	0.074	0.025	0.040	0.036	0.136	0.136
triazines	0.009	0.065	0.015	0.054	0.023	0.026	0.041	0.037	0.035
inflammations	0.279	0.316	0.266	0.133	0.048	0.248	0.373	0.272	0.277
vertebra-column	0.059	0.110	0.059	0.065	0.047	0.175	0.218	0.183	0.178
pm10	0.001	0.002	0.000	0.000	-0.000	0.001	-0.000	0.003	0.004
planning-relax	-0.002	-0.011	-0.004	0.003	-0.005	-0.004	0.004	0.017	0.016
sensory	0.000	0.002	-0.000	0.005	0.006	-0.001	0.002	0.003	0.004
xd6	0.029	0.029	0.016	0.055	0.021	0.049	0.058	0.178	0.176
segment	-0.162	-0.028	-0.165	-0.232	-0.240	0.043	0.155	0.222	0.225
fri_c4_500_10	0.002	0.007	0.007	0.028	0.006	0.026	0.029	0.033	0.034
vehicle	-0.062	-0.075	-0.068	-0.041	-0.051	-0.050	-0.029	-0.043	-0.045
ilpd	-0.026	0.035	0.058	-0.039	0.085	0.058	0.077	0.084	0.077
seeds	0.444	0.547	0.471	0.373	0.616	0.526	0.522	0.492	0.492
auto_price	0.429	0.377	0.433	0.322	0.409	0.435	0.467	0.402	0.398
wisconsin	0.021	0.029	0.029	0.005	0.035	0.019	0.035	0.034	0.034
Média	0.096	0.101	0.099	0.063	0.095	0.106	0.121	0.165	0.169
Mediana	0.051	0.051	0.060	0.042	0.040	0.057	0.063	0.122	0.125
Min.	-0.162	-0.157	-0.165	-0.235	-0.240	-0.093	-0.058	-0.043	-0.045
Max.	0.444	0.547	0.471	0.373	0.616	0.526	0.529	0.492	0.537