CARLOS HUMBERTO PORTO FILHO

**TÉCNICAS DE APRENDIZADO NÃO
SUPERVISIONADO BASEADAS NO ALGORITMO
DA CAMINHADA DO TURISTA**

VERSÃO CORRIGIDA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação Interunidades Bioengenharia - Escola de Engenharia de São Carlos / Faculdade de Medicina de Ribeirão Preto / Instituto de Química de São Carlos da Universidade de São Paulo como parte dos requisitos para a obtenção do título de mestre em Ciências.

Área de Concentração: Bioengenharia

Orientador: Prof. Dr. Joaquim Cezar Felipe

São Carlos,
2017

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

P839t Porto Filho, Carlos Humberto
Técnicas de Aprendizado Não Supervisionado Baseadas no Algoritmo da Caminhada do Turista / Carlos Humberto Porto Filho; orientador Joaquim Cezar Felipe. São Carlos, 2017.

Dissertação (Mestrado) - Programa de Pós-Graduação Interunidades em Bioengenharia e Área de Concentração em Bioengenharia -- Escola de Engenharia de São Carlos; Faculdade de Medicina de Ribeirão Preto; Instituto de Química de São Carlos, da Universidade de São Paulo, 2017.

1. Aprendizado não supervisionado. 2. Agrupamento de dados. 3. Caminhada do turista. I. Título.

AGRADECIMENTOS

À minha família, por todo o apoio e incentivo.

À minha namorada, por toda paciência e ajuda.

Ao meu orientador, Prof. Joaquim, pelos ensinamentos e confiança.

À CAPES, pela bolsa de mestrado.

RESUMO

PORTO FILHO, C. H. Técnicas de Aprendizado Não Supervisionado Baseadas no Algoritmo da Caminhada do Turista. 2017. 122f. Dissertação (Mestrado) - Programa de Pós-Graduação Interunidades em Bioengenharia - Escola de Engenharia de São Carlos / Faculdade de Medicina de Ribeirão Preto / Instituto de Química de São Carlos - Universidade de São Paulo, São Carlos, 2017.

Nas últimas décadas, a quantidade de informações armazenadas no formato digital tem crescido de forma exponencial, levando à necessidade cada vez maior de produção de ferramentas computacionais que auxiliem na geração do conhecimento a partir desses dados. A área de Aprendizado de Máquina fornece diversas técnicas capazes de identificar padrões nesses conjuntos de dados. Dentro dessas técnicas, este trabalho destaca o Aprendizado de Máquina Não Supervisionado onde o objetivo é classificar as entidades em *clusters* (grupos) mutuamente exclusivos baseados na similaridade entre as instâncias. Os *clusters* não são pré-definidos e daí o elemento não supervisionado. Organizar esses dados em *clusters* que façam sentido é uma das maneiras mais fundamentais de entendimento e aprendizado. A análise de *clusters* é o estudo dos métodos para agrupamento e se divide entre hierárquico e particional. A classificação hierárquica é uma sequência encadeada de partições enquanto que na particional há somente uma partição. O interesse deste trabalho são as técnicas baseadas em uma caminhada determinística parcialmente auto repulsiva conhecida como caminhada do turista. Partindo da hipótese de que é possível utilizar a caminhada do turista como uma técnica de Aprendizado de Máquina Não Supervisionado, foi implementado um algoritmo hierárquico baseado na caminhada do turista proposto por Campiteli et al. (2006). Foi avaliado, através de diferentes conjuntos de imagens médicas, como essa técnica se compara com técnicas hierárquicas tradicionais. Também é proposto um novo algoritmo de Aprendizado de Máquina Não Supervisionado particional baseado na caminhada do turista, chamado de *Tourist Walk Partitional Clustering* (TWPC). Os resultados mostraram que a técnica hierárquica baseada na caminhada do turista é capaz de identificar *clusters* em conjuntos de imagens médicas através de uma árvore que não impõe uma estrutura binária, com um número menor de hierarquias e uma invariabilidade à escala dos dados, resultando em uma estrutura mais organizada. Mesmo que a árvore não seja diretamente baseada nas distâncias dos dados, mas em um ranking de vizinhos, ela ainda preserva uma correlação entre suas distâncias cofenéticas e as distâncias reais entre os dados. O método particional proposto TWPC foi capaz de encontrar, de forma eficiente, formas arbitrárias de *clusters* com variações *inter-cluster* e *intra-cluster*. Além disso o algoritmo tem como vantagens: ser determinístico; funcionar com interações locais, sem a necessidade de conhecimento *a priori* de todos os itens do conjunto; incorporar o conceito de ruído e *outlier*; e funcionar com um ranking de vizinhos, que pode ser construído através de qualquer medida.

Palavras Chave: Aprendizado não supervisionado, Agrupamento de dados, Caminhada do turista.

ABSTRACT

PORTO FILHO, C. H. Unsupervised Learning Techniques Based on the Tourist Walk Algorithm. 2017. 122f. Dissertação (Mestrado) - Programa de Pós-Graduação Interunidades em Bioengenharia - Escola de Engenharia de São Carlos / Faculdade de Medicina de Ribeirão Preto / Instituto de Química de São Carlos - Universidade de São Paulo, São Carlos, 2017.

In the last decades, the amount of data stored in digital format has grown exponentially, leading to the increasing need to produce computational tools that help generate knowledge from these data. The Machine Learning field provides several techniques capable of identifying patterns in these data sets. Within these techniques we highlight the Unsupervised Machine Learning where the objective is to classify the entities in mutually exclusive clusters based on the similarity between the instances. Clusters are not predefined and hence the unsupervised element. Organizing this data into clusters that make sense is one of the most fundamental ways of understanding and learning. Cluster analysis is the study of methods for clustering and is divided between hierarchical and partitional. A hierarchical clustering is a sequence of partitions whereas in the partitional clustering there is only one partition. Here we are interested in techniques based on a deterministic partially self-avoiding walk, known as tourist walk. Based on the hypothesis that it is possible to use the tourist walk as an unsupervised machine learning technique, we have implemented a hierarchical algorithm based on the tourist walk proposed by Campiteli et al. (2006). We evaluate this algorithm using different sets of medical images and compare it with traditional hierarchical techniques. We also propose a new algorithm for partitional clustering based on the tourist walk, called Tourist Walk Partitional Clustering (TWPC). The results showed that the hierarchical technique based on the tourist walk is able to identify clusters in sets of medical images through a tree that does not impose a binary structure, with a smaller number of hierarchies and is invariable to scale transformation, resulting in a more organized structure. Even though the tree is not directly based on the distances of the data but on a ranking of neighbors, it still preserves a correlation between its cophenetic distances and the actual distances between the data. The proposed partitional clustering method TWPC was able to find, in an efficient way, arbitrary shapes of clusters with inter-cluster and intra-cluster variations. In addition, the algorithm has the following advantages: it is deterministic; it operates based on local interactions, without the need for a priori knowledge of all the items in the set; it is capable of incorporate the concept of noise and outlier; and work with a ranking of neighbors, which can be built through any measure.

Keywords: Unsupervised learning, Clustering, Tourist walk

LISTA DE ILUSTRAÇÕES

Figura 1 - Passo a passo de uma caminhada do turista com memória $\mu = 4$	39
Figura 2 – (a) Item na borda e dentro do cluster (b) Densidade alcançável diretamente	46
Figura 3 - (a) Densamente alcançável (b) Densamente conectado	46
Figura 4 - Heurística para determinar Eps.....	47
Figura 5 - Resultado do CLARANS	48
Figura 6 - Resultado do DBSCAN	49
Figura 7 - Imagens dos diferentes níveis de atipia em biópsias de pólipos colorretais	61
Figura 8 - Imagens de áreas de interesse de nódulos de mama benigno (a) e maligno (b).....	62
Figura 9 - Imagens de ressonâncias magnéticas de diversas partes do corpo humano.....	63
Figura 10 - Lesão do tecido do colo uterino	64
Figura 11 - Primeiros dois componentes do PCA dos vetores de características das imagens de pólipos colorretais	66
Figura 12 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de pólipos colorretais	66
Figura 13 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de pólipos colorretais	69
Figura 14 - Primeiros dois componentes do PCA dos vetores de características das imagens de ressonâncias magnéticas.....	70
Figura 15 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de ressonâncias magnéticas.....	71
Figura 16 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de ressonâncias magnéticas.....	73
Figura 17 - Primeiros dois componentes do PCA dos vetores de características das imagens de tecido epitelial do colo do útero	74
Figura 18 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de tecido epitelial do colo do útero.....	74

Figura 19 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de tecido epitelial do colo do útero.....	76
Figura 20 - Primeiros dois componentes do PCA dos vetores de características das imagens de regiões de interesse em nódulos de mama.....	77
Figura 21 - Dendogramas resultantes das técnicas hierárquicas para as imagens de regiões de interesse de nódulos de mama.....	77
Figura 22 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de regiões de interesse de nódulos de mama.....	79
Figura 23 - Duas distribuições normais formando dois clusters bem separados.....	82
Figura 24 - Conjunto de Dados com Ruído de fundo (background clutter)	82
Figura 25 - Fundamental Clustering Problem Suite - Conjunto de dados com presença de ruído.....	83
Figura 26 - Conjunto de dados com clusters de formas arbitrárias. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 153$..	84
Figura 27 - Conjunto de dados com clusters de formas arbitrárias. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 154$..	85
Figura 28 - Conjunto de dados com forma de duas espirais. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 500$	85
Figura 29 -Conjuntos de dados com diferentes formatos de cluster mostrando que os atratores da caminhada do turista têm preferência por formatos globulares.....	86
Figura 30 - Conjunto de dados com 4 clusters com variações inter-cluster e intra-cluster.....	88
Figura 31 – Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 3$	89
Figura 32 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 3$	89
Figura 33 - Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 4$	90
Figura 34 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 4$	90
Figura 35 - Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 5$	91

Figura 36 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 5$	91
Figura 37 – Resultado obtido com o K-means em um conjunto com o cluster central alongado para $K = 4$	92
Figura 38 - Heurística para determinar eps no DBSCAN: o gráfico mostra a média das distâncias de cada item para seus K vizinhos mais próximos.	93
Figura 39 - Resultado obtido com o DBSCAN em um conjunto com o cluster central alongado com $\text{eps} = 0,8$	94
Figura 40 - Resultado obtido com o Affinity propagation em um conjunto com o cluster central alongado com $K = 4$	95
Figura 41 - Resultado obtido com o TWPC em um conjunto de dados com o cluster central alongado com $\mu = 5$	96
Figura 42 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 5$	97
Figura 43 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 5$. Os itens em vermelho são itens com frequência em caminhadas < 2	98
Figura 44 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 50$. Os itens em vermelho são itens com frequência em caminhadas < 2	99
Figura 45 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 100$. Os itens em vermelho são itens com frequência em caminhadas < 2	99
Figura 46 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 150$. Os itens em vermelho são itens com frequência em caminhadas < 2	100
Figura 47 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes variâncias	101
Figura 48 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes variâncias e distâncias inter-cluster.....	101
Figura 49 - Fundamental Clustering Problem Suite - Conjunto de dados com Clusters muito próximos	102

Figura 50 - Fundamental Clustering Problem Suite - Conjunto de dados não separável linearmente.....	102
Figura 51 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes densidades	103
Figura 52 - As cores representam os clusters encontrados pelo K-means	104
Figura 53 - As cores representam os clusters encontrados pelo DBSCAN. Os itens em vermelho foram identificados como ruído	104
Figura 54 - As cores representam os clusters encontrados pelo Affinity propagation	105
Figura 55 - As cores representam os clusters encontrados pelo TWPC com $\mu = 6$...	105
Figura 56 - As cores representam os clusters encontrados pelo K-means	105
Figura 57 - As cores representam os clusters encontrados pelo DBSCAN. O item em vermelho foi identificado como ruído.....	106
Figura 58 - As cores representam os clusters encontrados pelo Affinity propagation	106
Figura 59 - As cores representam os clusters encontrados pelo TWPC com $\mu = 2$...	107
Figura 60 - Conjunto complexo onde dois clusters estão unidos por uma "ponte" (destacada pelo círculo pontilhado vermelho)	107
Figura 61 - As cores indicam os clusters encontrados pelo TWPC com $\mu = 10$. Note que dois clusters foram identificados como apenas um (em marrom).....	108
Figura 62 - Dois clusters unidos ao centro por uma "ponte"	108
Figura 63 - Dois clusters bem definidos sem "ponte" entre eles.....	109
Figura 64 - O algoritmo TWPC com otimização foi capaz de encontrar os 3 clusters com apenas 3 caminhadas. As cores representam os clusters encontrados	111
Figura 65 - Conjunto de dados com 4811 instâncias.....	111
Figura 66 - Temos no eixo horizontal o tamanho da memória μ da caminhada do turista. No eixo vertical temos (1) o tamanho médio das caminhadas feitas (2) o número de caminhadas feitas no TWPC com otimização e (3) o tempo de processamento	113

LISTA DE ALGORITMOS

Algoritmo 1 - Tourist Walk Partitional Clustering	86
Algoritmo 2 - Versão do TWPC onde a média das silhuetas dos itens é usada para definir o parâmetro μ de forma automática.....	87
Algoritmo 3 - Algoritmo do TWPC com otimização	110

SUMÁRIO

1. Introdução.....	19
1.1. Contextualização.....	19
1.2. Motivação.....	21
1.3. Objetivos.....	22
1.4. Estrutura.....	23
2. Base Conceitual.....	24
2.1. Aprendizado de Máquina.....	24
2.1.1. Aprendizado de Máquina Não Supervisionado.....	24
2.1.2. Extração de Atributos.....	25
2.1.3. Medidas de Distância.....	27
2.1.4. Métodos de <i>Clustering</i>	29
2.1.4.1. Métodos Baseados em Centróide.....	30
2.1.4.2. Métodos Baseados em Dissimilaridade.....	31
2.1.4.3. Métodos Não Paramétricos Inspirados na Natureza.....	34
2.2. Caminhadas em dados.....	35
2.2.1. Caminhadas aleatórias.....	36
2.2.2. Caminhadas Determinísticas.....	36
2.2.2.1. Caminhada do Turista.....	37
2.3. Validação de <i>Clusters</i>	41
2.3.1. Método da Silhueta.....	42
2.3.2. Correlação Cofenética.....	43
3. Trabalhos Correlatos.....	45
3.1. DBSCAN.....	45
3.2. Affinity Propagation.....	50
3.3. Aprendizado Supervisionado Baseado na Caminhada do Turista.....	53
3.4. Aprendizado Não Supervisionado Baseado na Caminhada do Turista.....	59
4. Estudo da Utilização da Caminhada do Turista como Método de <i>Clustering</i>	
Hierárquico.....	61
4.1. Materiais e métodos.....	61
4.1.1. Bases de Imagens Médicas.....	61
4.2. Ambiente de Desenvolvimento.....	64
4.3. Resultados.....	65
4.3.1. Biópsias de Pólipos Colorretais.....	65
4.3.2. Imagens de Ressonância Magnética.....	70
4.3.3. Imagens Histopatológicas de Tecido Epitelial do Colo Uterino.....	73
4.3.4. Imagens de Regiões de Interesse em Nódulos de Mama.....	76
5. Proposta: Algoritmo de <i>Clustering</i> Particional Baseado na Caminhada do	
Turista.....	81
5.1. Introdução.....	81
5.2. Tourist Walk Partitional Clustering (TWPC).....	86
5.2.1. Ruído e Outliers.....	96
5.3. Experimentos.....	100
5.3.1. Comparativo.....	103
5.3.2. Desvantagens.....	107
5.4. Otimização.....	109
6. Conclusões.....	115
6.1. Trabalhos Futuros.....	116

1. Introdução

1.1. Contextualização

Nas últimas décadas, a quantidade de informações armazenadas no formato digital tem crescido de forma exponencial, levando à necessidade cada vez maior da produção de ferramentas computacionais que auxiliem na geração do conhecimento a partir desses dados. Boa parte dessas informações, até pouco tempo, eram simplesmente descartadas ou não coletadas. Hoje elas têm sido armazenadas de forma sistemática e estão disponíveis para estudo. Embora certos tipos de informações possam ser analisados e entendidos a partir de estatísticas simples, uma parcela significativa exige técnicas mais complexas para a geração efetiva do conhecimento (HAIR et al., 2010).

Um tipo especial de informação que tem sido cada vez mais explorado no formato digital são as imagens. Avanços nas tecnologias de aquisição e armazenamento permitiram sua ampla disseminação. A produção e utilização das imagens digitais na sociedade moderna é enorme. As bases de imagens frequentemente possuem informações úteis, mas considerando seu grande volume e padrões complexos, torna-se difícil para os seres humanos descobrirem essas informações. O processamento de imagens se torna então essencial nesse contexto e é um elemento crítico na ciência da informação, de modo que muitas abordagens científicas simplesmente não existiriam sem ele. A aplicação do processamento de imagens digitais pode ser encontrada nas mais diversas áreas, tais como radiologia, microscopia, astronomia, geoprocessamento, visão computacional e geologia, entre muitas outras (GONZALEZ; WOODS, 2008). O processamento e a análise de imagens muitas vezes representam um desafio complexo, pois envolvem etapas como a aquisição da imagem, armazenamento, manipulação e a exibição da imagem. A análise de imagens propriamente dita envolve ainda a extração de medidas das imagens, preferencialmente de modo automático, e a utilização destas medidas para a classificação ou agrupamento automático das mesmas. A mineração de imagens pode nos ajudar a descobrir informações e padrões implícitos nas diversas bases disponíveis e é uma área de destaque dentro da mineração de dados. Há um desafio por exemplo em organizar grandes bancos de dados para melhorar o funcionamento de mecanismos de busca (HAIWEI; LI; WEI, 2005). O agrupamento de imagens é

utilizado em diferentes aplicações, tais como segmentação, recuperação baseada em conteúdo e categorização, entre outros. Considerando, por outro lado, a área de Aprendizado de Máquina, diversas técnicas têm sido aplicadas de forma bem-sucedida em sistemas reais. O Aprendizado de Máquina pode ser entendido como a área que estuda a descoberta automática de padrões em conjunto de dados e, usando esses padrões, permite tomar ações como classificar os dados em diferentes classes (BISHOP, 2006).

Organizar dados em grupos que fazem sentido é uma das maneiras mais fundamentais de entendimento e aprendizado. A análise de *clusters* é o estudo dos métodos para agrupamento de objetos que de algum modo apresentam similaridade. A análise de *cluster* não utiliza objetos rotulados e isso a diferencia da análise discriminante. O objetivo da análise de *cluster* é simplesmente encontrar uma organização válida e conveniente para os dados (JAIN; DUBES, 1988). A principal divisão entre métodos de *clustering* é entre hierárquico e particional. A classificação hierárquica é uma sequência encadeada de partições enquanto que na particional há somente uma partição, de forma que uma abordagem hierárquica pode ser vista como uma sequência especial de partições.

Quando temos informações *a priori* sobre as classes as quais os dados pertencem e exploramos tais informações para, através de padrões encontrados, classificar novos dados, temos então o aprendizado supervisionado, ou seja, há um "professor" que fornece informações externas. Porém esse nem sempre é o caso. Muito comumente não temos os rótulos disponíveis. Nesse tipo de problema temos um conjunto de vetores de características e o objetivo é revelar as similaridades e agrupar vetores parecidos. Essa abordagem é conhecida como aprendizado não supervisionado (THEODORIDIS; KOUTROUMBAS, 2009) (DUDA; HART; STORK, 2001).

O *clustering* é considerado o aspecto mais importante do aprendizado não supervisionado e muitas definições já foram propostas. De um modo geral as instâncias pertencentes a um mesmo *cluster* devem ser as mais similares possíveis, enquanto que instâncias em *clusters* diferentes as mais diferentes possíveis. Os algoritmos de *clustering* podem ser divididos em diferentes categorias. (XU; TIAN, 2015) propõe uma divisão em 9 categorias: baseado em partição, hierarquia, *fuzzy*, distribuição, densidade, teoria dos grafos, *grid*, teoria dos fractais e baseada em modelo.

Aqui estamos interessados em técnicas de aprendizado baseadas no que é conhecido como “caminhadas em conjuntos de dados”. Embora menos estudadas que as caminhadas aleatórias, as caminhadas determinísticas têm mostrado potencial para caracterizar dados (KINOUCI et al., 2002), (BACKES et al., 2006), (CAMPITELI et al., 2006), (SILVA; ZHAO, 2015). Dentro do grupo de caminhadas determinísticas, este projeto está focado na caminhada determinística parcialmente auto-repulsiva (ou auto-evitante) (*partially self-avoiding deterministic walk*) conhecida como “caminhada do turista” (LIMA; MARTINEZ; KINOUCI, 2001), (STANLEY; BULDYREV, 2001). A caminhada do turista é representada como um caminhante que, partindo de um ponto (cidade), segue para o próximo ponto (outra cidade), tomando como regra ir para o ponto mais próximo e que não tenha sido visitado nos últimos μ passos. Essa movimentação leva a uma trajetória formada por uma parte transiente e um ciclo (atrator) onde o caminhante fica "preso", caracterizando o fim da caminhada.

1.2. Motivação

Como dito anteriormente, a grande quantidade de dados gerados diariamente demanda novas técnicas para a efetiva geração do conhecimento. Nesse cenário, o caso da análise de imagens médicas pode se apresentar como um problema complexo. Não são incomuns, por exemplo, imagens que apresentem diferentes níveis de atipia com distinções graduais pouco definidas. Diferentemente de outras bases relacionais em que suas instâncias são bem definidas, as bases de imagens médicas possuem valores cujo significado precisa ser suportado pelo conhecimento e pela interpretação de especialistas da área médica. Kahneman (2011) argumenta que, em muitos casos, especialistas emitem decisões menos precisas que ferramentas computacionais desenvolvidas com esse objetivo. Diversos estudos mostraram que humanos são inconsistentes em seus julgamentos envolvendo informação de alta complexidade (DAWES, 1979). Quando questionados para avaliar a mesma informação duas vezes, frequentemente teremos respostas diferentes. Radiologistas experientes que avaliaram raios-x como normal ou anormal se contradisseram 20% das vezes quando viram a mesma imagem em situações diferentes (HOFFMAN; SLOVIC; RORER, 1968). Um *review* de 41 estudos sobre a confiabilidade das decisões tomadas por auditores,

patologistas, psicólogos e outros profissionais mostrou que este nível de inconsistência (20%) é o padrão (SHANTEAU, 1988).

Nesse contexto é interessante explorar novos algoritmos capazes de auxiliar a análise de dados. Neste trabalho usamos a caminhada do turista em duas abordagens distintas de aprendizado não supervisionado. A primeira abordagem é uma técnica de algoritmo de *clustering* hierárquico. Aqui estendemos o trabalho original aplicando-o em imagens médicas e formalizando como o dendrograma resultante se compara aos dendrogramas gerados por técnicas tradicionais.

Diferentes tipos de dados demandam diferentes algoritmos. Alguns problemas demandam resultados hierárquicos, outros, particionais. Isso nos motivou a desenvolver um novo algoritmo particional baseado na caminhada do turista. A nova técnica consegue, através de um movimento extremamente simples, apresentar diversas características desejáveis a algoritmos deste tipo.

1.3. Objetivos

O objetivo deste trabalho é implementar e demonstrar como a caminhada determinística parcialmente auto repulsiva, conhecida como caminhada do turista, pode ser usada no processo de *clustering*.

Especificamente iremos implementar o algoritmo hierárquico não supervisionado baseado na caminhada do turista. A implementação envolve o algoritmo e a visualização do dendrograma resultante. Iremos então avaliar como o algoritmo se compara com técnicas hierárquicas tradicionais. Para tanto serão usadas bases de imagens médicas e os dendrogramas resultantes serão comparados.

Será também proposto e implementado um novo algoritmo não supervisionado particional baseado na caminhada do turista.

A hipótese aqui investigada é de que é possível utilizar a caminhada do turista como uma técnica de aprendizado não supervisionado. Espera-se que o algoritmo hierárquico baseado na caminhada do turista permita gerar um dendrograma com características iguais ou superiores aos dendrogramas de algoritmos tradicionais. Já com a técnica particional é esperado um algoritmo simples e eficiente que apresente características desejáveis como: ser determinístico, aceitar formatos arbitrários de *clusters*, não ser sensível a variações de densidade *intra-cluster* e distâncias *inter-*

cluster, possuir o conceito de ruído e *outliers* e não exigir informações *a priori* dos dados.

1.4. Estrutura

Este trabalho está estruturado da seguinte forma:

- O capítulo 2 apresenta uma revisão dos temas que serão utilizados neste trabalho.
- O capítulo 3 descreve alguns trabalhos relevantes para os temas tratados aqui, incluindo a técnica de aprendizado não supervisionado hierárquico baseado na caminhada do turista que será usada como ponto de partida para os desenvolvimentos deste trabalho.
- O capítulo 4 apresenta os experimentos e resultados das comparações do algoritmo hierárquico baseado na caminhada do turista com algoritmos hierárquicos tradicionais.
- O capítulo 5 apresenta a proposta de um algoritmo de *clustering* particional baseado na caminhada do turista.
- O capítulo 6 apresenta as conclusões e possíveis trabalhos futuros.

2. Base Conceitual

2.1. Aprendizado de Máquina

Os humanos absorvem e assimilam conhecimento durante toda sua vida. A tentativa de simular essa característica em computadores caracteriza a área de pesquisa chamada Aprendizado de Máquina. Ela procura desenvolver métodos computacionais capazes de aprender com experiências passadas (BISHOP, 2006), (DUDA; HART; STORK, 2001), (THEODORIDIS; KOUTROUMBAS, 2009).

Usando a representação computacional dos dados, as técnicas de Aprendizado de Máquina podem gerar modelos capazes de reconhecer ou ainda imitar o comportamento de um especialista humano de modo automático. De uma maneira geral, essas técnicas são tradicionalmente divididas em duas abordagens: o aprendizado supervisionado e o aprendizado não supervisionado. No primeiro caso o processo procura construir uma função de mapeamento a partir do conjunto de treinamento. Quando as instâncias do conjunto de treinamento possuem rótulos com valores discretos, então chamamos a inferência de classificação, e quando os rótulos são contínuos, a chamamos de regressão. Já no aprendizado não supervisionado a ideia é encontrar estruturas intrínsecas aos dados e, nesse caso, o processo é guiado somente pelas relações entre os dados e nenhum conhecimento externo sobre os rótulos é fornecido.

2.1.1. Aprendizado de Máquina Não Supervisionado

O Aprendizado de Máquina não supervisionado é feito com dados não rotulados, ou seja, não se sabe quantas ou quais classes existem. É uma técnica usada para encontrar subgrupos de objetos. Especificamente, o objetivo é classificar as entidades em grupos mutuamente exclusivos baseados na similaridade entre as instâncias. Os grupos não são pré-definidos e não há um "professor", daí o elemento não supervisionado.

O *clustering* é a técnica mais importante do aprendizado de máquina não supervisionado e consiste em encontrar *clusters* (grupos) no conjunto de dados. Durante este processo esperamos encontrar *clusters* nos quais os itens de um mesmo

cluster sejam os mais similares possíveis entre si enquanto que itens em diferentes *clusters* sejam os mais distintos possíveis. A semelhança ou diferença dos dados é dada de acordo com alguma função de similaridade adotada (MITCHELL, 1997).

Existem diversas razões para o uso de uma técnica de *clustering*. Primeiramente, obter dados rotulados é extremamente custoso. Rotular grandes quantidades de dados de forma confiável pode ser inviável. Podemos usar a técnica também para encontrar características que nos serão úteis para uma categorização posterior (DUDA; HART; STORK, 2001). A classificação não supervisionada pode ser vista como uma atividade básica do aprendizado humano. O conhecimento da biologia cresceu mas a organização de todas as espécies conhecidas ainda é representada em *clustering* hierárquico, médicos definem categorias de tumores baseados em suas propriedades, astrônomos agrupam galáxias por seus formatos, empresas observam os usuários de seus produtos e os agrupam baseados em seus comportamentos. Esses são alguns dos muitos exemplos de *clustering* (HENNIG et al., 2016).

Podemos definir os passos requeridos para a análise de *clusters* como: escolher os objetos aos quais se deseja agrupar, escolher os valores a serem utilizados, padronizar esses valores, escolher a medida de similaridade, escolher o método de *clustering*, decidir o número de *clusters* (alguns métodos exigem o número *clusters* como parâmetro de entrada) e finalmente testar, interpretar e validar os *clusters* (MILLIGAN, 1996).

Diferentes estratégias de *clustering* atendem diferentes objetivos e é importante ressaltar que sem a compreensão dos objetivos não podemos afirmar que exista um algoritmo melhor que outro. A tarefa de selecionar um método e interpretar seu resultado implica no bom entendimento dos dados para que um método apropriado possa ser usado (HENNIG; LIAO, 2013) (LUXBURG; WILLIAMSON; GUYON, 2012).

2.1.2. Extração de Atributos

Uma vez escolhidos os objetos que desejamos analisar é preciso que estes sejam representados de forma apropriada para que sejam computacionalmente úteis. Neste trabalho os objetos de interesse serão imagens médicas. A representação de

imagens pode se dar de diferentes formas. Os atributos de uma imagem podem corresponder a informações de diferentes tipos, e a extração desses atributos é necessária para grande parte dos estudos que envolvem análise de imagens. É possível, por exemplo, representar uma imagem por características internas das regiões de interesse caracterizando os pixels que formam tais regiões.

Em algumas bases de imagens médicas a serem apresentadas em 4.1.1 usaremos descritores de textura. Embora não haja uma definição formal do que seja textura, podemos entender de forma intuitiva que um descritor de tal tipo consiga quantificar elementos como a suavidade e a regularidade dos padrões contidos na imagem (GONZALEZ; WOODS, 2008).

Os descritores utilizados aqui são conhecidos como descritores de textura de Haralick (HARALICK; SHANMUGAM; DINSTEN, 1973). Esses descritores são baseados em matrizes de co-ocorrência. Uma matriz de co-ocorrência é uma matriz G onde seus valores g_{ij} representam a frequência com que um par de pixels com intensidades z_i e z_j ocorre em uma imagem f na posição especificada por um operador Q para L níveis possíveis de intensidade. Esse operador Q pode ser associado a um ângulo, uma distância ou uma determinada forma de deslocamento horizontal ou vertical. Cada elemento dessa matriz pode então ser interpretado como uma função que determina a probabilidade de ocorrência de dois níveis z_i e z_j para pares de pixels nas direções especificadas por Q . Assim, o número total de pares de pixels que satisfazem Q é igual à soma de elementos de G e $p_{ij} = g_{ij}/n$ é uma estimativa da probabilidade de que um par de pontos que satisfazem Q terá os valores (z_i, z_j) . A soma dessas probabilidades é igual a 1 e pode ser vista na equação (1).

$$\sum_{i=1}^K \sum_{j=1}^K p_{ij} = 1 \quad (1)$$

onde K é o número de linhas (ou colunas) de G . Essas noções são utilizadas nos descritores de Haralick. Nas imagens estudadas neste trabalho foi utilizada uma combinação dos descritores que caracterizam Homogeneidade, Variância e Entropia. O modo como cada um é calculado pode ser visto na Tabela 1.

Tabela 1 - Equações dos descritores de Haralick utilizados

Descritor	Fórmula
Homogeneidade	$\sum_{i=1}^K \sum_{j=1}^K \frac{p_{ij}}{1 + i - j }$
Variância	$\sum_{i=1}^K \sum_{j=1}^K (i - j)^2 p_{ij}$
Entropia	$- \sum_{i=1}^K \sum_{j=1}^K p_{ij} \log_2 p_{ij}$

Fonte: Gonzalez e Woods (2008)

O descritor de Homogeneidade mede a distribuição de pixels a partir da proximidade da distribuição dos elementos em relação à diagonal de G . A Variância (ou contraste) mede o contraste das intensidades de um pixel e seu vizinho. A Entropia mede quão aleatório ("desordem") são os elementos da matriz.

As imagens histopatológicas de tecido epitelial do colo do útero que serão descritas em 4.1.1 foram caracterizadas usando uma abordagem desenvolvida em (MIRANDA et al., 2012). O descritor baseia-se em atributos topológicos capazes de descrever diferentes estruturas dos tecidos. Os núcleos das células são segmentados através de um *pipeline* de operadores morfológicos e a transformada de *watershed*. A partir dos núcleos das células é obtido então o diagrama de *Voronoi* que define fronteiras equidistantes entre os núcleos. Através do diagrama de *Voronoi* pode ser obtido então um grafo baseado na Triangulação de *Delaunay* que forma triângulos estabelecendo arestas a cada tripla de pontos. As métricas utilizadas se referem a medidas obtidas a partir desse grafo, sendo elas a média das áreas dos triângulos formados, a entropia e o grau médio do grafo. O objetivo aqui, no entanto, está em comparar o algoritmo hierárquico não supervisionado baseado na caminhada do turista com os algoritmos clássicos através de tais imagens, e não na análise das imagens propriamente dita. Mais informações sobre essas métricas podem ser vistas em (MIRANDA et al., 2012).

2.1.3. Medidas de Distância

Uma vez definidas as variáveis que formam o vetor de características que descrevem cada item, o próximo passo então é determinar uma medida que defina quão similar um item é de outro.

Para que um algoritmo de *clustering* possa agrupar observações similares ele usa o conceito de dissimilaridade. Matematicamente a dissimilaridade é uma função $d: X^2 \mapsto \mathbb{R}$ onde X é o espaço e $d(x, y) = d(y, x) \geq 0$ para $x, y \in X$.

A análise de *cluster* usualmente começa com uma matriz $n \times n$ e seus elementos refletem uma medida quantitativa de proximidade, mais comumente referida como dissimilaridade, distância ou similaridade (EVERITT et al., 2011). Dois itens são próximos quando sua dissimilaridade ou distância é pequena. As medidas de dissimilaridade podem ser divididas em medidas de distância e medidas de correlação. A medida de distância mais comumente usada é a distância euclidiana vista na equação (2).

$$d_{ij} = \left[\sum_{k=1}^p (x_{ik} - x_{jk})^2 \right]^{1/2} \quad (2)$$

onde x_{ik} e x_{jk} são respectivamente a k -ésima variável das observações i e j no espaço de p dimensões. Essa distância pode ser interpretada como a distância física entre dois pontos $x'_i = (x_{i1}, \dots, x_{ip})$ e $x'_j = (x_{j1}, \dots, x_{jp})$ no espaço euclidiano.

Existem muitas outras medidas de distância tais como a *City Block*, *Minkowski*, *Canberra*. Elas estão resumidas da Tabela 2.

Tabela 2 - Equações de algumas medidas de distância

Medida	Equação
City Block	$d_{ij} = \sum_{k=1}^p x_{ik} - x_{jk} $
Minkowski	$d_{ij} = \left(\sum_{k=1}^p x_{ik} - x_{jk} ^r \right)^{1/r} \quad (r \geq 1)$
Canberra	$d_{ij} = \begin{cases} 0 & \text{para } x_{ik} = x_{jk} = 0 \\ \sum_{k=1}^p \frac{ x_{ik} - x_{jk} }{ x_{ik} + x_{jk} } & \text{para } x_{ik} \neq x_{jk} \neq 0 \end{cases}$

Fonte: Everitt et al. (2011)

2.1.4. Métodos de *Clustering*

O próximo passo é o processo de *clustering* propriamente dito. Dado um conjunto de dados $D = \{x_1, x_2, \dots, x_n\}$ contendo n itens, o processo de *clustering* consiste em agrupá-los em K subconjuntos disjuntos de D , chamados $C_1, C_2, \dots, C_K$. Se um determinado item x_i pertence ao *cluster* C_K então dizemos que o rótulo de x_i é k . É importante ressaltar que *clustering* pode significar diferentes coisas em diferentes contextos. Não existe uma definição única ou o que seria a melhor maneira de executar um processo de *clustering*.

Diferentes abordagens de análise de *cluster* diferem em como os *clusters* se relacionam entre si e com o conjunto de dados. A divisão primária clássica dos métodos de *clustering* é entre os métodos particionais e os hierárquicos (JAIN; DUBES, 1988).

Os métodos particionais encontram partições nos dados na qual $C_j \cap C_k = \emptyset$ para $j \neq k$. Alguns métodos que tratam de *outliers* ou ruído podem ainda não atribuir tais itens a um *cluster*. Já os métodos hierárquicos criam uma sequência de partições $C = U_{j=1}^m C_j$ onde $C_j, i = j, \dots, m$ são partições com $K_1 = |C_1| > \dots > K_m = |C_m|$ onde $|C|$ é o número de elementos em C . Os conjuntos nos níveis mais baixos são então partições dos conjuntos nos níveis mais altos. A maneira mais natural de mostrar cada um desses passos é através de um dendrograma (árvore) que mostra como as amostras foram agrupadas. Cada nível da árvore revela quais agrupamentos foram feitos.

Existem diversas outras maneiras de categorizar os algoritmos de *clustering*. Algumas revisões recentes criaram novas subcategorias para abranger os avanços da área (KOTSIANTIS; PINTELAS, 2004), (XU; WUNSCH, 2005), (XU; TIAN, 2015). Xu e Tian (2015) dividem as abordagens entre algoritmos tradicionais e algoritmos modernos. Os algoritmos tradicionais são subdivididos em 9 categorias, sendo elas: i. baseado em partições; ii. Baseado em hierarquia; iii. Baseado em lógica fuzzy; iv. Baseado na distribuição dos dados; v. Baseado em densidade; vi. Baseado em teoria dos grafos; vii. Baseado em *grid*; viii. Baseado em teoria dos fractais; ix. Baseado em modelo. Já os algoritmos modernos são subdivididos em 10 categorias: i. Baseado em *Kernel*; ii. Baseado em um conjunto de técnicas (*ensemble*); iii. Baseado em inteligência de enxames (*Swarm Intelligence*); iv. Baseado em mecânica quântica; v. Baseado em teoria espectral de grafos; vi. Baseado em propagação por afinidade; vii.

Baseado em densidade e distância; viii. Algoritmos para dados espaciais; ix. Algoritmos para *stream* de dados e finalmente x. algoritmos para dados de grande escala.

Hennig et al. (2016) organiza os métodos em quatro grandes abordagens da análise de *clusters*. A primeira abordagem abrange métodos de otimização, ou seja, métodos que procuram otimizar uma função que descreva quão bem os dados se agrupam em torno de um centroide. O método mais popular nessa categoria e talvez o mais popular método de *clustering* é o *K-means*. Outra abordagem inclui os métodos baseados em dissimilaridade. Eles formalizam a ideia de que objetos em um mesmo *cluster* devem ser similares enquanto que objetos em *clusters* diferentes devem ser dissimilares. Métodos hierárquicos tradicionais são os exemplos mais conhecidos. A terceira abordagem abrange métodos baseados em probabilidade, ou seja, modelos de mistura (*mixture model*) e modelos de particionamento. O último tipo são métodos não paramétricos que estimam a densidade, ou seja, ao invés de especificar modelos para os *clusters*, eles os identificam como ilhas de alta densidade nos dados.

Aqui, para introduzir conceitos fundamentais que serão utilizados nos métodos apresentados iremos utilizar a divisão adotada em Hennig et al. (2016). Apresentaremos brevemente o K-means como método baseado em centroide dada sua importância e em seguida daremos ênfase nos métodos baseados em dissimilaridade, em especial os métodos hierárquicos, introduzindo os algoritmos que serão usados na comparação com o método hierárquico baseado na caminhada do turista; e por último apresentaremos métodos não paramétricos, mais especificamente sua subdivisão de métodos inspirados em fenômenos da natureza.

2.1.4.1. Métodos Baseados em Centróide

Dentro das abordagens de otimização temos os métodos baseados em um centroide. Neste caso o número de *clusters* é fixo e conhecido (quando não é conhecido existem algumas estratégias para estimar o número de *clusters* em um conjunto de dados). O principal exemplo é o K-means (MACQUEEN, 1967). A ideia central é encontrar K centroides que representem todos os objetos. Formalmente o *K-means* tenta minimizar a função vista em (3).

$$S(D, m_1, \dots, m_K) = \sum_{i=1}^n d(x_i, m_{c(i)}) \quad (3)$$

onde $c(i) = \operatorname{argmin}_j d(x_i, m_j)$, $i = 1, \dots, n$ e $j \in \{1, \dots, K\}$ escolhendo os centroides $m_1, \dots, m_K$ apropriados e d é a medida de dissimilaridade. Essa abordagem exige que todos os itens de um *cluster* estejam próximos de um centroide e de tal forma restringe o formato do *cluster* prejudicando o algoritmo caso o *cluster* tenha um formato alongado ou não linear.

2.1.4.2. Métodos Baseados em Dissimilaridade

Na abordagem de métodos baseados em dissimilaridade temos o método hierárquico. O método hierárquico pode ser de dois tipos: aglomerativo ou divisivo. No método aglomerativo (*bottom-up*) o procedimento inicia com n grupos (cada grupo é inicialmente formado por apenas um item) e a cada passo vai agrupando mais itens até que no último nível todos os itens estejam em um único *cluster*. Já no divisivo (*top-down*) acontece o contrário: o procedimento começa com todos os itens unidos em um único grupo e sucessivamente vão sendo divididos. O método aglomerativo costuma ser mais usado. (DUDA; HART; STORK, 2001).

Existem vários métodos aglomerativos que diferem basicamente em como é computada a dissimilaridade entre dois *clusters*. Os métodos mais conhecidos são: *single link*, onde a distância é calculada da seguinte forma: $D(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$, ou seja, é considerado o item de C_1 mais próximo de um item em C_2 ; e o *complete link*, calculado usando $D(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} d(x_1, x_2)$, ou seja, é considerado o item de C_1 mais distante de um item de C_2 . Uma outra forma de entender os algoritmos *single link* e *complete link* é vê-los como métodos que vão atualizando uma matriz de proximidade (KING, 1967). O algoritmo é nesse caso uma estratégia de *clustering* hierárquico aglomerativo que durante sua operação vai apagando linhas e colunas de uma determinada matriz de proximidade à medida que *clusters* antigos são unidos em *clusters* novos.

Usando a notação apresentada por Jain e Dubes (1988), a matriz de proximidade $n \times n$ é preenchida com as similaridades $D = [d(i, j)]$. Os *clusters* recebem um número $0, 1, \dots, (n - 1)$ e $L(k)$ é o nível do k -ésimo *cluster*. Um *cluster*

de número m é chamado de (m) e a proximidade entre os *cluster* (r) e (s) é dada por $d[(r), (s)]$.

O algoritmo inicia com *clusters* disjuntos de nível $L(0)$ e um número $m = 0$ que representa os *clusters*. No passo seguinte ele encontra o par de *clusters* menos dissimilar de acordo com a regra apresentada em (4).

$$d[(r), (s)] = \min\{d[(i), (j)]\} \quad (4)$$

onde $\{(r), (s)\}$ seria um determinado par de *clusters* e o *min* (mínimo) considera todos os pares de *clusters* disponíveis no nível atual. Depois, o número m é atualizado $(m + 1)$ e os *clusters* (r) e (s) são unidos em um novo *cluster* para formar a nova conformação de *clusters* m . O nível dessa conformação é então atualizado seguindo a equação (5).

$$L(m) = d[(r), (s)] \quad (5)$$

No último passo a matriz de proximidade D é atualizada deletando as linhas e colunas correspondentes aos *clusters* (r) e (s) e adicionando uma linha e coluna correspondente ao novo *cluster* formado. A proximidade entre o *cluster* novo (r, s) e o *cluster* antigo (k) é definida para o single-link seguindo a equação (6).

$$d[(k), (r, s)] = \min\{d[(k), (r)], d[(k), (s)]\} \quad (6)$$

E para o *complete link*, a equação (7)

$$d[(k), (r, s)] = \max\{d[(k), (r)], d[(k), (s)]\} \quad (7)$$

Enquanto o *single link* e *complete link* usam o mínimo e máximo das dissimilaridades entre os pares $\{(k), (r)\}$ e $\{(k), (s)\}$, outros métodos adotam diferentes combinações dessas distâncias. Uma fórmula genérica pode ser vista em (8) (LANCE; WILLIAMS, 1967), (JAIN; DUBES, 1988)

$$d[(k), (r, s)] = \alpha_r d[(k), (r)] + \alpha_s d[(k), (s)] + \beta d[(r), (s)] + \gamma |d[(k), (r)] - d[(k), (s)]| \quad (8)$$

A Tabela 3 mostra os valores dos parâmetros α_r , α_s , β e γ para os algoritmos hierárquicos usados na comparação com o algoritmo hierárquico baseado na caminhada do turista.

Tabela 3 - Valores dos parâmetros para os métodos hierárquicos

Método	α_r	α_s	β	γ
Single link	1/2	1/2	0	-1/2
Complete link	1/2	1/2	0	1/2
UPGMA	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	0	0
WPGMA	1/2	1/2	0	0
UPGMC	$\frac{n_r}{n_r + n_s}$	$\frac{n_s}{n_r + n_s}$	$\frac{-n_r n_s}{(n_r + n_s)^2}$	0
WPGMC	1/2	1/2	-1/4	0
Ward	$\frac{n_r + n_k}{n_r + n_s + n_k}$	$\frac{n_s + n_k}{n_r + n_s + n_k}$	$\frac{-n_k}{n_r + n_s + n_k}$	0

Fonte: Jain e Dubes (1988)

A sigla PGM presente no nome de alguns métodos significa *pair group method*, ou seja, se refere a um método que agrupa pares. As siglas U e W se referem respectivamente a *unweighted* e *weighted* (sem peso e com peso). Um método sem peso considera igualmente cada item de um *cluster* independentemente da estrutura do dendrograma. Um método com peso "pesa" todos os *clusters* da mesma maneira e assim itens em *clusters* menores acabam tendo um peso maior que itens em *clusters* maiores. As letras A e C se referem a *arithmetic averages* (média aritmética) e *centroids* (centroides). Enquanto o single-link procura o par de itens mais próximos em dois *clusters* e o complete-link procura o par mais distante, o UPGMA (*Unweighted Pair Group Method usando Arithmetic Averages*) e o WPGMA (*Weighted Pair Group Method usando Arithmetic Averages*) usam médias aritméticas das dissimilaridades. Os métodos baseados em centroides avaliam a dissimilaridade entre dois *clusters* pela distância entre seus centroides.

O último método, Ward ou variância mínima, é baseado na ideia de erro quadrático. Dado um conjunto n de padrões e um espaço d -dimensional, $x_{ij}^{(k)}$ é o valor da característica j do padrão i quando o padrão i está no *cluster* k para i de 1 a

n_k e j de 1 a d . O centroide do *cluster* k é dado por $[m_1^{(k)}, \dots, m_d^{(k)}]$ (equação (9)) e é o centro do *cluster* ou a média dos n_k padrões do *cluster* k .

$$m_j^{(k)} = (1/n_k) \sum_{i=1}^{n_k} x_{ij}^{(k)} \quad (9)$$

O erro quadrático para o *cluster* k é a soma das distâncias quadráticas para o centroide para todos os padrões do *cluster* k (equação (10)).

$$e_k^2 = \sum_{i=1}^{n_k} \sum_{j=1}^d [x_{ij}^{(k)} - m_j^{(k)}]^2 \quad (10)$$

O erro quadrático para todos os *clusters* é a soma dos erros quadráticos para cada *cluster* (equação (11)).

$$E_K^2 = \sum_{k=1}^K e_k^2 \quad (11)$$

O método une o par de *clusters* que minimiza ΔE_{pq}^2 (a mudança em E_K^2 causada pela fusão dos *clusters* p e q no *cluster* t).

2.1.4.3. Métodos Não Paramétricos Inspirados na Natureza

A última abordagem apresentada aqui inclui técnicas que se "auto organizam". Elas implementam o processo de *clustering* como um processo dinâmico em que os objetos se agrupam de acordo com certas regras e são guiadas pela vizinhança e atraídas pela densidade. A técnica baseada na caminhada do turista pode ser incluída nesta abordagem. Um tipo importante dessas técnicas são aquelas baseadas em fenômenos e movimentos da natureza. Hennig et al. (2016) divide esse tipo de estratégia em duas: as diretamente derivadas de fenômenos da natureza como as baseadas no comportamento auto organizador de algumas formigas, pássaros e outros animais, e aquelas que veem o problema de *clustering* como um problema de otimização e aplicam heurísticas inspiradas na natureza.

Esta categoria de algoritmos inclui aqueles que usam comportamentos emergentes. Um comportamento emergente, na teoria dos sistemas complexos, é um processo que forma padrões complexos a partir de interações simples. A emergência é o surgimento de uma estrutura, de padrões e propriedades durante o processo da auto-organização de um sistema complexo (GOLDSTEIN, 1999). Exemplos de auto-

organização e emergência podem ser vistos em muitas áreas da biologia e da física, como por exemplo o comportamento exibido por colônias de insetos. A auto-organização pode ser definida como um processo dinâmico e adaptativo onde os sistemas adquirem e mantêm estruturas por si próprios, sem controle externo (WOLF; HOLVOET, 2005). Um algoritmo de *clustering* inspirado nesse conceito é implementado baseado em regras locais, que definem interações individuais entre os elementos do sistema (HENNIG et al., 2016). Algum aspecto do sistema é progressivamente ajustado e o estado final do sistema é a solução do *clustering*.

Os algoritmos existentes que se auto organizam se diferenciam na maneira como as regras definem as interações locais. Um dos algoritmos que se auto organiza mais conhecido é o *Self-Organizing Maps* (SOM) (KOHONEN, 2001). Ele apresenta uma forma de rede neural artificial para aprendizado não supervisionado. Um mapa auto organizável consiste em um conjunto de neurônios que são arranjados em um *grid* retangular ou hexagonal. Cada unidade neuronal no *grid* é associada com um vetor numérico. O processo de aprendizado de um SOM envolve o ajuste desses vetores de maneira a prover uma representação apropriada dos dados de entrada.

Uma outra técnica baseada em auto-organização é o *clustering* baseado em colônias de formigas. Essa técnica se inspira na observação do comportamento de emergência de algumas espécies de formigas como a formação de cemitérios (CORNE; DORIGO; GLOVER, 1999). Diferentemente do SOM, a técnica baseada em colônias de formigas não requer dados numéricos em formato de vetor, pois pode funcionar diretamente sobre uma matriz de dissimilaridade.

Um outro exemplo de algoritmo é o por propagação de afinidade. Esse algoritmo foi proposto em (DUECK; FREY, 2007) e é explicado em detalhes na seção de trabalhos correlatos.

2.2. Caminhadas em dados

Um caminho pode ser definido como uma sequência ordenada $W = \{(v_1, v_2), (v_2, v_3), \dots, (v_{K-1}, v_K)\}$, onde $v_1, \dots, v_K \in V$, $K \geq 2$ e $\forall k \in \{2, \dots, K\} : (v_{k-1}, v_k) \in E$. No caso v_1 é a origem da caminhada enquanto que v_K é seu destino.

O uso de caminhadas em dados como uma estratégia para caracterizar tais dados tem recebido cada vez mais atenção (YANG, 2005), (KINOCHI et al., 2002).

As caminhadas aleatórias são mais estudadas, porém as caminhadas determinísticas também vêm despertando um interesse científico maior.

2.2.1. Caminhadas aleatórias

Uma caminhada aleatória é a formalização matemática de uma trajetória que consiste de sucessivos passos aleatórios (PEARSON, 1905). Este conceito tem sido utilizado para descrever muitos fenômenos naturais, assim como para resolver diferentes problemas da engenharia. Muitas dessas soluções envolvem o reconhecimento de padrões (GORI; MAGGINI; SARTI, 2005).

Dado um grafo $G = (V, E)$ e um ponto inicial $v \in V$, escolhemos um de seus vizinhos de forma aleatória e passamos então para esse novo ponto, assim, selecionamos um vizinho desse novo ponto, novamente de forma aleatória, e assim sucessivamente. A sequência aleatória de vértices selecionados dessa maneira é uma caminhada aleatória no grafo.

Dois conceitos encontrados na caminhada aleatória podem ser abstraídos e usados para a compreensão de conceitos similares que serão vistos na caminhada do turista. O primeiro é o conceito de estado absorvente. No estado absorvente há uma probabilidade 1 de que a caminhada siga para ele mesmo, ou seja, uma vez que um estado absorvente seja atingido em uma caminhada aleatória o caminhante fica preso para sempre. Outro conceito é um tipo de caminhada aleatória conhecida como caminhada auto repulsiva ou auto evitante. Uma caminhada auto repulsiva em um grafo G é um caminho que não visita nenhum vértice mais de uma vez (FLORY, 1953), (MADRAS; SLADE, 2013). Usualmente as caminhadas auto repulsivas são executadas em *lattices* com um número limitado de direções possíveis para o caminhante. Essas caminhadas não podem ser markovianas pois como um nó não pode ser visitado mais de uma vez, precisamos conhecer o caminho percorrido (passado) para saber os possíveis estados que a caminhada pode percorrer (futuro).

2.2.2. Caminhadas Determinísticas

Um outro tipo de caminhada é a determinística. Diferentemente da caminhada aleatória, ela é gerada por um movimento determinístico de uma partícula sobre o

meio distribuído sobre os vértices de um grafo (BUNIMOVICH, 2000). Existem diversos tipos de caminhadas determinísticas e aqui iremos focar em uma caminhada determinística parcialmente auto repulsiva conhecida como caminhada do turista.

2.2.2.1. *Caminhada do Turista*

A caminhada do turista pode ser conceitualizada como um caminhante (turista) que deseja visitar várias cidades (itens de dados) em um mapa de p dimensões que representa o conjunto de dados. A cada passo discreto o turista segue uma simples regra determinística: ele visita a cidade mais próxima que não tenha sido visitada nos μ passos anteriores, ou seja, o turista faz uma caminhada determinística parcialmente auto repulsiva no conjunto de dados, onde o fator auto repulsivo é limitado pelo tamanho da memória $\mu - 1$. Esse valor pode ser entendido como a força repulsiva que emana das cidades dentro da janela de memória e que impede que o turista as visite dentro desse intervalo de tempo. Assim, não é possível que uma trajetória interceda a si própria dentro da janela de memória. Embora seja uma regra simples, foi demonstrado que esse movimento possui um comportamento complexo quando $\mu > 1$ (STANLEY; BULDYREV, 2001), (LIMA; MARTINEZ; KINOCHI, 2001).

O comportamento do turista depende fortemente da configuração do conjunto de dados e do ponto inicial da caminhada. Cada caminhada é composta por dois termos: a parte inicial transiente de tamanho t e um ciclo (atrator) de período c .

O atrator é uma parte da caminhada que começa e termina no mesmo ponto. É importante ressaltar que o fato de um ponto ser visitado mais de uma vez não caracteriza um atrator. O atrator só é caracterizado quando o caminhante fica preso neste ciclo.

Na Figura 1 podemos ver uma caminhada do turista de memória $\mu = 4$ iniciada no vértice identificado com o número 1. Do vértice 1 o turista passa para o vértice mais próximo não visitado, no caso, o vértice identificado com o número 2 (a). Neste momento, embora o vértice 1 seja o mais próximo do turista (que está em 2), ele não pode visitá-lo pois este encontra-se dentro da memória $\mu = 4$, como pode ser visto no passo 1 da tabela explicativa da Figura 1 (h). O vértice mais próximo que não está na memória é o vértice 9. A caminhada segue na ordem 1, 2, 9, 4, 6, 7, 3 até

que chega ao vértice 9 novamente. Note-se que quando o turista está no vértice 9, o vértice 4 já não está mais na memória e assim ele pode visitá-lo. Após visitar o 4 o turista passa então ao vértice 6 e passa a repetir a caminhada configurando um atrator de tamanho 5 que segue 9, 4, 6, 7, 3 e 9 novamente.

Figura 1 - Passo a passo de uma caminhada do turista com memória $\mu = 4$

(a) Passo 1

(b) Passo 2

(c) Passo 3

(d) Passo 4

(e) Passo 5

(f) Passo 6

(g) Passo 7

Fonte: Elaborada pelo autor.

Passo	Memória μ	Caminhada
1	(1, -, -, -)	1
2	(1, 2, -, -)	1, 2
3	(1, 2, 9, -)	1, 2, 9
4	(1, 2, 9, 4)	1, 2, 9, 4
5	(2, 9, 4, 6)	1, 2, 9, 4, 6
6	(9, 4, 6, 7)	1, 2, 9, 4, 6, 7
7	(4, 6, 7, 3)	1, 2, 9, 4, 6, 7, 3
8	(6, 7, 3, 9)	1, 2, 9, 4, 6, 7, 3, 9
9	(7, 3, 9, 4)	1, 2, 9, 4, 6, 7, 3, 9, 4
10	(3, 9, 4, 6)	1, 2, 9, 4, 6, 7, 3, 9, 4, 6
11	(9, 4, 6, 7)	1, 2, 9, 4, 6, 7, 3, 9, 4, 6, 7
12	(4, 6, 7, 3)	1, 2, 9, 4, 6, 7, 3, 9, 4, 6, 7, 3
13	(6, 7, 3, 9)	1, 2, 9, 4, 6, 7, 3, 9, 4, 6, 7, 3, 9

(h) Passos de uma Caminhada do
Turista com Memória $\mu = 4$

A abordagem usada aqui para a detecção do atrator e por consequência do fim da caminhada é a abordada em (BACKES et al., 2010). A ideia de que um determinado item seja visitado mais de uma vez não é suficiente para afirmar que temos um atrator. Dependendo da configuração dos dados, uma caminhada pode se interceder uma ou mais vezes sem que isso caracterize um atrator. A abordagem adotada é de identificar se o item atual da caminhada já foi visitado 3 vezes, em caso afirmativo, é verificado se os passos anteriores foram repetidos como visto no passo 13 do quadro explicativo da Figura 1 (h). Uma vez que foi detectado que o item 9 apareceu pela terceira vez na caminhada é então verificado se os passos que antecedem o 9 até sua última aparição são os mesmos passos que antecedem os passos entre a penúltima aparição e sua antepenúltima aparição. Caso positivo temos então o atrator, ou seja, o caminhante permaneceria indefinidamente preso neste ciclo.

O turista pode visitar qualquer outro lugar que não esteja na janela de memória. Com o aumento de μ há uma grande chance de que o caminhante comece a fazer longas caminhadas uma vez que sua vizinhança está provavelmente contida na grande janela de memória e assim proibida de receber novas visitas.

Como já visto, a caminhada do turista possui uma dinâmica que permite explorar os dados e assim caracterizá-los. De modo simplificado, isso se dá por sua característica de mover-se, quando possível, para o vizinho mais próximo, revelando assim a similaridade.

Uma vantagem importante dessa estratégia é que não é necessário usar a matriz de distância uma vez que a tabela de ordenamento de vizinhos tenha sido criada para cada nó. Isso se torna interessante dependendo de como as relações entre

os dados forem estabelecidas. Caso sejam relações qualitativas, a tabela ordenada de vizinhos torna o processo mais intuitivo. A caminhada impõe também um critério de parada automático e isso reflete no algoritmo particional proposto neste trabalho. Diferentemente de outros algoritmos em que o usuário deve definir um determinado número de iterações ou critérios para o refinamento do algoritmo que pode se dar em um número indeterminado, o algoritmo baseado na caminhada do turista embute o critério de parada automático.

2.3. Validação de *Clusters*

Uma vez aplicado nosso método de *clustering* é importante validar o resultado pois os *clusters* encontrados não necessariamente possuem algum significado. A validação de um *cluster* é um problema complexo e mais difícil do que avaliar a predição de uma regressão em aprendizado supervisionado, dado que em análise de *clusters* normalmente não temos informações sobre a classe dos dados (HENNIG et al., 2016).

Existem diversos índices que quantificam a qualidade do processo de *clustering*. Esses índices podem ser usados tanto para medir a qualidade do *clustering*, para seleccionar os parâmetros ideais de um mesmo método ou ainda para comparar o resultado de diferentes métodos.

Os critérios usados para validação podem ser divididos em externos, internos e relativos (JAIN; DUBES, 1988). O critério externo compara o resultado do *clustering* com uma estrutura pré-definida. Essa estrutura pode ser originada de um outro método de *clustering* ou ainda dos rótulos já conhecidos dos dados (o que não é o usual em aprendizado não supervisionado). O critério interno avalia em que medida o resultado se ajusta aos dados com base nos próprios dados. O critério relativo decide qual de duas estruturas é a melhor em algum sentido. O critério interno e o relativo não são necessariamente classes distintas de índices, uma vez que o critério relativo pode ser obtido apenas com os próprios dados.

É importante ressaltar que não existe um resultado de *clustering* que possa ser identificado como o melhor. Essa definição requer um conhecimento prévio do pesquisador que deve escolher então o método e sua validação. Outra questão relevante é que os índices devem procurar um balanceamento entre a homogeneidade

dentro de um mesmo *cluster* e a separação entre diferentes *clusters*. Altos valores de K (número de *clusters*) normalmente aumentam a homogeneidade dos itens dentro de um mesmo *cluster* mas diminuem a separação entre *clusters*. Dependendo da aplicação, a preferência pode ser por homogeneidade maior (muito útil por exemplo para fazer buscas em bancos de dados) ou por uma separação maior (HENNIG et al., 2016).

O objetivo do processo de *clustering*, como já dito, é fazer com que objetos similares fiquem no mesmo *cluster* enquanto que objetos distintos em *clusters* diferentes, de modo que as medidas de validação baseadas em critérios internos são frequentemente baseadas nos conceitos de compactação e separação (AGGARWAL; REDDY, 2014). A compactação (ou coesão dos *cluster*) mede quão próximos são os objetos em um mesmo *cluster*. Uma variação pequena de valores dentro de um *cluster* indica uma boa compactação. A separação mede quão bem separado um *cluster* é de outro.

O procedimento mais usual para determinar a melhor partição e o número ideal de *clusters* de um conjunto de objetos começa com a definição dos algoritmos de *clustering* que serão aplicados ao conjunto de dados. Depois, para cada algoritmo, são usadas diferentes combinações de parâmetros para termos diferentes resultados. Podemos então computar os índices de validação interna para cada partição encontrada. Alguns dos índices que usam critérios internos mais usados são o coeficiente de silhueta, indicador Dunn, indicador Davies-Bouldin, *root-mean-square standard deviation* (RMSSD), *R-squared* (RS), *Modified Hubert Γ statistic* (Γ), entre muitos outros (AGGARWAL; REDDY, 2014).

2.3.1. Método da Silhueta

O método da silhueta (ROUSSEEUW, 1987) permite que um *cluster* seja representado por uma silhueta. Essa medida se baseia na comparação de sua compacticidade e separação. A largura média de silhueta permite uma avaliação da validade do *cluster* e tem sido usada para determinar o número apropriado de *clusters* em algoritmos particionais que exigem esse parâmetro.

No algoritmo particional proposto por este trabalho iremos usar o coeficiente de silhueta para a definição do melhor valor de tamanho de memória μ . O coeficiente

de silhueta mede a diferença da distância par a par entre *clusters* e dentro de um *cluster*. Para cada observação i é calculada a dissimilaridade média a_i entre i e todos os outros pontos do *cluster* que i pertence. Para todos os outros *clusters* C nos quais i não pertence, é calculada a dissimilaridade média $d(i, C)$ de i para todos os itens de C . O menor valor entre todos os resultados de $d(i, C)$ é definido como $b_i = \min_C(d(i, C))$. O valor b_i pode ser entendido como a dissimilaridade entre i e seus *clusters* vizinhos. Finalmente a largura da silhueta do item i é definida pela equação (12).

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (12)$$

Valores altos (próximos a 1) de S_i indicam que o item i está em um *cluster* apropriado. Valores de S_i próximos a 0 indicam que i está entre dois *clusters*. Valores negativos de S_i indicam que i provavelmente foi atribuído ao *cluster* errado.

2.3.2. Correlação Cofenética

Os algoritmos hierárquicos que serão estudados produzem como resultado árvores hierárquicas conhecidas como dendrogramas. Um dendrograma é uma representação gráfica de uma matriz ultramétrica (cofenética) e portando podemos comparar dendrogramas comparando suas matrizes cofenéticas (LAPOINTE; LEGENDRE, 1995) (SARAÇLI; DOGAN, N.; DOGAN, I., 2013). A similaridade entre os itens agrupados por diferentes métodos pode ser medida pela correlação desses itens. Considere os X_i dados que através de um determinado algoritmo de *clustering* hierárquico geraram o dendrograma T_i . Dada uma distância $x(i, j)$ entre o i – ésimo e j – ésimo itens (distância euclidiana no caso deste trabalho) e $t(i, j)$ a distância dendrogramática entre os pontos T_i e T_j , a distância cofenética entre dois objetos i e j é a altura na hierarquia de um dendrograma T_i em que os ramos aos quais esses objetos pertencem são unidos, ou seja, é a distância de um objeto ao outro dentro do dendrograma. Seja x a média de $x(i, j)$ e t a média de $t(i, j)$ o coeficiente de correlação cofenética é dada pela equação (13).

$$c = \frac{\sum_{i < j} (x(i, j) - x)(t(i, j) - t)}{\sqrt{[\sum_{i < j} (x(i, j) - x)^2][\sum_{i < j} (t(i, j) - t)^2]}} \quad (13)$$

A correlação cofenética pode medir quão fielmente um dendrograma preserva a distância par a par dos dados originais (SOKAL; ROHLF, 1962). Neste trabalho

comparamos a correlação da distância cogenética dos dendrogramas das diferentes técnicas com a distância real dos dados, assim como comparamos os métodos entre si. Isso nos permitirá investigar quão bem o dendrograma produzido pela técnica baseada na caminhada do turista representa os dados e como ela se compara com os dendrogramas de técnicas tradicionais. Essa investigação é especialmente interessante devido ao fato de que a estratégia baseada na caminhada do turista, diferentemente das técnicas tradicionais, não impõe uma estrutura binária à árvore.

3. Trabalhos Correlatos

Aqui apresentamos alguns trabalhos que de algum modo se relacionam com os temas abordados neste trabalho. Primeiramente vamos mostrar dois algoritmos de *clustering* particionais não paramétricos que usam interações locais para encontrar os *clusters*: o DBSCAN, que serve de base para vários algoritmos que têm surgido nos últimos anos, e o *Affinity Propagation*. Na sequência, apresentamos dois trabalhos que usam a caminhada do turista no Aprendizado de Máquina em uma abordagem supervisionada e em outra não supervisionada hierárquica, que servirá de ponto de partida deste trabalho.

3.1. DBSCAN

O *Density-Based Spatial Clustering of Application with Noise* (DBSCAN) (ESTER et al., 1996) é o algoritmo baseado em densidade mais conhecido e um dos algoritmos mais populares entre todos os algoritmos de *clustering*.

O algoritmo parte do princípio de que *clusters* são áreas densas separadas por regiões esparsas. A ideia central é que para cada item de um *cluster*, sua vizinhança, dentro de um determinado raio, deve ter um número mínimo de pontos. Para formalizar esse conceito de densidade é introduzido o conceito de *Eps-neighborhood* de um ponto p , chamado de $N_{Eps}(p)$ e é definido como $N_{Eps} = \{q \in D \mid dist(p, q) \leq Eps\}$, ou seja, a vizinhança inclui todos os pontos dentro desse raio limiar. A princípio, uma abordagem seria a de considerar que cada item em um *cluster* deve ter um número mínimo de itens em sua vizinhança (*Eps-neighborhood*), porém existem dois tipos de itens em um *cluster*: itens dentro do *cluster* e itens na borda. Os itens que ficam na borda possuem muito menos itens em sua vizinhança que um item dentro de um *cluster* e portanto uma solução é considerar que para cada item p em um *cluster* C deva existir um item q em C tal que p deve estar dentro do *Eps-neighborhood* de q e $N_{Eps}(q)$ possui pelo menos uma quantidade mínima de itens (MinPts). Essa definição é chamada de densidade alcançável diretamente (*directly density-reachable*), formalizada da seguinte maneira: um ponto p é densamente diretamente alcançável de um ponto q se $p \in N_{Eps}(q)$ e se $|N_{Eps}(q)| \geq MinPts$, ou seja, p está dentro da

vizinhança de q e q é um ponto dentro do *cluster* (não é borda e satisfaz uma vizinhança com mais pontos que o limiar MinPts), como visto na Figura 2.

Figura 2 – (a) Item na borda e dentro do cluster (b) Densidade alcançável diretamente

Fonte: Adaptado de Ester et al. (1996)

A próxima definição é a de densamente alcançável: um ponto p é densamente alcançável de um ponto q se existe um caminho de pontos $p_1, \dots, p_n$ sendo $p_1 = q$ e $p_n = p$ de modo que p_{i+1} é densamente diretamente alcançável de p_i , como visto na Figura 3 (a). A definição de densamente conectado consiste em um ponto p que é densamente conectado a um ponto q se existe um ponto o tal que p e q sejam densamente alcançáveis a partir de o , conforme a Figura 3 (b).

Figura 3 - (a) Densamente alcançável (b) Densamente conectado

Fonte: Adaptado de Ester et al. (1996)

Agora podemos definir o conceito de *cluster*. Seja D um conjunto de pontos, um *cluster* C é um subconjunto não vazio de D se ele satisfaz as seguintes condições: primeiramente $\forall p, q$: se $p \in C$ e q é densamente alcançável de p , então $q \in C$. E $\forall p, q \in C$: p é densamente conectado a q . Uma outra definição importante é a de ruído. Sejam $C_1, \dots, C_k$ os *clusters* dos conjuntos de dados D , definimos ruído como o conjunto de pontos em D que não pertencem a nenhum *cluster* C_i .

O algoritmo DBSCAN começa com um ponto arbitrário p e a partir dele identifica todos os pontos que são densamente alcançáveis a partir de p . Se p for um ponto dentro de um *cluster* (e não em uma borda), esse procedimento irá resultar em um *cluster*. Se p for um ponto na borda então nenhum ponto é densamente alcançável a partir de p e o algoritmo visita o próximo ponto. Idealmente os parâmetros de entrada Eps e $MinPts$ de cada *cluster* deveriam ser conhecidos. Não existe uma maneira fácil de se obter essa informação e o artigo propõe uma heurística para estimar esses valores.

A estratégia para determinar Eps e $MinPts$ é encontrar esses valores para o *cluster* menos denso e usar esses valores globalmente. A heurística proposta funciona da seguinte forma: seja d a distância de um ponto p para o seu k -ésimo vizinho mais próximo, então a vizinhança- d de p contém exatamente $k+1$ pontos para quase todos os pontos p . A vizinhança- d de p pode possuir mais que $k+1$ quando vários pontos tiverem exatamente a mesma distância d de p (o que seria muito raro). Para um dado k definimos a função k -dist que mapeia cada ponto p em um número real dado pela distância de cada ponto para seus k -ésimos vizinhos mais próximos. Quando os pontos são ordenados em ordem decrescente de valores k -dist, o gráfico de $\{1, \dots, n\}$ por k -dist nos dá uma ideia sobre a distribuição de densidade. Se escolhermos um ponto arbitrário p e selecionarmos o parâmetro Eps para k -dist(p) e selecionarmos o parâmetro $MinPts$ para k , todos os pontos com um valor igual ou menor de k -dist serão pontos dentro de um *cluster* (diferente dos pontos de borda). Se pudéssemos encontrar um ponto limiar com o máximo valor de k -dist para o *cluster* mais esparsos então teríamos os parâmetros ideais. O ponto limiar é o primeiro ponto no vale do gráfico do k -dist ordenado, como visto na Figura 4.

Figura 4 - Heurística para determinar Eps

Fonte: Ester et al. (1996)

Todos os pontos à esquerda do limiar (linha vertical), com valores altos de k -dist (os autores recomendam usar $k = 4$) são considerados ruído, e os demais pontos (à direita do limiar) são pontos em algum *cluster*. O artigo indica que valores de k -dist para $k > 4$ não diferem significativamente do gráfico de 4-dist, assim é usado sempre o valor 4 para MinPts. Já para determinar o valor de Eps os autores sugerem a seguinte interação: primeiramente é computado e exibido o gráfico de 4-dist, depois, se o usuário puder estimar o percentual de ruído, esse valor é inserido e o sistema sugere o limiar. O usuário pode aceitar o limiar sugerido ou selecionar outro ponto como limiar. O valor 4-dist do ponto limiar escolhido é usado como Eps.

Para avaliar a performance do DBSCAN os autores o compararam com o algoritmo CLARANS (NG; HAN, 1994) e usaram conjuntos de dados artificiais e o *benchmark* SEQUOIA 2000 (STONEBRAKER et al., 1993). Os autores justificam que como o DBSCAN e o CLARANS são algoritmos de tipos diferentes, eles não possuem medidas quantitativas comuns da acurácia da classificação e, portanto, a acurácia é avaliada com uma inspeção visual. Para essa inspeção foram usados alguns conjuntos de dados artificiais. Na Figura 5 podemos ver o resultado para o CLARANS.

Figura 5 - Resultado do CLARANS

Fonte: Ester et al. (1996)

O resultado para o DBSCAN pode ser visto na Figura 6.

Figura 6 - Resultado do DBSCAN

Fonte: Ester et al. (1996)

Como podemos ver, o DBSCAN encontra todos os *clusters* e detecta também o ruído, enquanto que o CLARANS divide o *cluster* se eles forem relativamente grandes ou se estiverem perto um do outro. Além disso, o CLARANS não possui o conceito de ruído e todos os pontos são associados ao seu *medoid* mais próximo.

Para o teste de eficiência foi usado o SEQUOIA 2000 *benchmark* (STONEBRASKER et al., 1993). Essa base possui dados reais de informações geográficas com 4 tipos de dados: dados *raster*, pontos, polígonos e grafos direcionados. Os dados com pontos contêm 62.584 nomes de cidades da Califórnia. Assim, foram extraídos para o teste subconjuntos dos dados de pontos do SEQUOIA contendo de 2% a 20% de todo o conjunto de dados. O tempo de processamento em segundos pode ser visto Tabela 4.

Tabela 4 - Tempo de processamento em segundos para o algoritmo DBSCAN e CLARANS

Número de pontos	1252	2503	3910	5213	6256	7820	8937	10.426	12.512
DBSCAN	3,1	6,7	11,3	16,0	17,8	24,5	28,2	32,7	41,7
CLARANS	758	3.026	6.845	11.745	18.029	29.826	39.265	60.540	80.638

Fonte: Ester et al. (1996)

Os resultados mostram que o DBSCAN é significativamente mais efetivo em descobrir *clusters* de formato arbitrário quando comparado ao CLARANS. Além disso, em termos de eficiência o DBSCAN supera o CLARANS por um fator de ao menos 100.

3.2. Affinity Propagation

O *Affinity Propagation* (FREY; DUECK, 2007) parte da tentativa de solucionar alguns problemas das estratégias baseadas em centroides. A estratégia tradicional consiste em usar os dados para encontrar um grupo de itens que sejam centrais e, assim, capazes de representar seus *clusters*. Dentro dessas estratégias o método mais comum é o *k-means*. A ideia dos algoritmos que usam "centros" (ou centroides) como representantes de seus *clusters* é começar com um conjunto de itens selecionados aleatoriamente e iterativamente refinar esse conjunto de modo a diminuir a soma dos erros quadráticos. Essas estratégias são muito sensíveis à seleção inicial de itens, razão pela qual é necessário repetir o procedimento várias vezes com diferentes itens iniciais para encontrar uma solução apropriada, e mesmo assim só funcionará se ao menos uma dessas seleções aleatórias obter itens próximos de uma boa solução.

O artigo propõe uma abordagem diferente e introduz um método que considera todos os itens simultaneamente como potenciais representantes de seus *clusters*. Esses representantes aqui são chamados de exemplares.

A ideia é de que cada item seja visto como um nó em um grafo e o método consiste em recursivamente transmitir mensagens pelas arestas do grafo até que um conjunto apropriado de exemplares surja.

Essas mensagens são atualizadas segundo fórmulas simples que procuram minimizar uma função de energia. A magnitude de cada mensagem reflete a afinidade (*affinity*) que um determinado item tem por escolher um outro item como seu exemplar, motivo pelo qual o método é chamado de propagação de afinidade (*affinity propagation*).

O algoritmo recebe como entrada uma coleção de valores reais que representam as similaridades entre itens $s(i, k)$, indicando quão bem um item k poderia ser um exemplar de um item i . Quando o objetivo é minimizar o erro quadrático, cada similaridade é atribuída a um erro quadrático negativo. Para os pontos x_i e x_k , temos que $s(i, k) = -\|x_i - x_k\|^2$. Ao invés de requerer que o número de clusters seja previamente especificado, o algoritmo recebe como entrada um número real $s(k, k)$ para cada item k , de modo que itens com valores maiores que $s(k, k)$ são mais prováveis de serem escolhidos como exemplares. Esses valores são chamados de preferências. O número de exemplares identificados é influenciado pelos valores das preferências usados como entrada, mas também surge durante o

procedimento de troca de mensagens. Se a princípio todos os itens são igualmente apropriados a serem exemplares então essas preferências devem receber o mesmo valor.

Existem dois tipos de mensagens trocadas entre os itens e cada uma tem um mecanismo diferente de competição. Mensagens podem ser combinadas a qualquer estágio para decidir quais itens serão exemplares, e para todos os outros itens, a qual exemplar eles pertencem. A responsabilidade (*responsibility*) $r(i, k)$ enviada de um item i a um candidato a exemplar k , reflete a evidência acumulada de quão apropriado o item k é de servir como exemplar do item i levando em consideração outros potenciais exemplares para o item i . A disponibilidade (*availability*) $a(i, k)$ enviada do candidato a exemplar k para o item i reflete a evidência acumulada de quão apropriado seria o ponto i escolher o ponto k como seu exemplar, levando em consideração outros itens em que o item k possa ser um exemplar.

No início do algoritmo as disponibilidades são inicializadas com 0: $a(i, k) = 0$. Depois, as responsabilidades são computadas usando a equação (14).

$$r(i, k) \leftarrow s(i, k) - \max_{k' \neq k} \{a(i, k') + s(i, k')\} \quad (14)$$

Na primeira iteração, devido ao fato de as disponibilidades serem 0, $r(i, k)$ recebe a similaridade entre o item i e o item k como seu exemplar menos a maior das similaridades entre o item i e outros candidatos a exemplar. Essa atualização competitiva é guiada pelos dados e não leva em consideração quantos outros itens favorecem cada candidato a exemplar. Nas próximas iterações, quando alguns itens são efetivamente associados a outros exemplares, a disponibilidade assume valores negativos conforme a regra de atualização. Essas disponibilidades negativas diminuem os valores efetivos de algumas similaridades $s(i, k')$ e assim removendo os candidatos a exemplares da competição.

A atualização de responsabilidade apresentada acima permite que todos os candidatos a exemplares compitam pelos itens. A equação (15) é a atualização da disponibilidade e coleta evidência dos itens de como cada candidato a exemplar seria um bom exemplar.

$$a(i, k) \leftarrow \min_{i' \notin \{i, k\}} \left\{ 0, r(k, k) + \sum_{i' \notin \{i, k\}} \max\{0, r(i', k)\} \right\} \quad (15)$$

A disponibilidade $a(i, k)$ recebe a auto responsabilidade $r(k, k)$ mais a soma das responsabilidades positivas de candidatos a exemplares que k recebe de outros itens. Para limitar a influência de uma responsabilidade positiva muito forte, a soma total é limitada de modo que não possa ser maior que 0. A auto responsabilidade $a(k, k)$ é atualizada usando a equação (16).

$$a(k, k) \leftarrow \sum_{i' \neq k} \max\{0, r(i', k)\} \quad (16)$$

Essa mensagem reflete a evidência acumulada de que o item k é um exemplar baseado na responsabilidade positiva enviada ao candidato a exemplar k de outros itens.

A regra acima requer apenas um simples cálculo e as mensagens só precisam ser trocadas entre pares de itens com similaridades conhecidas. A qualquer momento durante a propagação de afinidade, as disponibilidades e responsabilidades podem ser combinadas para identificar exemplares. Para o item i , o valor de k que maximiza $a(i, k) + r(i, k)$ deve identificar o item i como um exemplar se $k = i$ ou então identificar o item que é exemplar do item i .

O procedimento de troca de mensagens pode ser terminado após um determinado número de iterações, após as mudanças nas mensagens caírem abaixo de um limiar ou depois que decisões locais permaneçam constantes por repetidas iterações.

Para os testes o algoritmo foi aplicado a 900 imagens de faces em tons de cinza do banco de dados Olivetti. Para fins de comparação também é usado o algoritmo *k-centers*. O algoritmo de propagação de afinidade encontrou exemplares com valor de erro quadrático muito menor que as 100 melhores tentativas do *k-centers*.

Para investigar o algoritmo em uma situação em que as similaridades entre os itens não são conhecidas ou são grandes e negativas foi utilizada uma base de dados contendo 76.066 segmentos de DNA. O *k-centers* demorou mais de 1.000 vezes que o algoritmo proposto. Usando uma base em que se conhecem os rótulos para encontrar genes, o algoritmo conseguiu 3% de falsos positivos e 39% de verdadeiros positivos, ao passo que o melhor resultado do *k-centers* foi de 17%.

Uma das vantagens do algoritmo é que ele pode ser aplicado a problemas onde os dados não estão em um espaço contínuo. Também pode ser aplicado a problemas onde as similaridades não são simétricas ($s(i, k) \neq s(k, i)$) e a problemas onde as similaridades não satisfazem a desigualdade triangular ($s(i, k) < s(i, j) + s(j, k)$). Essas propriedades são compartilhadas pela caminhada do turista que usa uma matriz com um ranking de vizinhos. A caminhada é direcional e essa matriz não precisa ser simétrica.

3.3. Aprendizado Supervisionado Baseado na Caminhada do Turista

Outro grupo de artigos a ser discutido aqui refere-se aos que usam a caminhada do turista como técnica para a caracterização de dados. Primeiramente apresentamos uma proposta de aprendizado supervisionado que usa a caminhada do turista.

A ideia do uso da caminhada do turista em um classificador supervisionado foi utilizada em (SILVA; ZHAO, 2015). O trabalho propõe um *framework* de classificação híbrida que combina as decisões de classificadores de baixo nível e de alto nível. O classificador de baixo nível realiza a classificação considerando aspectos físicos dos dados tais como características geométricas ou estatísticas (a distância entre as instâncias, a distribuição dos dados e a composição de sua vizinhança). Em contraste, o classificador de alto nível pode extrair relações semânticas que extrapolam as relações ponto a ponto dos dados calculando a conformidade de uma nova instância de teste com os padrões formados por cada uma das classes que compõem o conjunto de treinamento. Instâncias de teste são declaradas membros de uma determinada classe quando as formações de padrões da classe são mantidas mesmo com a introdução dessas instâncias de teste. Para essa finalidade o classificador de alto nível extrai descritores organizacionais e topológicos (SILVA; ZHAO, 2012). A caminhada do turista pode ser usada para extrair tais descritores de alto nível.

A formalização do método é melhor descrita em (SILVA; ZHAO, 2016). Dado um conjunto de treinamento $X_{treinamento} = \{(x_1, y_1), \dots, (x_L, y_L)\}$ com L itens rotulados, onde o primeiro componente da i -ésima tupla $x_i = (x_{i1}, \dots, x_{ip})$ denota os

P atributos da i -ésima instância de treinamento e o segundo componente $y_i \in Y$ é o rótulo do item x_i .

O processo de classificação consiste em duas fases: a fase de treinamento e a fase de classificação. Na fase de treinamento o classificador é induzido ou treinado usando instâncias de treinamento (itens rotulados) no $X_{treinamento}$. No aprendizado supervisionado baseado em rede o classificador é representado por uma rede formada a partir dos dados de entrada e seus rótulos. A rede resultante é a rede de treinamento. Nessa fase os dados de treinamento são transformados em uma rede G usando a técnica de formação de rede k-NN com $\epsilon - radius$.

Na técnica do k-NN, dado um vértice de referência, simplesmente ordenamos todos os outros vértices de acordo com uma função de similaridade (distância). Dado um parâmetro k , selecionamos então os k primeiros vértices da lista ordenada e criamos arestas entre o vértice de referência e os vértices selecionados. Já com a técnica ϵ -radius as arestas são criadas quando os vértices em que a similaridade (distância) em relação ao vértice de referência é menor que um limiar ϵ .

Enquanto o ϵ -radius cria uma aresta entre dois vizinhos se eles estão dentro de uma distância ϵ , o k-NN cria uma aresta entre vértices i e j se i for um dos k vizinhos mais próximos de j . As duas abordagens possuem limitações no que diz respeito à esparsidade e densidade. Em regiões esparsas, o k-NN força um vértice a se conectar aos seus k vizinhos mais próximos mesmo que estejam longe. Nesse cenário podemos dizer que a vizinhança desse vértice pode conter pontos de dissimilaridade. Do mesmo modo, valores de ϵ inadequados podem facilmente resultar em um componente isolado. Com o objetivo de combinar as vantagens de ambas as abordagens é utilizada então uma combinação das duas. A técnica de construção de rede apresentada assume que não temos qualquer informação sobre a classe dos dados, porém, nas técnicas de Aprendizado de Máquina supervisionadas temos essas informações e elas podem ser usadas no estágio de construção de rede. Cada instância possui um rótulo correspondente indicando a classe a que pertence. Um refinamento na técnica de construção de rede seria justamente o uso dessa informação, de modo que permita criar arestas entre pares de vértices que pertencem à mesma classe. Essa restrição pode levar ao surgimento de mais de um componente de rede, cada qual com vértices da mesma classe. Dependendo da aplicação, poderia ser desejável haver conexões entre membros de diferentes classes onde os componentes poderiam ter uma

mistura dos mesmos. A maneira na qual essas restrições são empregadas será o que vai definir as propriedades topológicas da rede. Mesmo que a rede resultante tenha vários componentes podemos, por exemplo, criar regras para que uma nova instância de teste possa criar arestas com qualquer componente. Assim, cada vértice V representa uma instância de treinamento de $X_{\text{treinamento}}$ de modo que $V = L$. Para a construção da rede de treinamento temos a regra definida na equação (17).

$$N_{\text{treinamento}}(x_i) = \begin{cases} \epsilon - \text{radius}(x_i, y_i), & \text{se } |\epsilon - \text{radius}(x_i, y_i)| > k \\ k - NN(x_i, y_i), & \text{caso contrário} \end{cases} \quad (17)$$

Onde y_i representa o rótulo da instância de treinamento x_i , $\epsilon - \text{radius}(x_i, y_i)$ retorna o conjunto $\{x_j, j \in V : d(x_i, x_j) \leq \epsilon \wedge y_i = y_j\}$ e $k - NN(x_i, y_i)$ retorna o conjunto contendo os k vizinhos mais próximos e de mesma classe do vértice x_i . Assim, a técnica $\epsilon - \text{radius}$ é usada em regiões densas ($|\epsilon - \text{radius}(x_i, y_i)| > k$) enquanto que o k-NN é usado em regiões esparsas. Com essa estratégia é esperado que cada classe seja representada por um único componente.

Já na fase de classificação as instâncias de teste (sem rótulos) no X_{teste} são apresentadas ao classificador um a um. A vizinhança da instância de teste x_i é definida seguindo a equação (18).

$$N_{\text{classificação}}(x_i) = \begin{cases} \epsilon - \text{radius}(x_i), & \text{se } |\epsilon - \text{radius}(x_i)| > k \\ k - NN(x_i), & \text{caso contrário} \end{cases} \quad (18)$$

Esta equação mostra que a instância de teste pode se conectar a qualquer vértice dentro do raio ϵ independentemente da classe. Se a região for suficientemente esparsa ($|\epsilon - \text{radius}(x_i)| \leq k$) então o k-NN é empregado. A nova rede (com a instância de teste) é a rede de classificação. Uma vez que a instância de teste foi inserida, cada componente (composto por uma mesma classe) calcula as mudanças que ocorreram na formação de padrão com a inserção da instância de teste. O cálculo é feito para extrair métricas dinâmicas da rede. Tais métricas são compostas por duas quantidades diretamente derivadas da dinâmica da caminhada do turista, sendo elas o comprimento do transiente e do ciclo final (atrator). Se ocorrer nenhuma ou pouca mudança, então é dito que essa instância de teste está em conformidade com o padrão da classe. Como resultado, o classificador de alto nível retorna um alto valor

indicando que essa instância deve pertencer a essa classe. Por outro lado, se a inserção da instância de teste causa uma grande mudança no padrão da classe, então o classificador de alto nível retorna um baixo valor. É importante ressaltar que caso a instância de teste não compartilhe qualquer aresta com um determinado componente de uma classe, a instância de teste não está em conformidade com essa classe. Já nos componentes em que pelo menos uma aresta é compartilhada com a instância de teste, cada componente calcula o impacto (métricas da caminhada do turista) da inserção da instância de teste. Ao mesmo tempo o classificador de baixo nível também é construído. Esse classificador pode ser implementado por qualquer técnica tradicional da literatura (redes neurais, k-NN, *Support Vector Machine* etc.). Ao final, as previsões produzidas por ambos são combinadas para que tenhamos um resultado final. Cada técnica fornece uma visão diferente dos dados. Enquanto o classificador de baixo nível tem boa performance com dados bem distribuídos e separados, o classificador de alto nível tem a habilidade de identificar o significado semântico nos dados.

O classificador híbrido retorna uma decisão $F_i^{(y)}$ seguindo a equação (19). A decisão representa a qual classe $y \in Y$ a instância de teste $x_i \in X_{teste}$ pertence.

$$F_i^{(y)} = (1 - \rho)L_i^{(y)} + \rho H_i^{(y)} \quad (19)$$

Onde $L_i^{(y)} \in [0,1]$ e $H_i^{(y)} \in [0,1]$ são as previsões de baixo nível e alto nível da instância de teste x_i em relação à classe y e $\rho \in [0,1]$ é o termo de conformidade. Quando $L_i^{(y)} = 1$ e $H_i^{(y)} = 1$ então a instância x_i tem todas as características da classe y , tanto no baixo nível, ou seja, há similaridade, quanto no alto nível, ou seja, há conformidade com o padrão do componente da classe. Já quando temos $L_i^{(y)} = 0$ e $H_i^{(y)} = 0$, x_i não apresenta similaridades nem está em conformidade com a classe y . Valores entre esses dois extremos são os casos mais comuns. O classificador pode ser considerado um método de classificação *fuzzy* pois como temos que tanto $L_i^{(y)}$ quanto $H_i^{(y)}$ variam de 0 a 1, então temos que $F_i^{(y)} \in [0,1]$.

A instância de teste x_i recebe o rótulo da classe $y \in Y$ que maximiza $F_i^{(y)}$, ou seja, dado um $F_i^{(y)}$ para todos os $y \in Y$, a classe será aquela em que $F_i^{(y)}$ retornou o maior valor.

A saída do classificador de alto nível baseado na caminhada do turista é dada pela equação (20).

$$H_i^{(y)} = K_H \sum_{\mu=0}^{\mu_c^{(y)}} w_{inter}^{(y)}(\mu) [w_{intra}^{(y)}(\mu) T_i^{(y)}(\mu) + (1 - w_{intra}^{(y)}(\mu)) C_i^{(y)}(\mu)] \quad (20)$$

Onde $\mu_c^{(y)}$ é um valor crítico que indica o tamanho máximo da memória do turista que foi executada na fase de treinamento para a classe y . $T_i^{(y)}(\mu)$ e $C_i^{(y)}(\mu)$ são funções que dependem do tamanho do transiente do ciclo respectivamente da caminhada do turista aplicada ao i -ésimo item em relação à classe y . Essas funções são responsáveis por estimar quando o item i possui o mesmo padrão do componente da classe y . $w_{inter}^{(y)}(\mu)$ é o peso que é dado à caminhada do turista de memória μ na classe y . O termo inter indica que se refere à regulação da caminhada do turista com diferentes μ . $w_{intra}^{(y)}(\mu)$ é o peso do tamanho do transiente de uma caminhada do turista, em particular com memória μ na classe y . O termo complementar $(1 - w_{intra}^{(y)}(\mu))$ se refere à mesma informação, porém em relação ao comprimento do ciclo. O termo intra indica que o coeficiente modula a dinâmica de uma mesma caminhada do turista. Por último, K_H é uma constante de normalização que garante o "fuzziness" do classificador de alto nível.

Os descritores que caracterizam a variação estrutural no componente de uma determinada classe y dada a inserção de uma instância de teste x_i são definidas nas equações (21) e (22).

$$T_i^{(y)}(\mu) = 1 - \Delta t_i^{(y)}(\mu) p^{(y)} \quad (21)$$

$$C_i^{(y)}(\mu) = 1 - \Delta c_i^{(y)}(\mu) p^{(y)} \quad (22)$$

Onde $\Delta t_i^{(y)}(\mu)$ e $\Delta c_i^{(y)}(\mu) \in [0,1]$ são as variações do comprimento do transiente e do ciclo no componente da classe y se a instância de teste i se ligar ao componente, e $p^{(y)} \in [0,1]$ é a proporção de instâncias de treinamento pertencentes à

classe y . Para computar $\Delta t_i^{(y)}(\mu)$ e $\Delta c_i^{(y)}(\mu)$ para um dado tamanho de memória μ , é realizada uma caminhada do turista para cada vértice do componente da classe $y \in Y$. Desse modo podemos primeiramente calcular as médias dos comprimentos de transiente e de ciclo $t_i^{(y)}(\mu)$ e $c_i^{(y)}(\mu)$ respectivamente. Depois que a instância de teste x_i é inserida no componente da classe $y \in Y$ são recalculadas as médias dos comprimentos de transiente e de ciclo nesse componente denotadas por $t_i'^{(y)}(\mu)$ e $c_i'^{(y)}(\mu)$ respectivamente. Esse procedimento é feito para todas as classes $y \in Y$. Assim, calculamos $\Delta t_i^{(y)}(\mu)$ e $\Delta c_i^{(y)}(\mu)$, $\forall y \in Y$ seguindo as equações (23) e (24).

$$\Delta t_i^{(y)}(\mu) = \frac{|[t_i'^{(y)}(\mu)] - [t_i^{(y)}(\mu)]|}{\sum_{u \in Y} |[t_i'^{(u)}(\mu)] - [t_i^{(u)}(\mu)]|} \quad (23)$$

$$\Delta c_i^{(y)}(\mu) = \frac{|[c_i'^{(y)}(\mu)] - [c_i^{(y)}(\mu)]|}{\sum_{u \in Y} |[c_i'^{(u)}(\mu)] - [c_i^{(u)}(\mu)]|} \quad (24)$$

Uma grande variação nos comprimentos de transiente e de ciclo em um componente, $\Delta t_i^{(y)}(\mu)$ e $\Delta c_i^{(y)}(\mu)$ produz um baixo valor de membresia $T_i^{(y)}(\mu)$ e $C_i^{(y)}(\mu)$, enquanto que pequenas variações produzem altos valores de membresia. Os passos descritos anteriormente são feitos com a memória μ variando de 0 até um valor crítico μ_c .

O classificador híbrido foi testado em várias bases de dados da UCI *Machine Learning Repository* (FRANK; ASUNCION, 2010). Primeiramente o classificador de baixo nível foi treinado de forma independente usando *cross-validation* com 10 *folds*. Para compor o classificador de alto nível é preciso definir os parâmetros k e ϵ para a construção da rede. Também é preciso definir o termo de conformidade ρ . Para estimar os parâmetros (k, ϵ, ρ) é usado somente o conjunto de treinamento. Unicamente o conjunto de parâmetros (k, ϵ, ρ) que retorne a melhor acurácia é escolhido para ser testado com o conjunto de teste.

Foram usados dois tipos de classificadores de alto nível. No tipo "Sem rede" a caminhada do turista é executada sobre os dados sem a fase da construção da rede, ou seja, o turista pode visitar qualquer outro item (desde que não esteja em sua memória

μ). No tipo "Com rede" há a construção da rede e o turista só pode visitar vértices de sua vizinhança (vértices que compartilham uma aresta). Finalmente, no tipo "puro", o termo de conformidade $\rho = 0$, ou seja, apenas o classificador de baixo nível é utilizado e este serviu para comparação com o método proposto. Diferentes classificadores de baixo nível foram utilizados: rede bayesiana, *perceptron* multicamadas (MLP) e *support vector machine multi classes* (Fuzzy M-SVM).

Os resultados mostram que o classificador híbrido tem uma performance sempre igual ou melhor. Isso se deve ao fato de que quando o classificador de alto nível prejudica a classificação podemos simplesmente fazer $\rho = 0$ e usar apenas o classificador de baixo nível. Assim sendo, foi feito um teste de significância para verificar quando a melhoria trazida pelo classificador de alto nível era de fato relevante. Em vários casos, o classificador híbrido foi estatisticamente melhor do que quando utilizado apenas o classificador de baixo nível.

Como podemos ver, a caminhada do turista pode ser usada para capturar propriedades topológicas de uma rede construída a partir dos dados. O termo de alto nível deve ser aumentado para obter uma classificação correta quando a complexidade da configuração da classe aumenta. Isso significa que o termo de alto nível é útil em casos complexos de classificação.

3.4. Aprendizado Não Supervisionado Baseado na Caminhada do Turista

O outro trabalho apresentado aqui é uma técnica de aprendizado não supervisionado hierárquico que também usa a caminhada do turista. O uso da caminhada do turista como um algoritmo de reconhecimento de padrões foi proposto em (CAMPITELI et al., 2006). O artigo propõe o uso desse algoritmo como uma estratégia para identificar padrões em conjuntos de dados de maneira hierárquica, sendo que a memória μ representa o nível hierárquico onde pontos pertencentes a um mesmo atrator são agrupados. Quando a memória do turista é $\mu = 0$ (o turista não possui memória) cada ponto se torna um atrator, ou seja, o turista não sai do ponto inicial e assim, no nível hierárquico $\mu = 0$ cada item do conjunto de dados é um grupo, tal qual acontece em métodos aglomerativos tradicionais. Quando $\mu = 1$ (o turista lembra da cidade atual), o turista vai para a cidade mais próxima até que duas cidades mutualmente mais próximas formem um atrator. Para valores maiores de μ , o

turista executa caminhadas mais longas, eventualmente formando atratores maiores. Os atratores são formados independentemente do nível anterior e quando um novo atrator contém parte ou o total de um grupo formado em algum nível anterior, eles então são unidos em um único grupo. Quando $\mu = N - 1$, sendo N a quantidade de itens no conjunto de dados, a trajetória é completamente auto repulsiva e todos os itens do conjunto de dados formam um único atrator. Os movimentos do turista são executados usando uma tabela de vizinhança que, uma vez construída, não necessita da distância entre os pontos.

O resultado do processo é uma árvore similar às árvores dos métodos hierárquicos tradicionais, porém ela não obedece uma estrutura n -ária, ou seja, cada nível hierárquico pode agrupar um número arbitrário de itens.

Foram criados dois conjuntos de dados artificiais para destacar as principais características da técnica. Também foi usado um conjunto com dados reais. Os resultados foram comparados com o método *single-linkage*. O primeiro conjunto de dados é composto por 33 itens em um espaço bidimensional organizado em 4 grupos bem separados e 1 elemento colocado equidistante dos 4 grupos. O segundo conjunto de dados é composto por 100 elementos em um espaço bidimensional divididos em dois grupos. Os dois grupos são equivalentes porém plotados em escalas diferentes. Um grupo é um aumento em 10 vezes do outro. O terceiro conjunto de dados é o iris (FISHER, 1936), contendo 150 amostras de flores (iris) com medidas de largura e comprimento da pétala e sépala. As flores possuem 3 categorias. Foram aplicados em todos os conjuntos o *single-linkage* e o método proposto com a caminhada do turista.

O algoritmo proposto encontrou os *clusters* conforme o esperado, e as principais vantagens apresentadas pelo algoritmo foram: (1) não há necessidade de trabalhar com uma matriz de distâncias, é usada apenas uma tabela de vizinhança; (2) a caminhada do turista favorece informações locais e não faz restrições sobre a estrutura da árvore; e (3) ao não considerar as distâncias entre os elementos mas apenas as vizinhanças, ele se torna invariante de escala.

4. Estudo da Utilização da Caminhada do Turista como Método de *Clustering* Hierárquico

Neste capítulo é apresentado um estudo da implementação da estratégia não supervisionada hierárquica baseada na caminhada do turista aplicada a uma série de conjuntos de imagens médicas.

4.1. Materiais e métodos

4.1.1. Bases de Imagens Médicas

A primeira base de imagens médicas consiste em um conjunto de imagens microscópicas obtidas a partir de biópsias de pólipos colorretais laudadas por patologistas e separadas em 4 classes, de acordo com o grau de desenvolvimento da lesão. São 30 imagens categorizadas como sem atipia (Figura 7 (a)), 30 imagens categorizadas como baixo grau de atipia (Figura 7 (b)), 30 imagens categorizadas como alto grau de atipia (Figura 7 (c)) e 30 imagens categorizadas como câncer (Figura 7 (d)). As imagens possuem resolução de 2584 x 193 pixels. Estas imagens foram adquiridas a partir de um microscópio de luz Axiophot (Carl Zeiss Jena, Alemanha) com lentes objetivas de imersão em óleo, sendo capturadas digitalmente por uma câmera digital (TK-1270, JVC, Companhia Victor do Japão, Tokyo, Japão) (CROCETTI, 2014).

Figura 7 - Imagens dos diferentes níveis de atipia em biópsias de pólipos colorretais

(a) Sem atipia

(b) Baixo grau de atipia

(c) Alto grau de atipia

(d) Câncer

Fonte: Crocetti (2014)

A segunda base de imagens é composta por imagens de mamografia contendo regiões de interesse (ROI) de nódulos de mama. As imagens estão divididas em 100 imagens classificadas como de nódulos benignos (Figura 8 (a)) e 146 imagens classificadas como de nódulos malignos (Figura 8 (b)). Estas ROIs foram geradas previamente a partir da base de mamogramas *Digital Database for Screening Mammography* (DDSM) disponibilizada pela Universidade do Sul da Flórida (HEATH et al., 2001).

Figura 8 - Imagens de áreas de interesse de nódulos de mama benigno (a) e maligno (b)

(a) Nódulo benigno

(b) Nódulo maligno

Fonte: Heath et al. (2001)

A terceira base de imagens consiste em ressonâncias magnéticas de diferentes partes do corpo. São 30 imagens de angiograma Figura 9 (a), 34 imagens coronais de abdômen Figura 9 (b), 50 imagens axiais da cabeça Figura 9 (c) e 38 imagens sagitais Figura 9 (d) também da cabeça. Todas as imagens têm dimensões 256 x 256 pixels e estão em tons de cinza. (BALAN, 2007).

Figura 9 - Imagens de ressonâncias magnéticas de diversas partes do corpo humano

(a) Angiograma

(b) Coronal de abdômen

(c) Axial da cabeça

(d) Sagital da cabeça

Fonte: Balan (2007)

A quarta base é integrada por imagens histopatológicas de tecido epitelial do colo do útero (MIRANDA et al., 2012). As imagens se originam de lâminas de biópsias de colo de útero. O sistema de captura consiste em um microscópio Óptico Zeiss Axiophot com uma câmera CCD AxioCom-MRC. As imagens foram aumentadas em 320 vezes e a resolução é de 1388 x 1040 pixels. As imagens foram classificadas como normais e em 3 graduações distintas de lesões. São 39 itens classificados como normais, 30 com grau 1, 25 grau 2 e 22 com grau 3 (Figura 10).

Figura 10 - Lesão do tecido do colo uterino

Fonte: Miranda et al. (2012)

As imagens de pólipos colorretais, nódulos de mama e ressonâncias magnéticas foram representadas pelo descritor de textura de Haralick (HARALICK; SHANMUGAM; DINSTEIN, 1973). Diferentes combinações de medidas estatísticas propostas pelo descritor de Haralick foram utilizadas. As imagens de tecido epitelial do colo do útero são representadas por um vetor de características baseado em um método de análise estrutural que usa técnicas de redes complexas e foi desenvolvido em (MIRANDA et al., 2012).

4.2. Ambiente de Desenvolvimento

A implementação das técnicas aqui apresentadas foi desenvolvida na linguagem de programação R. Alguns pacotes disponibilizados no repositório CRAN¹ foram utilizados. A técnica de aprendizado não supervisionado hierárquico baseado na caminhada do turista aqui desenvolvido gera como resultado árvores no formato *newick*. Para ler e transformar essas árvores (dendrogramas) foi utilizado o pacote *ape* (PARADIS; CLAUDE; STRIMMER, 2004). A manipulação desses dendrogramas e a extração de algumas métricas foram feitas com o auxílio do pacote *dendextend* (GALILI, 2015). A métrica da silhueta foi extraída utilizando o pacote *cluster* (MAECHLER et al., 2016). A visualização das medidas de silhueta foi feita através do pacote *factoextra* (KASSAMBARA; MUNDT, 2017). O algoritmo de *clustering* DBSCAN foi utilizado através do pacote *dbscan* (HAHSLER; PIEKENBROCK, 2017) e o algoritmo de *clustering Affinity Propagation* foi utilizado através do pacote *apcluster* (BODENHOFER; KOTHMEIER; HOCHREITER, 2011).

¹ cran.r-project.org

4.3. Resultados

Aqui apresentamos os resultados obtidos com a implementação da técnica de *clustering* hierárquico aglomerativo baseada na caminhada do turista, assim como a comparação desses resultados com alguns algoritmos hierárquicos tradicionais. As comparações serão feitas utilizando as bases de imagens médicas apresentadas anteriormente.

Usualmente, em uma aplicação não supervisionada não temos informações sobre as classes das instâncias. Aqui, no entanto, usaremos dados rotulados para observar como os dendrogramas os separam. Para a melhor visualização dos dendrogramas resultantes dos algoritmos usaremos amostras reduzidas das bases de imagens. Isso nos permitirá visualizar facilmente como as classes esperadas foram agrupadas.

4.3.1. Biópsias de Pólipos Colorretais

Primeiramente vamos analisar o conjunto de imagens de biópsias de pólipos colorretais. Para esse experimento foi utilizada uma amostra contendo 20 imagens categorizadas como normais (sem atipias) e 20 imagens categorizadas como com câncer. Para termos uma ideia da distribuição dessas imagens temos na Figura 11 os dois primeiros componentes do PCA (*Principal Component Analysis*) dos vetores com as características obtidas com os descritores de Haralick de homogeneidade, variância (contrastes) e entropia explicados na seção 2.1.2.

Figura 11 - Primeiros dois componentes do PCA dos vetores de características das imagens de pólipos colorretais

Fonte: Elaborada pelo autor.

Apresentamos agora os dendrogramas obtidos usando os métodos *single link* (Figura 12 (a)), *complete link* (Figura 12 (b)), *average* (UPGMA) (Figura 12 (c)), McQuitty (WPGMA) (Figura 12 (d)), *centroid* (UPGMC) (Figura 12 (e)), *median* (WPGMC) (Figura 12 (f)), Ward (Figura 12 (g)) e a técnica baseada na caminhada do turista (Figura 12 (h)). As técnicas foram explicadas na seção 2.1.4.2. A distância utilizada foi a distância euclidiana.

Figura 12 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de pólipos colorretais

(a) Single link

(b) Complete link

Fonte: Elaborada pelo autor.

As cores representam as classes reais, de modo que podemos ter ideia da acurácia dos algoritmos. É importante ressaltar, no entanto, que em um Aprendizado de Máquina não supervisionado normalmente não temos informações acerca da classe das instâncias, e o objetivo nesse caso é encontrar uma organização guiada meramente

pelos dados. As classes aqui servem como uma validação externa do dendrograma gerado.

A técnica baseada na caminhada do turista apresenta algumas diferenças fundamentais com as técnicas tradicionais. Em primeiro lugar, ela é feita usando uma matriz de vizinhos e não as distâncias dos dados. Isso permite uma invariabilidade em relação a mudanças de escala. Outra diferença é que os níveis hierárquicos representam os tamanhos de memória nos quais aqueles atratores contendo os itens agrupados foram encontrados, ou seja, os níveis não representam diretamente os níveis de dissimilaridade dos itens e o resultado é uma árvore que pode unir um número arbitrário de itens em cada nível hierárquico. Isso permite que os itens sejam unidos em um número menor de níveis hierárquicos, permitindo uma árvore mais organizada. Nesse caso é especialmente interessante investigar como as distâncias cofenéticas de tais árvores se correlacionam com as distâncias reais dos dados.

Um passo que usualmente é seguido após a obtenção do dendrograma é o corte em uma determinada altura para se obter as partições desejadas. Os algoritmos hierárquicos tradicionais unem os *clusters* 2 a 2, gerando sempre uma árvore que se ramifica 2 a 2. Isso permite que a árvore seja cortada e um número determinado de *clusters* seja obtido. A estratégia baseada na caminhada do turista, por outro lado, adota uma estratégia em que se produz uma árvore não n-ária, onde em cada nível um número arbitrário de ramos pode ser unido. Nas Figura 12 podemos ver que na estratégia *single link* (a) e na *median* (WPGMC) (f) o corte da árvore em duas ramificações não produziria um resultado satisfatório, ao passo que no dendrograma gerado pela técnica baseada na caminhada do turista é possível dividir em duas ramificação com resultado satisfatório.

É interessante investigar o impacto de uma árvore não n-ária comparada com a configuração binária imposta por métodos tradicionais. Na Tabela 5 podemos ver os valores da correlação entre a matriz de distância dos dados (distância euclidiana) e a distância cofenética de cada dendrograma (distância entre os itens dentro do dendrograma).

Tabela 5 - Correlações das distâncias reais dos dados com as distâncias cofenéticas das diferentes técnicas de clustering hierárquico para as imagens de pólipos colorretais

Técnica	Correlação
Single link	0,587
Complete link	0,773
Average (UPGMA)	0,777
McQuitty (WPGMA)	0,774
Centroid (UPGMC)	0,770
Median (WPGMC)	0,619
Ward	0,761
Caminhada do Turista	0,711
Média	0,721

Fonte: Elaborada pelo autor.

Como podemos ver, o dendrograma gerado pela técnica baseada na caminhada do turista preserva correlação com as distâncias reais e se aproxima do valor médio da correlação das demais técnicas.

Na Figura 13 temos uma comparação direta entre os dendrogramas. A imagem mostra a correlação das distâncias cofenéticas entre as técnicas apresentadas.

Figura 13 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de pólipos colorretais

Pode ser observado que a técnica baseada na caminhada do turista é consistentemente similar às demais técnicas, com exceção da técnica *single link* e *median*, as quais não obtiveram partições satisfatórias (considerando uma divisão em

2) e se diferenciam das demais. É interessante notar como o dendrograma gerado pela técnica baseada na caminhada do turista, embora baseada no atratores, unindo vários itens em um mesmo nível hierárquico, ainda preserva correlação com a distância real dos itens.

4.3.2. Imagens de Ressonância Magnética

O próximo conjunto de imagens a ser analisado é o grupo de imagens de ressonâncias magnéticas de diferentes partes do corpo humano. Aqui usamos uma amostra com 10 imagens de angiograma, 20 imagens de cabeça sendo 10 axiais e 10 sagitais e 10 imagens coronais de abdômen. Na Figura 14 podemos ver como os dados estão distribuídos através dos dois primeiros componentes do PCA dos vetores obtidos com os descritores de Haralick de variância.

Figura 14 - Primeiros dois componentes do PCA dos vetores de características das imagens de ressonâncias magnéticas

Fonte: Elaborada pelo autor.

Apresentamos a seguir os dendrogramas obtidos com os métodos tradicionais.

Figura 15 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de ressonâncias magnéticas

Fonte: Elaborada pelo autor.

Os resultados obtidos com a técnica baseada na caminhada do turista são satisfatórios pois é possível ver que os 4 grupos, com alguns erros e *outliers*, podem

ser cortados em uma mesma altura. O número menor de níveis hierárquicos neste caso facilita a visualização da árvore.

Na Tabela 6 temos a correlação da distância real entre os dados e a distância cofenética de cada dendrograma.

Tabela 6 - Correlações das distâncias reais dos dados com as distâncias cofenéticas das diferentes técnicas de clustering hierárquico para as imagens de ressonância magnética

Técnica	Correlação
Single link	0,669
Complete link	0,739
Average (UPGMA)	0,822
McQuitty (WPGMA)	0,629
Centroid (UPGMC)	0,792
Median (WPGMC)	0,816
Ward	0,707
Caminhada do Turista	0,624
Média	0,728

Fonte: Elaborada pelo autor.

Como podemos ver, foi preservada uma correlação entre as distâncias reais dos dados e a distância dos mesmos no dendrograma, embora esse valor tenha sido mais baixo para a técnica baseada na caminhada do turista. Podemos ver mais uma vez que a técnica baseada na caminhada do turista apresentou uma árvore mais organizada, com um número menor de níveis hierárquicos e que ainda assim preserva correlação com a distância real dos dados.

Na Figura 16 temos as correlações da distância cofenéticas entre os dendrogramas das diferentes técnicas.

Figura 16 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de ressonâncias magnéticas

Fonte: Elaborada pelo autor.

A técnica baseada na caminhada do turista também foi capaz de separar satisfatoriamente os itens desse grupo de imagens, com uma árvore com um número reduzido de níveis hierárquicos e preservando uma correlação com as distâncias reais dos dados.

4.3.3. Imagens Histopatológicas de Tecido Epitelial do Colo Uterino

O conjunto apresentado em seguida é o de imagens histopatológicas de tecido epitelial do colo do útero. O objetivo primário aqui é examinar como o algoritmo hierárquico baseado na caminhada do turista se compara aos algoritmos de *clustering* hierárquicos tradicionais, e não fazer uma análise de imagens propriamente dita. Assim, para esse experimento usamos um *sample* de imagens que apresentam características mais marcantes a suas respectivas classes, contendo 14 imagens de tecido normal (representado pelo valor 1), 14 imagens com nível de lesão intermediária (representado pelo valor 2) e 14 imagens com nível de lesão alto (representado pelo valor 3). Para termos uma ideia da distribuição dos dados, na Figura 17 são exibidos os dois primeiros componentes do PCA dos vetores de características com as métricas baseadas em redes complexas utilizadas em (MIRANDA et al, 2012). Essas métricas foram explicadas na seção 2.1.2.

Figura 17 - Primeiros dois componentes do PCA dos vetores de características das imagens de tecido epitelial do colo do útero

Fonte: Elaborada pelo autor.

Na Figura 18 temos os dendrogramas obtidos com os métodos hierárquico tradicionais.

Figura 18 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de tecido epitelial do colo do útero

Fonte: Elaborada pelo autor.

Os dendrogramas que apresentam um resultado em que as classes foram corretamente separadas e que permitem um corte na altura apropriada foram os obtidos com o método Ward e com a técnica baseada na caminhada do turista. O dendrograma obtido com a técnica baseada na caminhada do turista apresenta poucos níveis hierárquicos, tornando a visualização da árvore mais fácil, sendo que níveis foram suficientes para agrupar os dados de modo apropriado.

Na Tabela 7 temos a correlação de distância real dos itens do conjunto de dados e suas distâncias dentro do dendrograma (distância cofenética).

Tabela 7 - Correlações das distâncias reais dos dados com as distâncias cofenéticas das diferentes técnicas de clustering hierárquico para as imagens de tecido epitelial do colo do útero

Técnica	Correlação
Single link	0,635
Complete link	0,753
Average (UPGMA)	0,757
McQuitty (WPGMA)	0,747
Centroid (UPGMC)	0,727
Median (WPGMC)	0,686
Ward	0,725
Caminhada do Turista	0,713
Média	0,718

Fonte: Elaborada pelo autor.

Pode ser observado que a caminhada mostrou uma correlação próxima à média. Abaixo na Figura 19 temos a correlação das distâncias cofenéticas entre as técnicas apresentadas.

Figura 19 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de tecido epitelial do colo do útero

Fonte: Elaborada pelo autor.

Podemos ver que mesmo com um número menor de níveis hierárquicos a técnica baseada na caminhada do turista se aproxima com o que é apresentado pelas demais técnicas, porém com uma árvore mais organizada e que apresenta os grupos da forma desejada, permitindo ainda um corte na árvore que representa de forma apropriada a partição desejada.

4.3.4. Imagens de Regiões de Interesse em Nódulos de Mama

O próximo conjunto de imagens será o grupo de regiões de interesse de nódulos de mama. Usamos um *sample* contendo 25 imagens classificadas como de nódulo benigno e 25 imagens classificadas como de nódulo maligno. Na Figura 20 podemos ver a distribuição dos dados através dos dois primeiros componentes do PCA dos vetores obtidos com o descritor de Haralick de variância, descrito em 2.1.2.

Figura 20 - Primeiros dois componentes do PCA dos vetores de características das imagens de regiões de interesse em nódulos de mama

Fonte: Elaborada pelo autor.

Os dendrogramas gerados pelos métodos tradicionais podem ser vistos a partir da Figura 21.

Figura 21 - Dendrogramas resultantes das técnicas hierárquicas para as imagens de regiões de interesse de nódulos de mama

Fonte: Elaborada pelo autor.

Na Tabela 8 podemos ver a correlação entre a distância real e a distância dentro do dendrograma.

Tabela 8 - Correlações das distâncias reais dos dados com as distâncias cofenéticas das diferentes técnicas de clustering hierárquico para as imagens de regiões de interesse de nódulos de mama

Técnica	Correlação
Single link	0,905
Complete link	0,883
Average (UPGMA)	0,913
McQuitty (WPGMA)	0,902
Centroid (UPGMC)	0,907
Median (WPGMC)	0,920

Ward	0,580
Caminhada do Turista	0,780
Média	0,849

Fonte: Elaborada pelo autor.

Podemos ver no dendrograma gerado pela técnica baseada na caminhada do turista vários ramos unidos em um mesmo nível hierárquico, gerando uma árvore organizada com um número menor de hierarquias no qual é possível, considerando os *outliers*, perceber dois *clusters* definidos. O fato de a árvore não ser binária impede que a árvore seja cortada em um número arbitrário de *clusters* porém podemos observar ramos com apenas um item e considerando-os *outliers* podemos selecionar os *clusters* desejados.

A distância cofenética preserva correlação com a distância real, mas nesse caso fica claro que o valor abaixo da média dessa correlação mostra como a invariabilidade da escala pode produzir uma árvore n-ária em que não se pode facilmente cortar em um número arbitrário de *clusters*. Nesse caso se faz necessário uma inspeção visual e podemos considerar ramos fundidos em um alto nível hierárquico como *outliers*.

Na Figura 22 temos a correlação da distância cofenética entre os diferentes métodos.

Figura 22 - Correlação das distâncias cofenéticas entre as diferentes técnicas de clustering hierárquico para as imagens de regiões de interesse de nódulos de mama

Fonte: Elaborada pelo autor.

Aqui formalizamos, através de bases de imagens médicas, que a técnica de *clustering* hierárquico aglomerativo baseado na caminhada do turista é capaz de identificar *clusters* em conjuntos de imagens médicas através de uma árvore que não impõe uma estrutura binária, e que possui um número menor de hierarquias e uma invariabilidade à escala dos dados resultando em uma estrutura mais organizada e de mais fácil visualização. Mesmo que a árvore não seja diretamente baseada nas distâncias dos dados, mas sim em uma matriz de vizinhança, ela ainda preserva uma correlação com a distância real dos dados.

5. Proposta: Algoritmo de *Clustering* Particional Baseado na Caminhada do Turista

5.1. Introdução

Uma diferença a ser notada entre os métodos hierárquicos tradicionais e o hierárquico baseado na caminhada do turista é que nos métodos tradicionais cada nível hierárquico depende diretamente do passo anterior onde os *clusters* vão sendo unidos 2 a 2. Na técnica da caminhada do turista é comum que um atrator encontrado em um determinado nível englobe todos os itens daquele nível (englobando atratores encontrados em passos anteriores). Os casos em que um atrator compartilha apenas alguns itens com um atrator encontrado em um nível anterior acontecem, porém não são tão comuns quanto o primeiro caso. Em conjuntos de dados em que os *clusters* são bem separados, caminhadas do turista com um único tamanho de memória apropriado já é capaz de retornar atratores que representem bem estes *clusters*. Isso nos indica que a caminhada com apenas um tamanho de memória pode ser usada como estratégia de *clustering* particional. Um outro aspecto relevante é que neste caso a parte transiente da caminhada pode ser um indicador de ruído nos dados. Um algoritmo desse tipo, embora similar a um corte no dendrograma da técnica hierárquica, seria mais eficiente, exigindo caminhadas de apenas um tamanho de memória μ e incorporaria melhor o conceito de ruído e *outliers*.

Consideremos, por exemplo, o conjunto de dados apresentado na Figura 23, formado por duas distribuições normais bem separadas onde cada *cluster* possui 100 instâncias

Figura 23 - Duas distribuições normais formando dois clusters bem separados

Fonte: Elaborada pelo autor.

Para esse exemplo, caminhadas do turista de tamanho de memória $\mu = 80$ iniciadas em cada item encontram, através dos atratores das caminhadas, um resultado satisfatório dos *clusters* presentes nos dados.

Uma característica interessante é que ao usarmos o atrator como identificador do *cluster*, a parte transiente da caminhada pode ser utilizada para identificar o ruído. Um tipo de ruído que pode ser encontrado é o ruído de fundo (ou *background clutter*). Para exemplificar a questão, é usada a Figura 24 contendo 4 *clusters* bem definidos e um ruído de fundo. Esse conjunto de dados contém 4 classes: cada qual com 111, 114, 150 e 109 instâncias e como ruído de fundo 138 instâncias.

Figura 24 - Conjunto de Dados com Ruído de fundo (*background clutter*)

Fonte: Elaborada pelo autor com dados baseados em Zelnik-Manor e Perona (2005)

Caminhadas do turista de tamanho de memória $\mu = 100$ são capazes de identificar os *clusters* e o ruído. O ruído não é identificado como um *cluster*, trata-se de instâncias que não são associadas a nenhum *cluster*. Caminhadas iniciadas nesses itens terminam formando atratores nos *clusters* de fato.

Outro conjunto artificial de dados interessante que demonstra o comportamento de um algoritmo de *clustering* na presença de ruído é o provido pela *Fundamental Clustering Problem Suite* (ULTSCH, 2005). Esse conjunto possui 2 *clusters* bem definidos: um contendo 395 instâncias e outro 363 instâncias e como ruído/outliers temos 4 conjuntos com 3 instâncias cada, conforme mostrado na Figura 25.

Figura 25 - *Fundamental Clustering Problem Suite* - Conjunto de dados com presença de ruído

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

Caminhadas do turista de tamanho de memória $\mu = 270$ foram capazes de encontrar os *clusters* e os *outliers*.

Um problema desta abordagem é que ela exige valores de tamanho de memória μ altos, normalmente valores superiores a $(n/K) * (2/3)$, onde n é a quantidade de itens no conjunto de dados e K o número de *clusters*. Valores de memória altos impõem maiores restrições ao caminhante forçando-o a caminhadas mais longas e aumentando assim o custo computacional do algoritmo. O segundo problema é que o algoritmo é sensível ao tamanho da memória, ou seja, pequenas diferenças nos valores desse parâmetro produzem resultados diferentes e não temos como saber *a priori* qual o valor ideal deste parâmetro. Uma abordagem possível seria

testar diferentes valores e através de métricas de validação de *cluster* escolher o mais apropriado. O custo computacional de caminhadas longas e a pouca tolerância a diferentes tamanhos de memória (o que exigiria muitos testes até encontrar o parâmetro ideal) o tornam de difícil execução em problemas reais.

O algoritmo também não é capaz de separar determinados formatos arbitrários de *clusters*. Evitar a preferência de algoritmos por formatos globulares (ou outros formatos específicos) de *clusters* tem sido uma das motivações para propostas de novos algoritmos. Na Figura 26 vemos uma configuração de formas arbitrárias em que o algoritmo tem dificuldade. Esse conjunto artificial possui um *cluster* com 420 instâncias e outro com 380. O resultado para o tamanho de memória $\mu = 153$ mostra que metade de um *cluster* foi identificada erroneamente como ruído (em vermelho), ao passo que a Figura 27 com $\mu = 154$ evidencia que os dois *clusters* foram unidos em apenas um, indicando que não existe um valor de μ capaz de separar corretamente os dois *clusters*.

Figura 26 - Conjunto de dados com clusters de formas arbitrárias. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 153$

Fonte: Elaborada pelo autor.

Figura 27 - Conjunto de dados com clusters de formas arbitrárias. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 154$

Fonte: Elaborada pelo autor.

Na Figura 28 temos um conjunto artificial de dados formado por duas espirais, cada uma contendo 500 instâncias. As cores indicam os *clusters* encontrados para caminhadas do turista de tamanho $\mu = 500$. Como podemos ver, o algoritmo favorece um determinado formato no *cluster* e não é capaz de "seguir" a espiral, uma vez que não é possível fechar o ciclo (atrator) e o caminhante "escapa" para o *cluster* vizinho.

Figura 28 - Conjunto de dados com forma de duas espirais. As cores representam os clusters encontrados com a caminhada do turista com $\mu = 500$

Fonte: Elaborada pelo autor a partir de dados de Chang e Yeung (2008).

Na Figura 29 podemos ver de forma mais clara como o algoritmo favorece um determinado formato de *cluster*. As instâncias identificadas com a cor vermelha indicam o ruído / *outliers*.

Figura 29 -Conjuntos de dados com diferentes formatos de *cluster* mostrando que os atratores da caminhada do turista têm preferência por formatos globulares

Fonte: Elaborada pelo autor a partir de dados de Piantoni et al. (2015).

5.2. Tourist Walk Partitional Clustering (TWPC)

Dadas as restrições apresentadas e ainda buscando por uma técnica de *clustering* particional baseada na caminhada do turista, propomos o algoritmo TWPC (*Tourist Walk Partitional Clustering*) que se define conforme descrito no Algoritmo 1.

Algoritmo 1 - Tourist Walk Partitional Clustering

Entrada: Matriz com ranking de vizinhos; Tamanho de memória μ	
Início:	
1	Para cada item i da Matriz de vizinhos e para uma memória μ
2	Encontrar a caminhada do turista $C_{i\mu}$ de memória μ iniciada no item i
3	Se uma caminhada qualquer $C_{i\mu}$ compartilha algum item com outra caminhada qualquer $C_{j\mu}$
4	Os itens das caminhadas são unidos em um grupo
5	Cada grupo de itens de caminhada(s) formam um <i>cluster</i>
Fim.	

Diferentemente do algoritmo hierárquico aglomerativo baseado na caminhada do turista e de uma possível versão particional descrita na seção 5.1 onde apenas o atrator é usado para identificar itens em um *cluster*, no TWPC usamos toda a

caminhada. Desse modo não podemos usar o transiente para identificar ruído ou *outliers*, os quais serão identificados usando outra abordagem que será descrita mais à frente.

Essa estratégia possui uma série de vantagens. Primeiramente, por usar a caminhada inteira, ela permite que os *clusters* sejam encontrados com um tamanho de memória μ muito menor, ou seja, com uma caminhada curta, já que o caminhante não tem muitas restrições e rapidamente se define um atrator. Isso permite que o algoritmo TWPC seja consideravelmente mais rápido. Outra vantagem é que há uma grande insensibilidade ao parâmetro de entrada (tamanho de memória μ), uma vez que uma grande faixa de valores produz o resultado desejado. Os atratores funcionam como "âncoras" que prendem os itens em seus *clusters* e diferentes tamanhos de "âncoras" podem produzir o resultado desejado.

Definir os parâmetros *a priori* de um algoritmo particional é um dos problemas mais complexos do aprendizado de máquina não supervisionado. Usualmente os algoritmos exigem informações sobre os dados, tais como o número de *clusters* presentes, distâncias entre os itens de um mesmo *cluster*, entre outros parâmetros complexos.

Aqui, para definir o valor apropriado da memória do caminhante μ , adotamos a média das silhuetas calculadas para cada item, conforme explicado na seção 2.3.1, como uma estratégia para uma versão completamente automática do algoritmo, sem que qualquer parâmetro seja exigido. O procedimento pode ser visto no Algoritmo 2.

Algoritmo 2 - Versão do TWPC onde a média das silhuetas dos itens é usada para definir o parâmetro μ de forma automática

	Entrada: Matriz com ranking de vizinhos
	Início:
1	Enquanto a Silhueta média atual > Silhueta média anterior
	Inicializa $\mu = 3$
2	Para cada item i da Matriz de vizinhos e para uma memória μ
3	Encontrar a caminhada do turista $C_{i\mu}$ de memória μ iniciada no item i
4	Se uma caminhada qualquer $C_{i\mu}$ compartilha algum item com outra caminhada qualquer $C_{j\mu}$
5	Os itens das caminhadas são unidos em um grupo
6	Cada grupo de itens de caminhada(s) formam um <i>cluster</i>
7	Silhueta média anterior = Silhueta média atual
8	Silhueta média atual = a média da silhueta de cada item
9	$\mu = \mu + 1$
10	Fim Enquanto
11	O valor de $\mu = \mu - 1$ é o apropriado. O menor μ com a maior Silhueta média.
	Fim.

Os experimentos mostraram que valores muito baixos de μ já produzem resultados satisfatórios e que tamanhos de memória pequenos produzem caminhadas pequenas, de modo que o custo computacional do TWPC é baixo mesmo com a necessidade de testar diferentes valores.

Para exemplificar o mecanismo considere o seguinte conjunto de dados artificiais formados por 4 *clusters* com 421, 86, 294 e 62 instâncias cada, conforme a Figura 30.

Figura 30 - Conjunto de dados com 4 clusters com variações inter-cluster e intra-cluster

Fonte: Elaborada pelo autor.

Usando o valor de $\mu = 3$ foram encontrados 12 *clusters*, como visto na Figura 31, e temos o valor médio de silhueta = 0,04, conforme a Figura 32.

Figura 31 – Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 3$

Fonte: Elaborada pelo autor.

Figura 32 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 3$

Fonte: Elaborada pelo autor.

Para o tamanho de memória $\mu = 4$ o TWPC encontrou 5 clusters (Figura 33) e temos o valor médio de silhueta = 0,59 (Figura 34).

Figura 33 - Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 4$

Fonte: Elaborada pelo autor.

Figura 34 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 4$

Fonte: Elaborada pelo autor.

Para o tamanho de memória $\mu = 5$ temos 4 clusters (Figura 35) e o valor médio de silhueta = 0,62 (Figura 36).

Figura 35 - Conjunto de dados com clusters com variações inter-cluster e intra-cluster. As cores representam os clusters encontrados pelo TWPC com $\mu = 5$

Fonte: Elaborada pelo autor.

Figura 36 - Valores das silhuetas de cada item para os clusters encontrados pelo TWPC com $\mu = 5$

Fonte: Elaborada pelo autor.

Para o tamanho de memória $\mu = 6$ temos também 4 clusters (Figura 35) e o mesmo valor médio de silhueta = 0,62 (Figura 36), ou seja, o valor estabilizou e podemos então usar $\mu = 5$ (o valor do passo anterior). Valores maiores, até um determinado limite, também irão gerar o mesmo resultado, porém é interessante usar

valores baixos de tamanho de memória para que as caminhadas sejam curtas, de modo que o algoritmo seja mais eficiente.

Note-se que o algoritmo TWPC proposto nos permite encontrar *clusters* em um conjunto de dados onde os *clusters* têm formas arbitrárias com variações *intra-cluster* e *inter-cluster*, sem que qualquer conhecimento *a priori* dos dados seja fornecido.

Como dito anteriormente, a definição de parâmetros *a priori* é uma tarefa complexa. Os algoritmos particionais existentes, para gerar bons resultados, exigem do usuário algum conhecimento dos dados e fornecem heurísticas complexas para estimar os parâmetros de entrada. Aqui iremos discutir como a dificuldade de encontrar parâmetros apropriados leva a resultados insatisfatórios mesmo em conjuntos de dados simples. Iremos analisar brevemente os algoritmos *K-means*, DBSCAN e *Affinity Propagation* (explicados na seção 3 de Trabalhos Correlatos).

O *K-means* é o algoritmo de *clustering* mais comum e exige como parâmetro de entrada o número de *clusters*, ou seja, é preciso ter essa informação sobre os dados (ou alguma técnica que sugira a quantidade de *clusters* que deve ser utilizada). O *K-means* e os métodos baseados nele estimam modelos, e quando os dados estão em conformidade com tais modelos os resultados são satisfatórios, ou seja, ele favorece um determinado formato de *cluster*. Considere por exemplo o conjunto de dados da Figura 37.

Figura 37 – Resultado obtido com o *K-means* em um conjunto com o cluster central alongado para $K = 4$

Fonte: Elaborada pelo autor.

Como podemos ver o *K-means* favorece um determinado formato de *cluster* e não lida bem com formas arbitrárias tais como *clusters* alongados. O item que representa o *cluster* acaba ficando muito distante dos itens nas extremidades.

O DBSCAN, como explicado na seção de trabalhos correlatos, não exige o número de *clusters* mas exige dois parâmetros: *eps* e *minPts*. Segundo os próprios autores, não existe uma maneira fácil de obter-se essa informação sobre todos os *clusters* do conjunto de dados. Os autores sugerem uma heurística para determinar esses valores. Essa heurística é implementada pelo pacote *dbscan* (HAHSLER; PIEKENBROCK, 2017) onde é possível gerar um gráfico com a média das distâncias de cada ponto para seus *K* vizinhos mais próximos. O valor de *K* pode ser especificado (corresponde ao *minPts*) e os autores sugerem usar o valor 4. As *K*-distâncias são exibidas em ordem crescente. Nesse gráfico devemos procurar pelo valor de *eps* onde a curva forma um "joelho". Temos na Figura 38 o gráfico para *minPts* = 4. Procuramos então um valor de *eps* próximo à curva do gráfico e esse valor está indicado na linha tracejada.

Figura 38 - Heurística para determinar *eps* no DBSCAN: o gráfico mostra a média das distâncias de cada item para seus *K* vizinhos mais próximos.

Fonte: Elaborada pelo autor.

Como podemos ver, um valor de *eps* = 0,8 parece apropriado. Usando então como parâmetro de entrada *eps* = 0,8 e *minPts* = 4 temos o resultado da Figura 39.

Figura 39 - Resultado obtido com o DBSCAN em um conjunto com o cluster central alongado com $\text{eps} = 0,8$

Fonte: Elaborada pelo autor.

Note-se na figura acima a existência de alguns itens indicados em vermelho que foram identificados como ruído. No *cluster* central, no entanto, houve a identificação de mais dois pequenos *clusters* (em amarelo e marrom). Apesar disso, é importante ressaltar que usando um eps maior ($= 2$) seria possível encontrar os *clusters* de forma apropriada, embora esse valor (2) esteja completamente fora da área de "joelho" e não seria encontrado através da heurística proposta.

No algoritmo *Affinity Propagation* também não é exigido o número de *clusters*, porém uma série de outros parâmetros podem ser exigidos. O algoritmo pode ser inicializado apenas com a matriz de similaridade, mas nesse caso vários parâmetros serão configurados automaticamente dependendo da implementação. O parâmetro mais importante é o *input preference* que define a tendência de cada item em se tornar um exemplar de seu *cluster*. Esse valor pode ser escolhido individualmente para cada item ou pode ser usado um valor compartilhado por todos. Essas preferências determinam de forma indireta o número de *clusters*. O pacote *apcluster* (BODENHOFER; KOTHMEIER; HOCHREITER, 2011) fornece um parâmetro q que permite ajustar essas preferências e assim retornar mais ou menos *clusters*. Outros parâmetros também podem ser configurados, tais como o número máximo de iterações que devem ser executadas e o *damping factor*. Aqui usaremos a função

apclusterK do pacote *apcluster*, onde podemos fornecer o número de *clusters* desejados e o algoritmo tenta diferentes combinações de preferências até que os exemplares encontrados formem o número desejado de *clusters*. É importante ressaltar que o algoritmo é computacionalmente custoso e essa estratégia para determinar os parâmetros ideais pode torná-lo inviável dependendo do tamanho do conjunto de dados. A Figura 40 mostra o resultado para 4 *clusters*.

Figura 40 - Resultado obtido com o Affinity propagation em um conjunto com o cluster central alongado com $K = 4$

Como podemos ver, por ser um algoritmo baseado em exemplares, acaba por favorecer um determinado formato de *cluster* e classifica erroneamente os itens nas extremidades do *cluster* central. É importante ressaltar, entretanto, que diferentes problemas exigem diferentes algoritmos, e os métodos que elegem exemplares ou que são baseados em centroides possuem aplicações relevantes.

O método particional proposto no TWPC encontra um resultado apropriado de forma automática, com $\mu = 5$, conforme a Figura 41.

Figura 41 - Resultado obtido com o TWPC em um conjunto de dados com o cluster central alongado com $\mu = 5$

Fonte: Elaborada pelo autor.

5.2.1. Ruído e *Outliers*

Conforme explicado anteriormente, quando consideramos apenas o atrator para definir se um determinado item pertence a um *cluster*, o transiente pode ser interpretado como ruído. Aqui, por usarmos toda a caminhada, perdemos essa característica. No algoritmo hierárquico podemos considerar como ruído ou *outlier* aqueles itens que foram agrupados em níveis mais altos do dendrograma e que não fazem parte de grandes grupos, pois à medida que a memória μ cresce, os atratores crescem e agregam os valores periféricos por último. No TWPC, além de usarmos toda a caminhada, usamos apenas um valor de μ (o valor que gerou o maior valor de silhueta). Com estas duas características, não temos a possibilidade de identificar os *outliers* do mesmo modo que o algoritmo hierárquico. Para tratar o problema do ruído e *outliers* propomos então uma outra estratégia:

- Considere a frequência com que cada item do conjunto de dados aparece no conjunto de caminhadas do turista iniciadas por todos os itens no conjunto de dados.
- Os itens cuja frequência esteja abaixo de um determinado limiar serão candidatos a ruído ou *outlier*.

Esta definição baseia-se na constatação de que o ruído ou *outlier* normalmente estará presente em poucas caminhadas. Este item estará presente na caminhada iniciada por si próprio, mas por estar em uma área esparsa ou isolado raramente será parte de uma caminhada de um outro item. O conceito de ruído e *outlier* é fortemente dependente do problema em questão e cabe ao pesquisador sua definição e como tratá-lo. A técnica descrita é capaz de filtrar alguns poucos candidatos, dependendo da distribuição dos dados; assim, os candidatos a ruído ou *outliers* podem ser avaliados usando alguma outra técnica definida pelo pesquisador, como por exemplo definir uma densidade mínima esperada em torno desses itens.

Considere o conjunto constituído por 402 instâncias distribuídas em 2 *clusters* e 2 *outliers*, mostrado na Figura 42. Ela exhibe os *clusters* encontrados pelo TWPC para o tamanho de memória $\mu = 5$. Este valor de μ foi encontrado usando a técnica da melhor média de silhueta e, portanto, seria o valor recomendado em uma aplicação real. Percebe-se que temos a separação em 2 *clusters*, porém os *outliers* foram associados a um dos *clusters* (em verde). Se, seguindo a técnica descrita para detecção de candidatos a *outliers*, selecionarmos todos os itens cuja frequência em caminhadas é menor que 2, temos os itens em vermelho da Figura 43.

Figura 42 - Conjunto de dados com 2 *clusters* e 2 *outliers*. As cores representam os *clusters* encontrados pelo TWPC com $\mu = 5$

Fonte: Elaborada pelo autor.

Figura 43 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 5$. Os itens em vermelho são itens com frequência em caminhadas < 2

Fonte: Elaborada pelo autor.

Note-se que temos como candidatos a *outliers* itens periféricos e isolados, mas também itens em áreas esparsas dentro do *cluster*. Podemos associar esses poucos itens a alguma outra técnica compatível com o que se considera ruído ou *outliers* para o problema investigado. Se desejarmos, no entanto, reduzir o número de candidatos, podemos dar uma vantagem à estratégia aumentando o tamanho da memória. Isso forçará uma restrição maior ao caminhante, que precisará executar caminhadas mais longas até ficar preso em um atrator, aumentando a probabilidade de que um item dentro de um *cluster* esteja contido em alguma caminhada.

A continuação podemos ver, em vermelho, os itens cuja frequência em caminhadas é menor que 2 para caminhadas com tamanho de memória $\mu = 50$ (Figura 44), $\mu = 100$ (Figura 45) e $\mu = 150$ (Figura 46).

Figura 44 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 50$. Os itens em vermelho são itens com frequência em caminhadas < 2

Fonte: Elaborada pelo autor.

Figura 45 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 100$. Os itens em vermelho são itens com frequência em caminhadas < 2

Fonte: Elaborada pelo autor.

Figura 46 - Conjunto de dados com 2 clusters e 2 outliers. As cores representam os clusters encontrados pelo TWPC com $\mu = 150$. Os itens em vermelho são itens com frequência em caminhadas < 2

Fonte: Elaborada pelo autor.

Como podemos ver, o número de candidatos a ruído ou *outlier* foi reduzindo-se à medida que o tamanho da memória μ foi aumentando. Vale notar que há uma consistência na redução de candidatos, de modo que, quanto maior a memória μ , os itens restantes foram os que estavam mais distantes dos *clusters*. Outro fator interessante é como o algoritmo não depende de um valor exato como parâmetro de entrada, ou seja, uma grande faixa de valores produz resultados satisfatórios. É importante ressaltar que o aumento do tamanho da memória μ degrada a performance do algoritmo, portanto seu uso deve ser bem avaliado.

5.3. Experimentos

Para verificarmos como o algoritmo TWPC proposto se comporta em diferentes problemas de *clustering*, vamos utilizar alguns conjuntos de dados artificiais que enfatizam certos desafios. Esses conjuntos são fornecidos pelo *Fundamental Clustering Problem Suite* (FCPS) (ULTSCH, 2005). Para encontrar o valor de μ apropriado foi utilizada a estratégia já descrita de calcular a média das silhuetas para valores de μ crescentes até que estabilize ou decaia.

Primeiramente temos um conjunto de itens com 212 instâncias separadas em 7 *clusters* em 3 dimensões com diferentes variâncias (Figura 47).

Figura 47 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes variâncias

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

O algoritmo TWPC foi capaz de identificar corretamente os *clusters* com um tamanho de memória $\mu = 4$.

O próximo conjunto de dados, mostrado na Figura 48, representa o problema de *clusters* com diferentes variâncias e diferentes distâncias *inter-cluster*. Esse conjunto possui 400 instâncias divididas em 3 *clusters*.

Figura 48 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes variâncias e distâncias inter-cluster

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

O algoritmo TWPC conseguiu identificar os *clusters* usando $\mu = 5$.

O conjunto de dados da Figura 49 possui 400 instâncias de 3 dimensões divididas em 4 classes. O desafio neste caso é que os *clusters* estão muito próximos.

Figura 49 - Fundamental Clustering Problem Suite - Conjunto de dados com Clusters muito próximos

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

Os *clusters* foram corretamente identificados usando $\mu = 3$.

O conjunto de dados da Figura 50 é representado por 1000 instâncias com 3 dimensões divididas em 2 *clusters*. O problema aqui representado é de dados que não são linearmente separáveis.

Figura 50 - Fundamental Clustering Problem Suite - Conjunto de dados não separável linearmente

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

O 2 *clusters* foram identificados usando $\mu = 13$.

O conjunto de dados da Figura 51 possui 1070 instâncias divididas entre dois *clusters*. O problema proposto é de dois *clusters* com densidades *intra-cluster* diferentes, onde as áreas mais densas estão em lados opostos.

Figura 51 - Fundamental Clustering Problem Suite - Conjunto de dados com diferentes densidades

Fonte: Distribuição dos dados baseada em Ultsch (2005)

O algoritmo TWPC foi capaz de identificar corretamente os 2 *clusters* usando $\mu = 5$.

5.3.1. Comparativo

A seguir, serão apresentados resultados comparativos entre o algoritmo proposto TWPC e outros algoritmos de *clustering* particionais. Aqui será investigada a dificuldade em definir os parâmetros ideais dos algoritmos e o problema de encontrar formas arbitrárias de *clusters* e com variações *intra-cluster* e *inter-cluster*.

Os parâmetros de entrada utilizados em cada algoritmo foram determinados seguindo os passos definidos por seus autores e descritos na seção 5.2.

Para o *K-means*, temos os resultados na Figura 52 e na Figura 56; para o DBSCAN, os resultados na Figura 53 e na Figura 57; para o *Affinity Propagation*, os resultados na Figura 54 e na Figura 58; e para a caminhada do turista, os resultados na Figura 55 e na Figura 59.

Figura 52 - As cores representam os clusters encontrados pelo K-means

Fonte: Elaborada pelo autor

Figura 53 - As cores representam os clusters encontrados pelo DBSCAN. Os itens em vermelho foram identificados como ruído

Fonte: Elaborada pelo autor

Figura 54 - As cores representam os clusters encontrados pelo Affinity propagation

Figura 55 - As cores representam os clusters encontrados pelo TWPC com $\mu = 6$

Figura 56 - As cores representam os clusters encontrados pelo K-means

Figura 57 - As cores representam os clusters encontrados pelo DBSCAN. O item em vermelho foi identificado como ruído

Fonte: Elaborada pelo autor a partir de dados de Chang e Yeung (2008).

Figura 58 - As cores representam os clusters encontrados pelo Affinity propagation

Fonte: Elaborada pelo autor a partir de dados de Chang e Yeung (2008).

Figura 59 - As cores representam os clusters encontrados pelo TWPC com $\mu = 2$

Fonte: Elaborada pelo autor a partir de dados de Chang e Yeung (2008).

5.3.2. Desvantagens

Como sabemos, a definição do que é um *cluster* depende do objetivo da análise. Um algoritmo pode por exemplo não separar *clusters* que estejam de algum modo ligados entre si. A Figura 60 mostra o problema descrito, onde o círculo pontilhado vermelho mostra a área em que os *clusters* estão colados.

Figura 60 - Conjunto complexo onde dois clusters estão unidos por uma "ponte" (destacada pelo círculo pontilhado vermelho)

Fonte: Distribuição dos dados baseada em Salvador e Chan (2004)

O algoritmo TWPC, para um tamanho de memória $\mu = 10$ gera o resultado mostrado na Figura 61.

Figura 61 - As cores indicam os clusters encontrados pelo TWPC com $\mu = 10$. Note que dois clusters foram identificados como apenas um (em marrom)

Fonte: Elaborada pelo autor a partir de dados de Salvador e Chan (2004)

Vale ressaltar, no entanto, que este problema é compartilhado por diversos algoritmos que não favorecem um determinado formato de *cluster* e que se "moldam" aos dados, não exigindo por exemplo que o número de *clusters* seja definido *a priori*. A consequência é que mesmo quando os *clusters* estiverem de algum modo unidos, pode ser que o objetivo da análise seja de considerá-los como *clusters* distintos. Esse efeito é bastante conhecido no algoritmo hierárquico *single link* e é conhecido como *single link effect*; em outros algoritmos também é conhecido como *chaining effect* ou simplesmente *chaining*.

O problema fica mais evidente no conjunto de dados mostrado na Figura 62.

Figura 62 - Dois clusters unidos ao centro por uma "ponte"

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

Podemos ver que o ponto verde funciona como uma ligação entre os dois *clusters*, fazendo com que uma caminhada possa passar para o *cluster* vizinho e uma solução aceitável não seja possível. A remoção desse único item, no entanto, gera o resultado desejado com tamanho de memória $\mu = 6$, como visto na Figura 63.

Figura 63 - Dois clusters bem definidos sem "ponte" entre eles

Fonte: Elaborada pelo autor a partir de dados de Ultsch (2005).

O algoritmo proposto incorpora naturalmente a premissa do aprendizado não supervisionado de que os dados “guiam” a solução, sem qualquer conhecimento *a priori* dos mesmos.

5.4. Otimização

A distribuição de frequências dos itens em caminhadas nos indica que há uma grande quantidade de repetições de trechos de caminhadas, especialmente de trechos atratores, uma vez que muitas caminhadas terminam no mesmo atrator. Isso nos motiva a desenvolver uma otimização em que não é necessário iniciar caminhadas em todos os itens. O processo pode ser visto no Algoritmo 3.

Algoritmo 3 - Algoritmo do TWPC com otimização

	Entrada: Matriz com ranking de vizinhos; Tamanho de memória μ
	Início:
1	Para cada item i da Matriz de vizinhos e para uma memória μ
2	Se o item i já está em alguma caminhada
3	Ir para a próxima iteração
4	Senão
5	Encontrar a caminhada do turista $C_{i\mu}$ de memória μ iniciada no item i
6	Se uma caminhada qualquer $C_{i\mu}$ compartilha algum item com outra caminhada qualquer $C_{j\mu}$
7	Os itens das caminhadas são unidos em um grupo
8	Cada grupo de itens de caminhada(s) formam um <i>cluster</i>
	Fim.

Os itens que não iniciarão caminhadas já estão agregados a um *cluster* por pertencer a uma caminhada, e os novos itens que essa caminhada não feita poderia encontrar irão eventualmente fazer parte de alguma caminhada ou, caso contrário, iniciar uma própria.

Como já explicado, tamanhos maiores de memória μ criam mais restrições ao caminhante, que por sua vez irá percorrer caminhadas maiores. Caminhadas maiores demandam um tempo de processamento maior, porém o ponto mais crítico do algoritmo está na fase em que as caminhadas são comparadas e eventualmente unidas. Caminhadas grandes tornam esse processo mais custoso, mas o número de caminhadas comparativamente é o maior responsável pelo custo computacional, razão pela qual essa otimização representa um grande ganho de performance. Para conjuntos de dados com grandes quantidades de itens, o custo de comparar n caminhadas, onde n é o número de itens do conjunto de dados, torna o custo do tamanho da caminhada pouco relevante.

Uma caminhada do turista terá um tamanho mínimo de $\mu + 1$ para o caso em que não há parte transiente e a caminhada já inicia em um atrator, no entanto, não há limitação do tamanho máximo, a não ser, é claro, o número de itens do conjunto. Dependendo da distribuição dos dados, um caminhante poderia percorrer todos os itens de um *cluster*. Considerando a otimização aqui apresentada, uma vez que não seria mais necessário iniciar novas caminhadas para os itens já encontrados na primeira caminhada, temos então no melhor caso que o número de caminhadas é o número de *clusters*. A capacidade da otimização de explorar o melhor caso o torna ainda mais eficiente. A Figura 64 mostra essa situação, onde com tamanho de memória $\mu = 3$ o algoritmo foi capaz de encontrar os *clusters* com apenas 3 caminhadas.

Figura 64 - O algoritmo TWPC com otimização foi capaz de encontrar os 3 clusters com apenas 3 caminhadas. As cores representam os clusters encontrados

Fonte: Elaborada pelo autor a partir de dados de Chang e Yeung (2008).

O ganho de performance fica evidente em grandes conjuntos de dados. Considere o conjunto da Figura 65, contendo 4811 instâncias divididas em um *cluster* com 2520 instâncias e outro com 2291.

Figura 65 - Conjunto de dados com 4811 instâncias

Fonte: Elaborada pelo autor

Para esse conjunto de dados, o menor tamanho de memória μ capaz de gerar o resultado desejado na técnica sem otimização foi $\mu = 6$, enquanto que na técnica com otimização foi $\mu = 40$, ou seja, a faixa de μ capaz de resultar nos *clusters* esperados difere nas duas técnicas. A técnica otimizada, contudo, foi 30 vezes mais rápida (com $\mu = 40$) que a sem otimização (com $\mu = 6$). A premissa do uso da média da silhueta de

cada item como estratégia de definição do parâmetro μ é de que a convergência do algoritmo se dará em número baixo de μ , que é iniciado em 3 e assume valores discretos crescentes, não acarretando então em um grande custo computacional. O fato de o menor μ da faixa que produz o resultado desejado ser relativamente mais alto ($\mu = 40$) poderia indicar que nesse caso a estratégia de medir as médias das silhuetas não seria apropriada. Todavia, os experimentos mostraram que a soma do tempo de processamento das tentativas de μ de 3 a 41 (necessário segundo a abordagem explicada em 5.2) foi 2,3 vezes menor que a soma do tempo de processamento das tentativas de μ de 3 a 6 na técnica sem otimização, justificando assim o uso da técnica da média da silhueta como estratégia para definição do μ ideal.

Além de ser mais rápido, a insensibilidade ao parâmetro de entrada, onde uma ampla faixa produz o resultado desejado, temos ainda que os valores dessa faixa são igualmente rápidos. Isso se deve ao fato de que tamanhos de memória μ maiores produzem caminhadas maiores, agrupando mais itens, e esses itens por sua vez não irão mais iniciar novas caminhadas. Esse fenômeno pode ser observado na Figura 66.

Figura 66 - Temos no eixo horizontal o tamanho da memória μ da caminhada do turista. No eixo vertical temos (1) o tamanho médio das caminhadas feitas (2) o número de caminhadas feitas no TWPC com otimização e (3) o tempo de processamento

Fonte: Elaborada pelo autor

Como podemos ver, o aumento do tamanho da memória μ aumenta o tamanho médio das caminhadas, diminuindo o número de caminhadas necessárias e permitindo uma compensação no tempo de processamento que se mantém constante dentro de uma faixa.

Isso pode levar a uma heurística para a definição de μ em que não é necessário se preocupar em usar valores baixos, uma vez que os valores em toda a faixa irão demandar na média o mesmo tempo de processamento. Para conjuntos de dados com *clusters* compactos e bem definidos como os da Figura 65, uma ampla faixa de μ produz o resultado desejado. Essa faixa normalmente compreende valores de μ de $n \times 0,01$ até $n \times 0,1$ para n itens no conjunto de dados. O valor médio dessa faixa, de $\mu = n \times 0,055$, gera um resultado satisfatório para diferentes conjuntos de dados. Essa faixa, no entanto, depende da configuração dos itens e não há garantias de como ela pode ser definida. Buscar um modo de incorporar algum conhecimento dos dados

para estimar um μ ideal, sem testar diversos valores, será objeto em um trabalho futuro.

Os resultados indicam que a versão otimizada tem grande potencial de uso para conjuntos de dados de grande escala.

6. Conclusões

O objetivo deste trabalho era demonstrar como a caminhada do turista pode ser usada no processo de *clustering*. Para tanto, foi implementado o algoritmo hierárquico baseado na caminhada do turista e foi demonstrado que a técnica é capaz de produzir dendrogramas que não impõem uma estrutura binária, podendo criar múltiplas fusões de ramos em um mesmo nível hierárquico. Isso resulta em uma árvore com um menor número de níveis e por conseguinte mais organizada. Ficou demonstrado que essa árvore ainda assim preserva uma correlação entre suas distâncias cofenéticas e a distância real entre os dados.

Além disso, foi proposto e implementado um novo algoritmo não supervisionado particional baseado na caminhada do turista. O método particional proposto, chamado aqui de TWPC, foi capaz, através de uma estratégia extremamente simples, de encontrar os *clusters* esperados em diferentes configurações de dados exigindo apenas um parâmetro. Esse parâmetro assume valores discretos e os experimentos mostraram que valores baixos já resultavam nos *clusters* desejados. Isso tornou possível a proposta de uma estratégia para definir esse parâmetro de modo automático, de forma que o algoritmo fosse capaz de encontrar os *clusters* desejados sem que qualquer informação sobre os dados fosse dada.

Ao final apresentamos ainda uma otimização para o algoritmo, tornando-o extremamente eficiente e apto a aplicações que envolvem grandes conjuntos de dados.

Algumas vantagens que podemos destacar do algoritmo TWPC:

- Possui um dos mecanismos mais simples dentro dos algoritmos de *clustering* particionais, tendo um valor didático e uma interessante aplicação de comportamento emergente, inspirado em fenômenos e movimentos da natureza.
- Não tem predileção por formatos de *cluster*, podendo encontrar formas arbitrárias.
- Não requer conhecimento *a priori* dos dados:
 - Não exige o número de *cluster*.
 - Requer apenas um parâmetro e uma faixa desses valores produz resultados satisfatórios.

- É determinístico. A maior parte dos algoritmos não o é e muitos dependem de estados iniciais aleatórios para um resultado adequado como o *K-means*.
- É rápido:
 - Não envolve cálculos complexos. Uma vez que a matriz de distância é transformada em um ranking de vizinhos, as distâncias não são mais usadas.
 - As caminhadas iniciadas nos itens são independentes e podem ser paralelizadas.
- O *cluster* é identificado por interações locais, não precisando do conhecimento *a priori* de todos os itens do conjunto.
- O conceito de ruído pode ser incorporado, usando dados já disponíveis (frequências dos itens nas caminhadas), sem introduzir mais complexidade.
- É invariável à escala, pois considera vizinhança e não distância, podendo, assim, identificar *clusters* com diferentes densidades sem perda de eficiência.
- Necessita apenas de um ranking de vizinhos independentemente de como ele é criado, podendo ser usada para tal qualquer medida que não precisa sequer ser métrica.

Entre as desvantagens podemos citar o fato de ser sensível a "pontes" entre os *clusters* (*single-link effect*). Outra questão é o fato de não exigir informações *a priori* sobre os dados. Via de regra isso é desejável e quase necessário em problemas não supervisionados mas uma vez que algumas informações são conhecidas, como o número de *clusters*, é interessante incorporar esse conhecimento ao algoritmo. À medida que se conheçam mais informações sobre os dados, métodos mais sofisticados devem ser utilizados.

É importante ressaltar, no entanto, que o processo de *clustering* não provê rótulos e apenas uma interpretação externa pode dar real significado aos dados, e tal significado depende do problema em questão, assim, nenhum algoritmo de *clustering* pode dar garantia que o resultado terá algum significado.

6.1. Trabalhos Futuros

Como propostas para trabalhos futuros, podemos citar:

Será estudada a relação dos tamanhos de memória μ capazes de gerar o resultado esperado com informações sobre os dados, procurando com isso incorporar algum conhecimento em uma heurística que permita escolher o tamanho de memória μ de um modo mais eficiente.

Será feita também uma implementação usando paralelismo para estudar como isso pode melhorar a performance do algoritmo.

Será estudado se a adição de uma fase de formação de grafo, restringindo os caminhos possíveis da caminhada, pode reduzir o *chaining effect*.

O trabalho apresentado representa um passo a mais no estado da arte das técnicas de aprendizado não supervisionado, e com isto contribui para a efetiva geração do conhecimento a partir da imensa quantidade de dados gerados diariamente pelo homem moderno.

Referências Bibliográficas

- AGGARWAL, C. C.; REDDY, C. K. Data Clustering: Algorithms and Applications. Boca Raton, FL: CRC Press, 2014.
- BACKES, A. R.; BRUNO, O. M.; CAMPITELI, M. G.; MARTINEZ, A. S. Deterministic Tourist Walks as an Image Analysis Methodology Based. Lecture Notes in Computer Science, p.784-793, 2006.
- BACKES, A. R.; GONÇALVES, W. N.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis and classification using deterministic tourist walk. Pattern Recognition, v.43, p.685-694, 2010.
- BALAN, A. G. R. Métodos adaptativos de segmentação aplicados à recuperação de imagens por conteúdo. 2007. Tese de Doutorado - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2007.
- BISHOP, C. M. Pattern Recognition and Machine Learning. New York, NY: Springer, 2006.
- BODENHOFER, U.; KOTHMEIER, A.; HOCHREITER, S. APCluster: an R package for affinity propagation clustering. Bioinformatics, v.27, p.2463-2464, 2011.
- BUNIMOVICH, L. A. Walks in rigid environments. PHYSICA A, v.279, p.169-179, 2000.
- CAMPITELI, M. G.; BATISTA, P. D.; KINOUCI, O.; MARTINEZ, A. S. Deterministic walks as an algorithm of pattern recognition. Physical Review E, v.74, 2006
- CHANG, H.; YEUNG, D. Y. Robust path-based spectral clustering. Pattern Recognition, v.41(1), p.191-203, 2008. Dataset disponível em: <<http://cs.joensuu.fi/sipu/datasets/>>. Acesso em 10 de jul. 2017.
- CORNE, D.; DORIGO, M.; GLOVER, F. New Ideas in Optimization. New York: McGraw-Hill, 1999.
- CROCETTI, G. M. Técnicas de Cluster e Análise de Redes Complexas para Segmentação de Imagens Médicas. 2014. Trabalho de Conclusão de Curso (Informática Biomédica) - Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto e Faculdade de Medicina de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2014.
- DAWES, R. M. The Robust Beauty of Improper Linear Models in Decision Making. American Psychologist, v.34, p.571-82, 1979.
- DUDA, R. O.; HART, P. E.; STORK, D. G. Pattern Classification. 2nd Ed. New York, NY: John Wiley & Sons, 2001.

DUECK, D.; FREY, B. J. Non-metric affinity propagation for unsupervised image categorization. In: 11th IEEE International Conference on Computer Vision (ICCV 2007), IEEE Press, p.1–8, 2007.

ESTER, M.; KRIEGEL, H.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise, In: 2nd International Conference on Knowledge Discovery and Data Mining. Palo Alto, CA: AAAI Press, p.226–231, 1996.

EVERITT, B. S.; LANDAU, S.; LEESE, M.; STAHL, D. Cluster analysis. 5th Ed. Chichester: John Wiley & Sons, 2011.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, v.7, p.1936-37, 1936.

FLORY, P. J. Principles of polymer chemistry. Ithaca: Cornell University Press, 1953.

FRANK, A.; ASUNCION, A. UCI machine learning repository. Irvine, University of California, School of Information and Computer Science. Disponível em: <<http://archive.ics.uci.edu/ml>>. Acesso em 10 de jul. 2017.

FREY, B.; DUECK, D. Clustering by passing messages between data points. *Science*, v.315, p.972–976, 2007.

GALILI, T. dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*, v.31(22), p.3718-20, 2015. Disponível em: <<https://CRAN.R-project.org/package=dendextend>>. Acesso em 10 de jul. 2017.

GOLDSTEIN, J. Emergence as a construct: History and issues. *Emergence: Complexity and Organization* 1, v.1, p.49–72, 1999.

GONZALEZ, R. C.; WOODS, R. W. Digital Image Processing. 3rd Ed. Upper Saddle River, NJ: Prentice Hall, 2008.

GORI, M.; MAGGINI, M.; SARTI, L. Exact and approximate graph matching using random walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.27, p.1100-11, 2005.

HAHSLER, M.; PIEKENBROCK, M. dbscan: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms. R package version 1.1-1. 2017, Disponível em: <<https://CRAN.R-project.org/package=dbscan>>. Acesso em 10 de jul. 2017.

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. Multivariate Data Analysis. 7th Ed. Upper Saddle River, NJ: Prentice Hall, 2010.

HAIWEI P.; LI J.; WEI Z. Medical image clustering for intelligent decision support. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference, v.3, p.3308-11, 2005.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textual Features for Image Classification. IEEE Transaction on Systems, Man and Cybernetics, v.SMC-3, i.6, p.610–621, 1973.

HEATH, M.; BOWYER, K.; KOPANS, D.; MOORE, R.; KEGELMEYER, W. P. The digital database for screening mammography. In: Proceedings of the 5th International Workshop on Digital Mammography, Medical Physics, p.212–218, 2001.

HENNIG, C. M.; MEILÄ, M.; MURTAGH, F.; ROCCI, R. Handbook of cluster analysis. Boca Raton: CRC Press, Taylor & Francis Group, 2016.

HENNIG, C.; LIAO, T. F. Comparing latent class and dissimilarity based clustering for mixed type variables with application to social stratification (with discussion). Journal of the Royal Statistical Society, Series C, v.62, p.309–369, 2013.

HOFFMAN, P. J.; SLOVIC, P.; RORER, L. G. An analysis-of-variance model for the assessment of configural cue utilization in clinical judgment. Psychological Bulletin, v.69, p.338-49, 1968.

JAIN, A. K.; DUBES, R. C. Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall, 1988.

KAHNEMAN, D. Thinking, fast and slow. New York: Farrar, Straus and Giroux, 2015.

KASSAMBARA, A.; MUNDT, F. factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.4.999. 2017, Disponível em: <<http://www.sthda.com/english/rpkgs/factoextra>>. Acesso em 10 de jul. 2017.

KING, B. Step-wise clustering procedures. Journal of the American Statistical Association, v.62 (317), p.86-101, 1967.

KINOUCI, O.; MARTINEZ, A. S.; LIMA, G. F.; LOURENCO, G. M.; RISAU-GUSMAN, S. Deterministic walks in random networks: an application to thesaurus graphs. PHYSICA A, v.315, p.665-676, 2002.

KOHONEN, T. Self-Organizing Maps. 3rd Ed. New York: Springer, 2001.

KOTSIANTIS, S.; PINTELAS, P. Recent Advances in Clustering: A Brief Survey. WSEAS Transactions on Inf., V.1, n.1, p.73-81, 2004.

LANCE, G.N.; WILLIAMS, W.T. A general theory of classificatory sorting strategies. Hierarchical systems, Computer Journal, v.9 (4), p.373–380, 1967.

LAPOINTE, F.; LEGENDRE, P. Comparison tests for dendrograms: A comparative evaluation. Journal of Classification. v.12, i.2, p.265–282, 1995.

LIMA G. F.; MARTINEZ A. S.; KINOUCI O. Deterministic walks in random media. Physical Review Letters, v.87, 2001.

LUXBURG, U.; WILLIAMSON, R.; GUYON, I. Clustering: Science or art? JMLR Workshop and Conference Proceedings, v.27, p.65–79, 2012.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, Calif.: University of California Press, v.1: Statistics, p.281–297, 1967.

MADRAS, N. N.; SLADE, G. The self-avoiding walk (reprint of the 1996 edition). New York: Birkhäuser, 2013.

MAECHLER, M.; ROUSSEEUW, P.; STRUYF, A.; HUBERT, M.; HORNIK, K. cluster: Cluster Analysis Basics and Extensions. R package version 2.0.5. 2016, Disponível em: <<https://CRAN.R-project.org/package=cluster>>. Acesso em 10 de jul. 2017.

MILLIGAN, G. W. Clustering validation: Results and implications for applied analyses. Clustering and Classification, Singapore: World Scientific, p.341–375, 1996.

MIRANDA, G. H. B.; SOARES, E. G. ; BARRERA, J. ; FELIPE, J. C. . Method to Support Diagnosis of Cervical Intraepithelial Neoplasia (CIN) Based on Structural Analysis of Histological Images. In: 25th International Symposium on Computer-Based Medical Systems (CBMS), 2012. IEEE, 2012, Roma. Proceedings of the 25th International Symposium on Computer-Based Medical Systems, v.1. p.1-6, 2012.

MITCHELL, T. M. Machine Learning. Singapore: McGraw-Hill, 1997.

NG, R. T.; HAN J. Efficient and Effective Clustering Methods for Spatial Data Mining. Proc. 20th Int. Conf. on Very Large Data Bases, Santiago, Chile, p.144-155, 1994.

PARADIS E.; CLAUDE J.; STRIMMER K. APE: analyses of phylogenetics and evolution in R language. Bioinformatics v.20, p.289-290, 2004. Disponível em: <<https://CRAN.R-project.org/package=ape>>. Acesso em 10 de jul. 2017.

PEARSON, K. The Problem of the Random Walk. Nature, v.72, n.1867, p.294, 1905.

PIANTONI, P.; FACELI, K.; SAKATA, T. C.; PEREIRA J. C.; SOUTO, M. C. P. Impact of Base Partitions on Multi-objective and Traditional Ensemble Clustering Algorithms. In: Neural Information Processing, Lecture Notes in Computer Science, v. 9489, p.696-704, 2015. Dataset disponível em: <<http://lasid.sor.ufscar.br/2sp2globBPCollection>>. Acesso em 10 de jul. 2017.

ROUSSEEUW, P.J. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics, v.20, p.53–65, 1987.

SALVADOR, S.; CHAN, P. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. In Proceedings of the 16th IEEE

International Conference on Tools with Artificial Intelligence (ICTAI '04), Washington, DC: IEEE Computer Society, p.576-584, 2004.

SARAÇLI, S.; DOĞAN, N.; DOĞAN, I. Comparison of hierarchical cluster analysis methods by cophenetic correlation. *Journal of Inequalities and Applications*, v.1, p.1-8, 2013.

SHANTEAU, J. Psychological characteristics and strategies of expert decision makers. *Acta Psychologica*, v.68, p.203-215, 1988.

SILVA, T. C.; ZHAO, L. High-level pattern-based classification via tourist walks in networks. *Information Sciences*, v.294, p.109-126, 2015.

SILVA, T. C.; ZHAO, L. Machine learning in complex networks. Springer International Publishing, 2016.

SILVA, T. C.; ZHAO, L. Network-Based High Level Data Classification. *IEEE Transactions on Neural Networks and Learning Systems*. v.23, i.6, p.954-970, 2012.

SOKAL, R. R.; ROHLF, F. J. The comparison of dendrograms by objective methods. *Taxon*, v.11, n.2, p.33-40, 1962.

STANLEY, H. E.; BULDYREV, S. V. Statistical physics: The salesman and the tourist. *Nature*, v.413, p.373, 2001.

STONEBRAKER, M.; FREW, J.; GARDELS, K.; MEREDITH J. The SEQUOIA 2000 Storage Benchmark. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Washington, DC, p.2-11, 1993.

THEODORIDIS, S.; KOUTROUMBAS, K. Pattern Recognition. Asterdam: Elviesier, 2009.

ULTSCH, A. Clustering with SOM: U*C. In *Proc. Workshop on Self-Organizing Maps*, Paris, p.75-82, 2005.

WOLF, T. D.; HOLVOET, T. Emergence versus self-organisation: Different concepts but promising when combined. In *Engineering Self-Organising Systems*, Berlin/Heidelberg: Springer-Verlag, p.1-15, 2005.

XU R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v.16, n.3, p.645-678, 2005.

XU, D.; TIAN, Y. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*. v.2, p.165-193, 2015.

YANG, S. J. Exploring complex networks by walking on them. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*. v.71, 2005.

ZELNIK-MANOR, L.; PERONA, P. Self-Tuning Spectral Clustering. *Advances in Neural Information Processing Systems 17*, MIT Press, p.1601-1608, 2005.