UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

IRINEU JUNIOR PINHEIRO DOS SANTOS

TRACTS: Um Método para Classificação de Trajetórias de Objetos Móveis Usando Séries Temporais

Dissertação de Mestrado

Prof. Dr. Luis Otávio Álvares
Orientador

Porto Alegre, novembro de 2011

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Santos, Irineu Júnior Pinheiro dos

TRACTS: Um Método para Classificação de Trajetórias de Objetos Móveis Usando Séries Temporais/ Irineu Júnior Pinheiro dos Santos – Porto Alegre: Programa de Pós-Graduação em Computação, 2011.

70 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2011. Orientador: Luis Otávio Álvares.

1.Descoberta de conhecimento. 2.Dados espaço-temporais. 3.trajetórias. I. Álvares, Luis Otávio.

SUMÁRIO

SUMÁRIO 3
LISTA DE ABREVIATURAS E SIGLAS 5
LISTA DE FIGURAS 6
LISTA DE TABELAS 7
RESUMO 8
ABSTRACT 9
1 INTRODUÇÃO 10
1.1 Motivação 10
1.2 Contribuição desse Trabalho 11
1.3 Estrutura do Texto 12
2 BASE CONCEITUAL 13
2.1 Trajetórias 13
2.2 Descoberta de Conhecimento e Classificação 13
2.3 Séries Temporais 14
2.3.1 Análise de tendências 15
2.3.2 Busca de similaridade em uma série temporal 16
2.3.3 Redução de dados 16
2.4 O Método SAX 17
2.4.1 Representação Simbólica de Múltiplas Séries Temporais 19
2.5 Matrizes da Série Temporal 21
2.6 Ferramenta de Análise Weka 24
2.6.1 Pré-Processamento de Dados 24
2.6.2 Mineração de Dados 24
2.7 Trabalhos Relacionados 29
3 MÉTODO *TRACTS* 33
3.1 Introdução 33
3.2 Preparação 34
3.2.1 Formatação dos dados 34
3.2.2 Reconstrução e limpeza das trajetórias 35
3.3 Caracterização 37
3.3.1 Extração de características no método *TRACTS* 39
3.4 Transformação 42
3.4.1 Aplicação do método SAX 43
3.4.2 Aplicação do método TSB 44
3.5 Classificação 45
3.5.1 Ferramenta de classificação 46
3.5.2 Montagem do arquivo de classificação 46
3.5.3 Pré-processamento e escolha do algoritmo de classificação 48
3.5.4 Geração do modelo de classificação 48
3.5.5 Avaliação do modelo gerado 48
4 EXPERIMENTOS REALIZADOS 50
4.1 Experimentação – Trajetórias de Animais 51
4.2 Experimentação – Trajetórias de Barcos 56
4.3 Experimentação – Trajetórias de Furacões 61
4.4 Resultado Comparativo 65
5 CONCLUSÃO 66

5.1 Contribuições..... 66
5.2 Artigo Publicado 67
5.3 Trabalhos Futuros 67
REFERÊNCIAS..... 68

LISTA DE ABREVIATURAS E SIGLAS

APCA	*Adaptative piecewise constant approximation*
DFT	*Discrete Fourier transform*
DWT	*Discrete wavelet transform*
GPS	*Global Positioning System* (Sistema de Posicionamento Global)
KDD	*Knowledge Data Discovery* (Descoberta de Conhecimento em Dados)
KNN	*K-Nearest Neighbors* (K-vizinhos mais próximos)
MCMC	*Monte Carlo Markov Chain* (Cadeia de Makov Monte Carlo)
PAA	*Piecewise Aggregate Approximation*
PCA	*Principle Component Analysis*
POI	*Point of Interest* (Ponto de Interesse)
RFID	*Radio-Frequency Identification* (Identificação por Rádio Freqüência)
SAX	*Symbolic Aggregate Approximation*
SVD	*Singular value decomposition*
SVM	*Suport Vector Machines*
TSB	*Time Series Bitmap* (Mapa de bits de Séries Temporais)
UTM	*Universal Transverse Mercator*
WEKA	*Waikato Enviroment for Knowledge Analysis*

LISTA DE FIGURAS

Figura 1: Série temporal de uma ação na bolsa de valores 15
Figura 2: Redução dimensional (Lin, Keogh, Wei, & Lonardi, 2007) 17
Figura 3: Representações de séries temporais (Lin, Keogh, Lonardi, & Chiu, 2003) 18
Figura 4: Série temporal na representação PAA 19
Figura 5: Método SAX 20
Figura 6: Bitmap de série temporal 22
Figura 7: Bitmaps de C_1 e C_2 22
Figura 8: Apresentação visual dos bitmaps de C_1 e C_2 23
Figura 9: Apresentação visual de bitmaps de genomas (Kumar, Lolla, Keogh, Lonardi, Ratanamahatana, & Wei, 2005) 23
Figura 10: Arquivo de entrada de dados Weka 25
Figura 11: Resultado da etapa de mineração de dados 26
Figura 12: Resultado da etapa de mineração de dados após discretização 28
Figura 13: Método *TRACTS* ..... **Erro! Indicador não definido.**
Figura 14: Segmentação de trajetória 36
Figura 15: Ruído no conjunto de dados 37
Figura 16: Trajetória e algumas de suas características 38
Figura 17: Arquivo ARFF para os animais (configuração *TRACTS*.3.1) 53
Figura 18: Mineração de dados de trajetórias de animais 54
Figura 19: Atributo "Duração" para as trajetórias de barcos 58
Figura 20: Mineração dos dados de trajetórias dos barcos 59
Figura 21: Trajetórias de furacões 61
Figura 22: Mineração dados de trajetórias de furacões 63

LISTA DE TABELAS

Tabela 1: Obtenção dos símbolos SAX 21
Tabela 2: Geração das séries temporais para os animais 51
Tabela 3: Tempo de geração da *string* SAX para os animais 52
Tabela 4: Tempo de geração das matrizes TSB para os animais 52
Tabela 5: Resultado geral do método *TRACTS* para os animais 55
Tabela 6: Resultado comparativo para as trajetórias de animais 55
Tabela 7: Geração das séries temporais para os animais 56
Tabela 8: Tempo de geração da *string* SAX para os barcos 57
Tabela 9: Tempo de geração das matrizes TSB para os barcos 57
Tabela 10: Resultado geral do método TRACTS para os barcos 60
Tabela 11: Resultado comparativo para as trajetórias de barcos 60
Tabela 12: Classes dos furacões 61
Tabela 13: Geração das séries temporais para os furacões 62
Tabela 14: Tempo de geração da *string* SAX para os furacões 62
Tabela 15: Tempo de geração das matrizes TSB para os furacões 63
Tabela 16: Resultado geral do método TRACTS para os furacões 64
Tabela 17: Resultado comparativo para as trajetórias de furacões 64
Tabela 18: Comparação dos resultados 65

RESUMO

O crescimento do uso de sistemas de posicionamento global (GPS) e outros sistemas de localização espacial tornaram possível o rastreamento de objetos móveis, produzindo um grande volume de um novo tipo de dado, chamado *trajetórias de objetos móveis*. Existe, entretanto, uma forte lacuna entre a quantidade de dados extraídos destes dispositivos, dotados de sistemas GPS, e a descoberta de conhecimento que se pode inferir com estes dados. Um tipo de descoberta de conhecimento em dados de trajetórias de objetos móveis é a classificação. A classificação de trajetórias é um tema de pesquisa relativamente novo, e poucos métodos tem sido propostos até o presente momento. A maioria destes métodos foi desenvolvido para uma aplicação específica. Poucos propuseram um método mais geral, aplicável a vários domínios ou conjuntos de dados. Este trabalho apresenta um novo método de classificação que transforma as trajetórias em séries temporais, de forma a obter características mais discriminativas para a classificação. Experimentos com dados reais mostraram que o método proposto é melhor do que abordagens existentes.

Palavras-Chave: trajetória de objetos móveis, classificação de trajetórias, descoberta de conhecimento, mineração de dados, séries temporais.

ABSTRACT

The growing use of global positioning systems (GPS) and other location systems made the tracking of moving objects possible, producing a large volume of a new kind of data, called trajectories of moving objects. However, there is a large gap between the amount of data generated by these devices and the knowledge that can be inferred from these data. One type of knowledge discovery in trajectories of moving objects is *classification*. Trajectory classification is a relatively new research subject, and a few methods have been proposed so far. Most of these methods were developed for a specific application. Only a few have proposed a general method, applicable to multiple domains or datasets. This work presents a new classification method that transforms the trajectories into time series, in order to obtain more discriminative features for classification. Experiments with real trajectory data revealed that the proposed approach is more effective than existing approaches.

Keywords: moving object trajectory, trajectory classification, knowledge discovery, data mining, time series.

1 INTRODUÇÃO

1.1 Motivação

Nos anos recentes, observamos o crescimento do uso de sistemas baseados em localização, como o Sistema de Posicionamento Global (Global Positioning System – GPS), triangulação de rádio e de celulares, sistemas wireless e RFID. Esses sistemas possibilitaram e facilitaram o rastreamento de objetos móveis com uma resolução de apenas alguns metros no espaço, ou com maior precisão ainda, em alguns casos.

O aumento do uso de tais sistemas está levando à maior disponibilidade de dados espaço-temporais. Colares RFID monitoram, durante dias seguidos, as diversas posições geográficas de animais. Aparelhos de GPS acoplados a veículos possibilitam o rastreamento do trajeto percorrido por automóveis e outros tipos de veículos nas estradas. Celulares, também equipados de rastreadores GPS, armazenam os diversos locais frequentados por pessoas.

Essa coleção de dados espaço-temporais representa um ambiente extremamente rico para análise de comportamento individual e coletivo. A análise dessas informações pode ser possível a partir do estudo de trajetórias, que podem ser construídas com os dados coletados a partir desses dispositivos baseados em localização. A mineração de dados espaço-temporais e a descoberta de conhecimento a partir dessas trajetórias têm emergido como um tópico frequente em pesquisas acadêmicas, bem como em aplicações práticas.

A análise de diversas trajetórias de objetos móveis permite, por exemplo, realizar a identificação de lugares importantes, ou POIs (*Points of Interest*), para esse determinado objeto móvel. Isso possibilita a armazenagem de históricos e frequência de sua localização em cada um desses POIs (Hariharan & Toyama, 2004), descobrir a sequência de locais que geralmente são visitados pelo objeto móvel (Zhou, Bhatnagar, Shekhar, & Terveen, 2007), realizar a predição de destinos possíveis (Ashbrook & Starner, 2003) e até mesmo permitir que um usuário seja avisado em tempo real que está pegando um ônibus errado (Liao, Patterson, Fox, & Kautz, 2006).

Em outra abordagem, quando foca-se nas propriedades geométricas das trajetórias, é possível realizar a detecção de trajetórias similares, estabelecendo correlação entre objetos móveis (Tiakas, Papadopoulos, Nanopoulos, Manolopoulos, Stojanovic, & Kajan, Maio 2009), padrões de congruência das trajetórias (Gudmundsson, Kreveld, & Speckmann, 2007) e predições de comportamento, como, por exemplo, no tempo de viagem de um veículo (Tiesyte & Jensen, 2008). Indo além da análise das propriedades geométricas das trajetórias e considerando-se também os objetos geográficos estáticos em suas proximidades, tais como hotéis, pontos turísticos de uma cidade e pedágios, é possível atribuir semântica a essas trajetórias, estabelecendo, por exemplo, rotas

turísticas ou trechos de congestionamento no fluxo do trânsito (Palma, Bogorny, Kuijpers, & Alvares, 2008).

Uma forma de análise dos dados espaço-temporais, que ainda foi pouco explorada na literatura atual, é a classificação de trajetórias. Essa abordagem utiliza o mesmo conceito que a tarefa de classificação tradicional da mineração de dados (Tan, Steinbach, & Kumar, 2006), onde é aprendida uma função alvo f que mapeia um conjunto de atributos x para um dos rótulos de classe predefinidos y. Dessa forma é possível realizar um processo de descoberta de conhecimento em diversos domínios de aplicação diferentes. Na classificação de trajetórias, a função f, ou modelo de classificação, mapeia cada trajetória para um determinado rótulo de classe, que possibilita a sua identificação. Esse rótulo de classe y determina o objetivo do processo de classificação, que pode ser desde a identificação de diferentes tipos de objetos móveis (Lee, Han, Gonzalez, & Li, 2008), até a identificação de meios de transporte (carro, bicicleta, etc.) utilizado pelo objeto móvel (Zheng, Liu, Wang, & Xie, 2008).

Entre os métodos existentes para classificação de trajetórias, a técnica que é utilizada para realizar a identificação das características para a classificação pode variar bastante. É possível utilizar, por exemplo, técnicas de análise de similaridade (Panagiotakis, Pelekis, & Kopanakis, 2009) ou de análise de regiões do espaço (Lee, Han, Gonzalez, & Li, 2008). Entretanto, a maioria dos trabalhos geralmente descreve métodos específicos para a aplicação que está sendo abordada.

Este trabalho apresenta um novo método para classificação de trajetórias, denominado *TRACTS* (***Tra****jectory* ***C****lassification using* ***T****ime* ***S****eries*), baseado na transformação de cada trajetória em um conjunto de séries temporais para que destas sejam extraídas as características que serão submetidas ao processo de geração do modelo de classificação. Este método é, no nosso conhecimento, o primeiro método que utiliza séries temporais para a classificação de trajetórias.

1.2 Contribuição desse Trabalho

Existem poucos trabalhos, atualmente, que utilizam a classificação de trajetórias como um método de análise de dados espaço temporais. Desses trabalhos, nem todos consideram a possibilidade de generalização do método utilizado, o que possibilitaria a realização do processo de classificação das trajetórias em diversos domínios de aplicação diferentes.

As trajetórias consideradas são produzidas por objetos em movimento. Dessa forma, um dos pontos fundamentais para possibilitar a generalização do método proposto, é considerar que algumas características são comuns a todos esses objetos móveis em qualquer domínio considerado. Na grande maioria dos casos, essas características distinguem uma trajetória (ou um grupo delas) das demais, possibilitando que elas sejam utilizadas como critérios de classificação das trajetórias.

Este trabalho propõe um conjunto inicial padrão de características importantes das trajetórias em qualquer domínio considerado, permitindo uma grande flexibilidade de uso do método de classificação de trajetórias para os domínios desejados.

Outra funcionalidade desejável é a de se utilizar o vasto suporte atualmente existente para o processo de classificação tradicional de dados. Entretanto, para se utilizar os algoritmos tradicionais de classificação, os dados espaço-temporais necessitam passar

por algum processo de transformação, processo esse que também é contemplado no presente método.

O método *TRACTS* propõe a transformação das características extraídas das trajetórias em atributos, através de técnicas que utilizam séries temporais, permitindo a discretização dos valores contínuos das características em atributos discretos propícios para o processo de classificação de dados tradicional. Essa transformação é realizada de forma que não haja prejuízo para a generalização do método proposto.

1.3 Estrutura do Texto

O restante do trabalho é organizado desta forma: o Capítulo 2 apresenta a base conceitual utilizada nesse artigo bem como alguns trabalhos relacionados. A proposta do método de classificação *TRACTS* é abordada no Capítulo 3. Os experimentos e resultados comparativos são descritos no Capítulo 4. Finalmente, o Capítulo 5 apresenta as conclusões do trabalho e propõe trabalhos futuros.

2 BASE CONCEITUAL

Nesse capítulo, será apresentada a base conceitual utilizada para construir o método de classificação descrito nesse trabalho. Será realizada uma revisão conceitual de trajetórias e do processo de descoberta de conhecimento. Logo após, aborda-se séries temporais e alguns métodos de tratamento de séries temporais, tais como SAX e TSB. A ferramenta *Weka*, utilizada extensivamente na mineração de dados para esse método, também terá as suas funcionalidades mais relevantes para o presente trabalho discutidas brevemente.

No final do capítulo, serão descritos os principais trabalhos relacionados à classificação de trajetórias.

2.1 Trajetórias

Trajetória é o nome dado ao percurso realizado por um dado objeto móvel através do espaço em função do tempo. Na matemática discreta, trajetória é uma sequência $\left(f^k(x)\right)_{k\in\mathbb{N}}$ de valores calculados pela iterada aplicação do mapeamento de f para um elemento x. De forma geral, uma trajetória se refere ao conjunto ordenado de estados intermediários assumido por um sistema dinâmico como resultado de uma evolução temporal.

É interessante notar que a trajetória de um objeto móvel depende do referencial que se esteja utilizando, uma vez que não existe espaço absoluto. Assim é que, por exemplo, para um observador fixo em relação ao planeta Terra, a trajetória de um foguete sendo lançado de uma base na Terra, que acaba ficando curva logo após o lançamento, é totalmente diversa da registrada por um observador situado no interior da Estação Espacial Internacional, para o qual a trajetória do foguete se mantém em linha reta na maior parte do tempo após seu lançamento. Como regra geral, os sistemas referenciais mais utilizados para identificar a posição de um objeto móvel são baseados em sistemas de coordenadas geográficas do planeta Terra.

Dessa forma, considerando-se um espaço tridimensional, a componente posicional do objeto móvel na trajetória pode ser representada pelas coordenadas (x, y, z). O tempo, o qual pode ser representado por t, indica o momento de cada posição do objeto na trajetória. Tomando id como a identificação do objeto que está em movimento, podemos definir trajetória como uma sequência de pontos, cada um dos quais representados pela tupla (id, x, y, z, t).

2.2 Descoberta de Conhecimento e Classificação

O processo da descoberta de conhecimento em base de dados, internacionalmente conhecido como *Knowledge Data Discovery* (KDD) é a resposta para a análise do imenso volume de dados em bases de dados operacionais e científicas. Onde as técnicas analíticas e de consultas tradicionais falham, o KDD tenta transformar os dados brutos em informações, possibilitando o uso dessa informação para a obtenção de conhecimento sobre o domínio considerado. A ideia central do KDD é que existe

conhecimento oculto em grandes bases de dados, na forma de padrões interessantes e previamente desconhecidos.

Um dos métodos mais utilizados para realizar descoberta de conhecimento em base de dados é a classificação. Segundo (Tan, Steinbach, & Kumar, 2006), classificação é a tarefa de aprender uma função alvo f que mapeia cada conjunto de atributos x para um dos rótulos de classe predefinidos y. A função alvo é também conhecida como um modelo de classificação, o qual é produzido a partir dos dados de entrada do algoritmo de classificação, que por sua vez formam os conjuntos de treinamento e de teste. Esse modelo de classificação é capaz de predizer o valor do atributo classe (rótulo de classe) de exemplos não vistos previamente, baseado apenas nos valores dos seus atributos preditivos.

Um elemento crucial ao processo de classificação é a precisão com a qual os rótulos de classe são atribuídos às instâncias desconhecidas. Para que essa precisão possa ser estimada, uma fase de teste é empregada. Nessa fase de teste, é utilizada uma parte dos dados, que não foi utilizada anteriormente na fase de treinamento, quando foi construído o modelo de classificação. Esse conjunto de dados é chamado de conjunto de teste. O modelo aprendido então é aplicado no conjunto de teste e suas predições são comparadas com os reais rótulos de classe das instâncias de teste. Uma das medidas mais utilizadas para avaliar o desempenho da classificação é a acurácia, que é o percentual das instâncias corretamente preditas no conjunto de teste.

Vários algoritmos podem ser utilizados para a construção do modelo de classificação. Dependendo dos algoritmos utilizados, a estrutura dos modelos de classificação resultantes podem ser muito diferentes um do outro (árvore de decisão, modelo estatístico, modelo não linear, etc.). Entre os exemplos de algoritmo de classificação (e seus tipos de modelos de classificação gerados) estão:

- *Naive Bayes* e *Support Vector Machines* (modelo estatístico);
- C4.5 (árvore de decisão);
- *Ripper* (regras de classificação);
- *K-nearest neighbors* ou KNN (aprendizado baseado em instância);
- *Multilayer Perceptron* (modelo não linear);

A análise e classificação de padrões de movimentação, mais especificamente de trajetórias, possibilitam a rotulação das trajetórias de um objeto movimentando-se no domínio espaço-temporal. Essa rotulação tem como objetivo realizar a identificação do comportamento do objeto, de forma a classificá-lo de acordo com um determinado critério pré-estabelecido. Dessa forma, padrões de movimentação ajudam a identificar comportamentos, ou seja, conhecimento interessante, ocultos nos conjuntos de dados formados por objetos móveis (Laube, Kreveld, & Imfeld, 2004). Isso evidencia o grande potencial do uso do conceito tradicional de classificação de dados para realizar descoberta de conhecimento em grandes conjuntos de dados de trajetórias de objetos móveis.

2.3 Séries Temporais

Quando falamos de trajetórias, é possível traçar uma analogia bastante próxima ao conceito de séries temporais. Segundo (Kamber & Han, 2006), séries temporais consistem de sequência de valores ou eventos obtidos sobre repetidas medidas de tempo. Os valores são geralmente medidos em intervalos regulares de tempo (a cada

hora, diariamente, semanalmente, etc.). Aplicações envolvendo séries temporais têm sido muito utilizadas em áreas como bolsa de valores, análise orçamentária, projeções de produtividade, controle de qualidade, observações de fenômenos naturais (temperatura, ventos, terremotos, etc.) e tratamento médico.

Com o crescente uso de sensores remotos, dispositivos de telemetria e outras ferramentas de coleta on-line de dados, a quantidade de dados de séries temporais está aumentando rapidamente.

É possível realizar diversos tipos de análise a partir de séries temporais. Dois importantes tipos que podem ser citados são a análise de tendências e a busca de similaridades em séries temporais. Esses dois tipos de análises são detalhados a seguir.

2.3.1 Análise de tendências

Uma série temporal envolvendo uma variável Y, representando, por exemplo, o valor de fechamento diário de uma ação na bolsa de valores, pode ser visto como uma função do tempo t, ou seja, $Y = F(t)$. Essa função pode ser ilustrada como um gráfico de série temporal, tal como a ilustração mostrada na Figura 1.

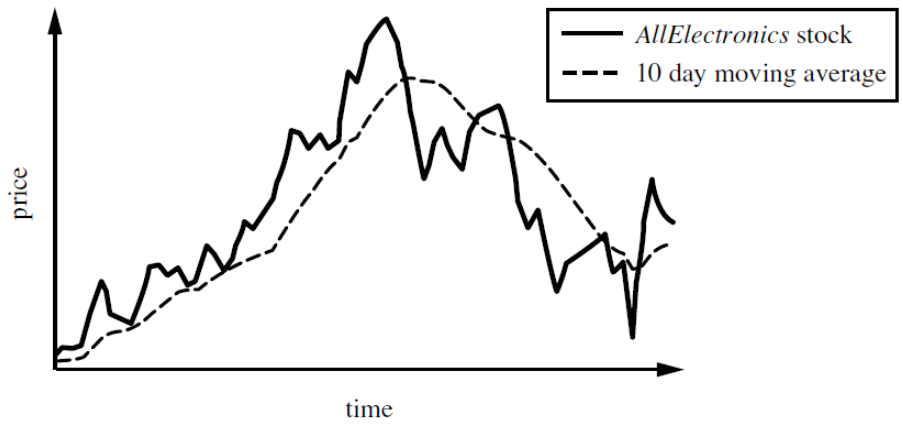


Figura 1: Série temporal de uma ação na bolsa de valores

A análise de tendências de séries temporais consiste de quatro grandes componentes, ou movimentos, para caracterização dos dados de uma série temporal. Eles são detalhados a seguir.

2.3.1.1 Movimentos de tendência ou de longa duração

Indica a direção geral para a qual um gráfico de série temporal está se movendo sobre um longo intervalo de tempo. Na Figura 1, essa curva (ou linha) de tendência está representada por uma linha pontilhada.

2.3.1.2 Movimentos ou variações cíclicas

São ciclos, ou seja, oscilações de grande duração sobre uma curva de tendência, que podem ou não ser periódicos. Isto é, os ciclos não necessariamente seguem padrões similares em intervalos regulares de tempo.

2.3.1.3 Movimentos ou variações sazonais

São sistemáticos, ou relacionados a um calendário. Um exemplo desse tipo de movimento é a súbita aumenta de venda de flores antes do dia dos namorados, ou de

uma loja de departamento antes do natal. Nesses exemplos, movimentos sazonais são padrões idênticos, ou quase idênticos, que uma série temporal parece seguir durante os meses correspondentes de sucessivos anos.

2.3.1.4 Movimentos randômicos ou irregulares

São caracterizados pelo movimento esporádico de séries temporais devido a eventos randômicos, como enchentes, disputas trabalhistas e mudança de empregados dentro de uma companhia.

2.3.2 Busca de similaridade em uma série temporal

Diferentemente de consultas em banco de dados, que encontram um dado que segue exatamente os parâmetros da consulta, uma busca de similaridade encontra sequências de dados que diferem muito pouco de um dado parâmetro de consulta.

Dada uma sequencia de série temporal C, existem dois tipos de busca de similaridade: **busca de subsequência** e **busca de sequência inteira**.

A **busca de subsequência** encontra a sequência em C que contém subsequências que são similares a uma dada consulta de sequência x. A **busca de sequência inteira** encontra o conjunto de sequências em C que são similares umas as outras (como um todo).

Para buscas de subsequências, cada sequência pode ser quebrada em um conjunto de peças, ou janelas, com o tamanho w. Em uma abordagem, as características da subsequência dentro de cada janela são então extraídas. Cada sequência é mapeada para uma "trilha" no espaço de conjuntos. A trilha de cada sequência é dividida em "sub trilhas", cada um representando por um retângulo de junção mínimo. Um algoritmo de montagem das múltiplas peças pode então ser utilizado para procurar por sequências mais longas (Kamber & Han, 2006).

2.3.3 Redução de dados

Devido ao grande tamanho e a alta dimensionalidade de muitas séries temporais, a redução de dados frequentemente serve como um primeiro passo na análise de séries temporais. A redução de dados leva não somente a um espaço de armazenamento muito menor, mas também a um processamento dos dados muito mais rápido.

Entre as estratégias para redução de dados, é possível encontrar:

- **Seleção de um subconjunto de atributos:** remove atributos, ou dimensões, redundantes ou irrelevantes;
- **Redução dimensional:** emprega geralmente técnicas de processamento de sinal para obter versões reduzidas do dado original;
- **Redução numérica:** os dados são substituídos, ou estimados, por menores representações alternativas, tais como histogramas, clusterização e amostragem.

Como uma série temporal pode ser vista como uma sequência de dados de alta dimensionalidade, onde cada ponto de tempo pode ser visto como uma dimensão, a redução dimensional pode ser considerada o maior foco. Por exemplo, para computar a relação entre duas curvas de série temporal, a redução da série temporal do comprimento (ou dimensão) n para k pode levar a redução de $O(n)$ para $O(k)$ em

termos de complexidade computacional. Se $k \ll n$, a complexidade da computação será largamente reduzida.

Várias técnicas de redução dimensional podem ser utilizadas na análise de séries temporais, tais como:

- *Discrete Fourier transform* (DFT) (Faloutsos, Ranganathan, & Manolopoulos, 1994);
- *Discrete wavelet transform* (DWT) (Chan & Fu, 1999);
- *Piecewise Aggregate Approximation* (PAA) (Chakrabarti, Keogh, Mehrotra, & Pazzani, 2002);
- *Adaptative piecewise constant approximation* (APCA) (Geurts, 2001);
- *Singular value decomposition* (SVD) (Chakrabarti, Keogh, Mehrotra, & Pazzani, 2002), baseada na técnica *Principle Component Analysis* (PCA).

É possível visualizar o efeito da redução dimensional de algumas dessas técnicas na Figura 2.

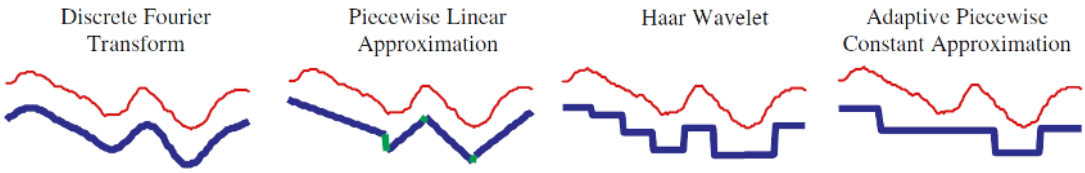


Figura 2: Redução dimensional (Lin, Keogh, Wei, & Lonardi, 2007)

Métodos como DFT, DWT e SVD são técnicas baseadas em processamento de sinal. Uma dada série temporal pode ser considerada como uma sequencia finita de valores reais (ou coeficientes), gravados sobre o tempo em um determinado espaço. O dado, ou sinal, é transformado (utilizando-se uma função específica de transformação) em um sinal dentro de um espaço transformado. Um pequeno subconjunto dos coeficientes transformados, considerados “mais fortes”, é salvo como um conjunto de características. Essas características formam um espaço de características, que são simplesmente a projeção do espaço transformado.

Essa representação é esparsa, de tal forma que operações que possam tomar vantagem sobre dados esparsos são computacionalmente muito mais rápidas se realizados nesse espaço de características do que no espaço original dos dados da série temporal.

Entretanto, para longas séries temporais, a vantagem dessa representação ainda pode ser computacionalmente ineficiente, devido a grande quantidade de dados sendo representados.

Isso pode exigir técnicas mais avançadas de redução dimensional e numérica.

2.4 O Método SAX

Muitas representações de alto nível para séries temporais têm sido propostas para a mineração de dados. A Figura 3 representa um apanhado hierárquico de várias representações de séries temporais na literatura atual.

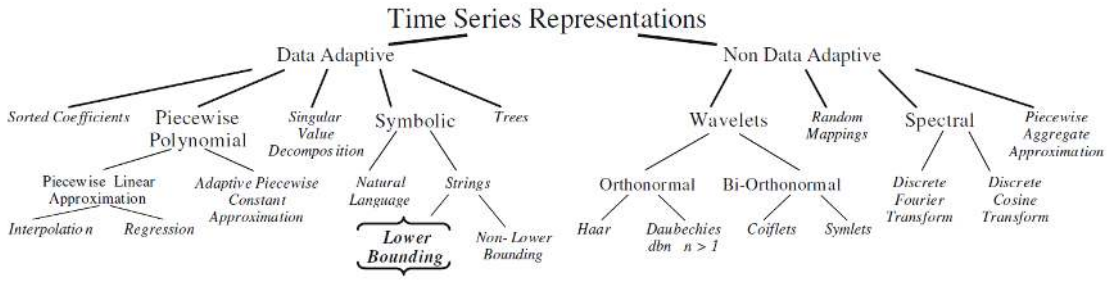


Figura 3: Representações de séries temporais (Lin, Keogh, Lonardi, & Chiu, 2003)

Um importante aspecto de grande parte dessas representações é que elas são valoradas em $\mathbb{R}$. Isso limita os algoritmos, estrutura de dados e definições disponíveis para elas. Em detecção de anomalias, por exemplo, não é possível definir de forma significativa a probabilidade de se observar um conjunto de coeficientes de onda, pois a probabilidade de observar qualquer número real é zero (Larsen & Marx, 1986).

Uma representação ainda pouco considerada em detalhes é a discretização dos dados originais em um conjunto de caracteres (*strings*) simbólicos. Como existe uma grande quantidade de algoritmos e estruturas de dados que permitem a manipulação de sequências de caracteres, essa representação se torna viável.

O método chamado ***S**ymbolic **A**ggregate Appro**x**imation* (SAX), proposto por (Lin, Keogh, Wei, & Lonardi, 2007) permite realizar uma representação simbólica de uma série temporal que realiza, ao mesmo tempo, a redução numérica e dimensional, além de permitir medidas de distância entre as séries temporais representadas por esse método como se fossem feitas pelas séries temporais representadas diretamente a partir dos dados originais.

O método SAX permite que uma série temporal de tamanho n seja reduzida para uma sequência de caracteres de tamanho w, onde $w \leq n$ (geralmente, $w \ll n$). O alfabeto utilizado para a sequência de caracteres é de tamanho a, onde $a \geq 2$. No método SAX, para cada um dos valores obtidos nas séries temporais do conjunto de dados, é realizada uma transformação que permite a sua representação simbólica.

A primeira transformação da série temporal, nesse método, é a representação dos dados através da técnica PAA (Chakrabarti, Keogh, Mehrotra, & Pazzani, 2002). A propriedade do SAX que estabelece a redução dimensional e a equivalência da distância entre a série temporal simbólica e a série temporal original, é advinda dessa técnica. Essas propriedades são detalhadas em (Chakrabarti, Keogh, Mehrotra, & Pazzani, 2002), (Keogh, Chakrabarti, Pazzani, & Mehrotra, 2000) e (Yi & Faloutsos, 2000).

Dessa forma, realiza-se a transformação de uma série temporal C em uma série temporal na representação PAA, exibida por $\hat{\mathrm{C}}$. A Figura 4 mostra essa transformação.

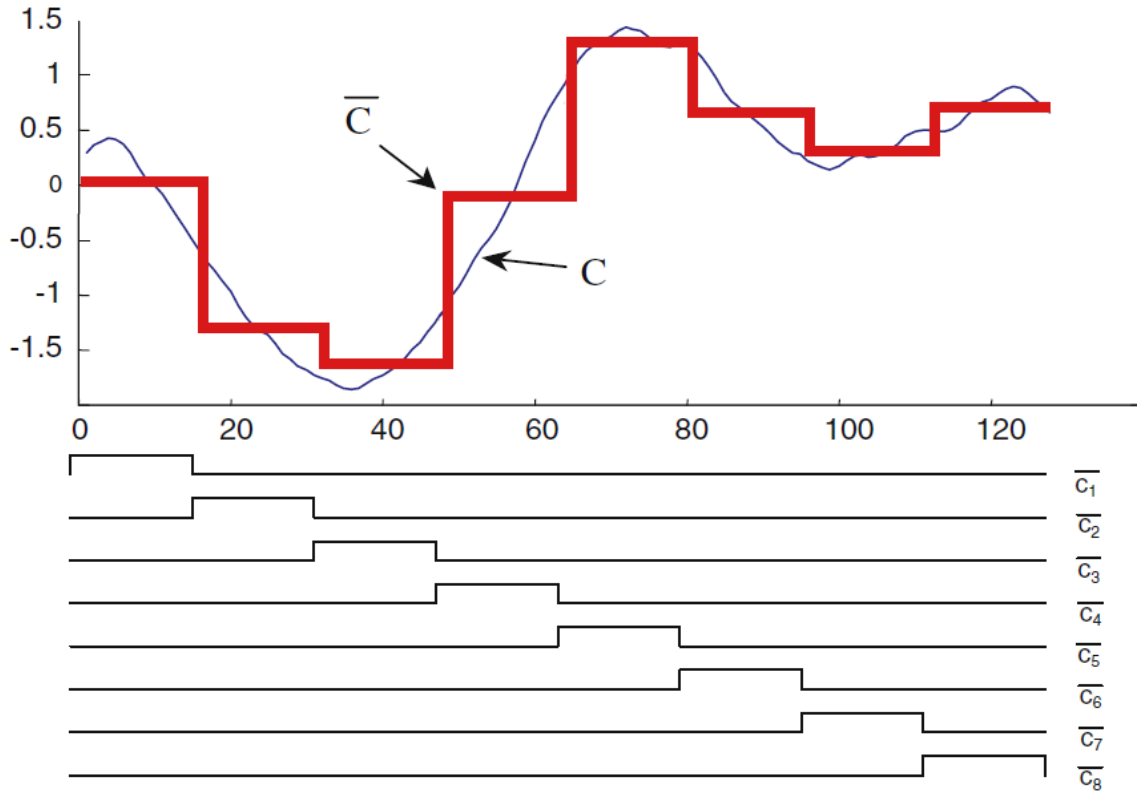


Figura 4: Série temporal na representação PAA

É possível observar nessa representação transformada, uma redução dimensional de 128 para oito. Como citado anteriormente, a tratabilidade computacional da série temporal representada por $\hat{C}$ é muito superior a da que poderia ser aplicada diretamente na série temporal C.

Após a redução dimensional da série temporal, é realizada a etapa de representação simbólica da série temporal. É desejável que a técnica de discretização para a representação simbólica produza símbolos com equiprobabilidade (Lonardi, 2001). As séries temporais normalizadas tem uma alta distribuição Gaussiana, onde os valores intermediários são muito mais prováveis que os valores extremos, ou seja, existe uma concentração muito grande de elementos de valores próximos. Dessa forma, é possível apenas definir as faixas de valores para as quais serão produzidas áreas com igual quantidade de elementos simbólicos.

2.4.1 Representação Simbólica de Múltiplas Séries Temporais

Quando diversas séries temporais são consideradas simultaneamente e deseja-se realizar uma correlação entre elas, o processo de redução dimensional se mantem para cada série temporal do conjunto de dados, mas o processo de representação simbólica deve ser modificado. Nessa situação, todas as séries temporais do conjunto de dados devem ser normalizadas, de forma a possibilitar uma correta correlação entre elas no momento de realizar, por exemplo, uma medição de distância entre duas ou mais séries temporais do conjunto considerado.

A faixa de valores, correspondente a cada símbolo, é determinada pelo valor dos elementos das diversas séries temporais consideradas e pela quantidade de faixas que são desejadas na representação. Nessa abordagem, todos os valores $(v_1, v_2, v_3, \ldots, v_n)$

de todas as séries temporais consideradas são primeiramente ordenados em ordem crescente de valor. Esse conjunto de valores é então dividido igualmente em f faixas $(f_1, f_2, f_3, \ldots, f_n)$, sendo possível obter um número de $\frac{n}{f}$ elementos em cada faixa. Dessa forma, é possível estabelecer que o tamanho do alfabeto é $a = f$.

Considerando que a primeira faixa f_1 será representada pelo caractere **a**, e que o elemento de valor v_1 é o menor entre todos os valores dos elementos normalizados das séries temporais, esse elemento deixará de ser representado pelo valor v_1 e será representado pelo caractere **a**.

A Figura 5 exemplifica a aplicação do método SAX, onde duas séries temporais, C_1 e C_2, são primeiramente reduzidas dimensionalmente, através do método PAA e logo depois recebem uma representação simbólica dos seus elementos.

É importante observar que nesse exemplo, os dados das duas séries temporais são considerados em conjunto, e não isoladamente, como seria o caso onde houvesse somente uma série temporal considerada. Isso permite uma normalização dos dados, e, consequentemente, facilita a análise das diversas séries temporais de forma correlacionada.

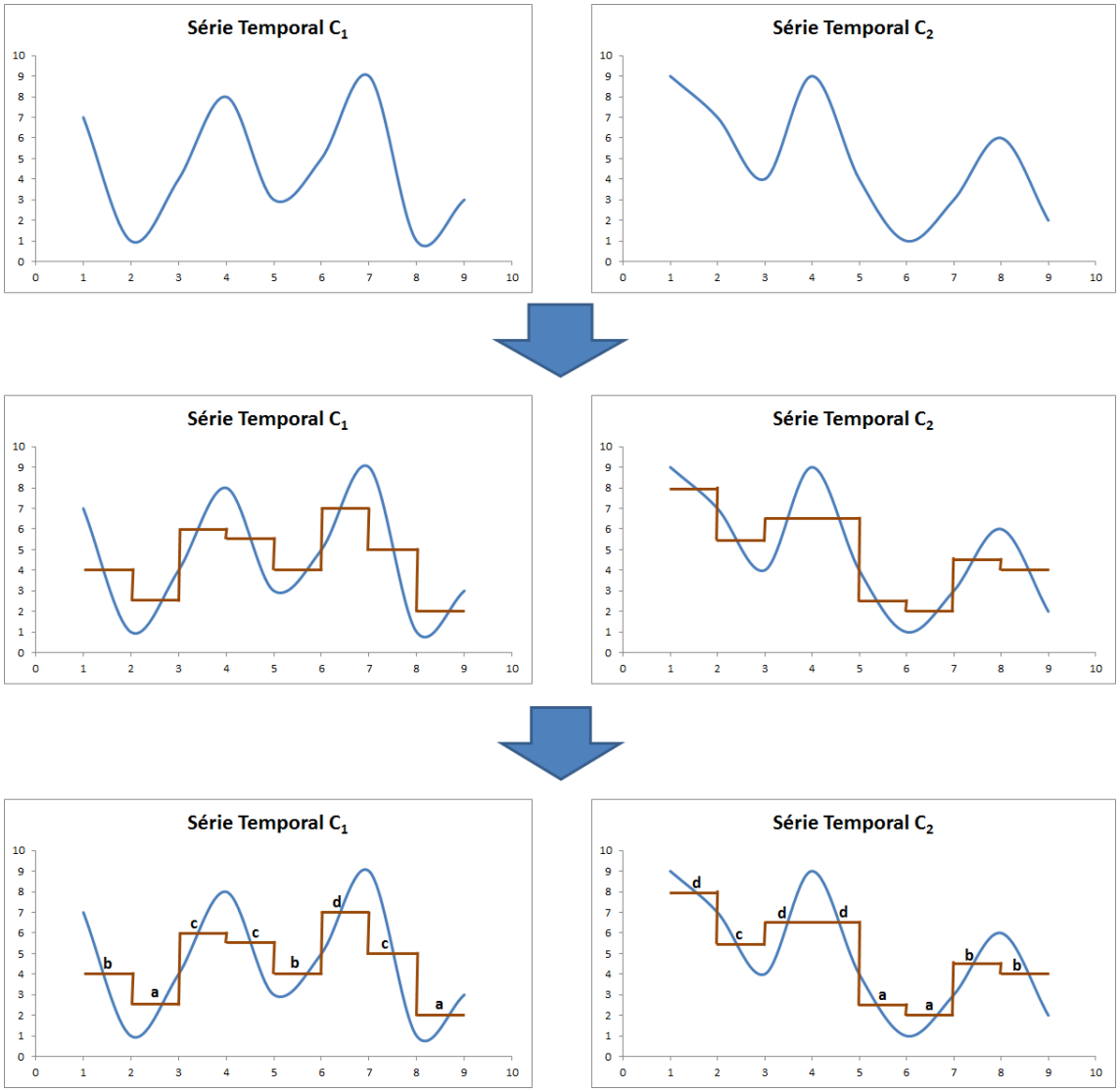


Figura 5: Método SAX

Como se pode ver na tabela 1, os valores das duas séries temporais são considerados de forma conjunta para estabelecer a faixa de valores para a qual cada símbolo será atribuído. Cada símbolo de representação é distribuído, em sua faixa correspondente, de forma equiprovável entre cada um dos n elementos das duas séries temporais.

Série Temporal	Valor do Elemento	Faixa	Símbolo
C_1	2	f_1	**a**
C_2	2	f_1	**a**
C_1	2,5	f_1	**a**
C_2	2,5	f_1	**a**
C_1	4	f_2	**b**
C_1	4	f_2	**b**
C_2	4	f_2	**b**
C_2	4,5	f_2	**b**
C_1	5	f_3	**c**
C_1	5,5	f_3	**c**
C_2	5,5	f_3	**c**
C_1	6	f_3	**c**
C_2	6,5	f_4	**d**
C_2	6,5	f_4	**d**
C_1	7	f_4	**d**
C_2	8	f_4	**d**

Tabela 1: Obtenção dos símbolos SAX

A última sequência de imagens das séries temporais da Figura 5 mostra o resultado do processo de representação simbólica realizado pelo método SAX. É possível observar que cada uma das séries temporais, C_1 e C_2, foi transformada em uma sequência de caracteres, **baccbdca** e **dcddaabb**, respectivamente.

A transformação de séries temporais contínuas em sequências de caracteres facilita a tratabilidade computacional e reduz de forma significativa o espaço de armazenamento de uma série temporal. Com essa representação, é possível utilizar todas as estruturas de dados e algoritmos atualmente disponíveis em áreas como, por exemplo, bioinformática e mineração de textos, facilitando a descoberta de soluções para os diversos desafios associados com a mineração de dados existentes atualmente.

2.5 Matrizes da Série Temporal

Entre os métodos atualmente disponíveis para mineração de séries temporais que possuem capacidade de tratamento de sequências de caracteres, encontra-se o método que transforma séries temporais em mapas de bits, chamado *Time-Series Bitmaps* (TSB).

Esse método foi proposto por (Kumar, Lolla, Keogh, Lonardi, Ratanamahatana, & Wei, 2005) e permite a manipulação de séries temporais de forma que:

- Não seja exigida do usuário uma grande quantidade de parâmetros de entrada, os quais geralmente são difíceis de serem antevistos pelo usuário;
- Permita fácil implementação;
- Possibilite a visualização dos dados das séries temporais, caso necessário, de forma amigável para o usuário.

A ideia central do método é transformar sequências de caracteres, as quais podem ser, por exemplo, desde uma cadeia de DNA até uma representação discretizada de uma série temporal, para um mapa de bits.

Vamos tomar como exemplo o processo de discretização realizado para as duas séries temporais mostradas na Figura 5. O resultado da aplicação do método SAX foi a obtenção das *strings* **baccbdca** e **dcddaabb** para as séries temporais C_1 e C_2, respectivamente.

A quantidade de faixas utilizada no processo de discretização nos fornece o primeiro parâmetro de entrada para o método TSB, que é a dimensão da matriz, representado por ∂. No caso das séries temporais C_1 e C_2, $f = 4$. Dessa forma, $\partial = 4$.

Outro parâmetro importante para o método TSB é a resolução, aqui representado por ρ. A resolução permite estabelecer a profundidade desejada de análise da série temporal, já que para uma resolução $\rho \geq 2$, existe o agrupamento de letras da sequência de caracteres, possibilitando, por exemplo, uma detecção de um padrão de comportamento local na trajetória.

A Figura 6 mostra três mapas de bits, de dimensão $\partial = 4$, em três resoluções diferentes. A representação dos símbolos foi feita pelo alfabeto, de forma sequencial, a partir da letra **a**.

$\rho = 1$	
a	b
c	d

$\rho = 2$			
aa	ab	ba	bb
ac	ad	bc	bd
ca	cb	da	db
cc	cd	dc	dd

$\rho = 3$							
aaa	aab	aba	abb	baa	bab	bba	bbb
aac	aad	abc	abd	bac	bad	bbc	bbd
aca	acb	ada	adb	bca	bcb	bda	bdb
acc	acd	adc	add	bcc	bcd	bdc	bdd
caa	cab	cba	cbb	daa	dab	dba	dbb
cac	cad	cbc	cbd	dac	dad	dbc	dbd
cca	ccb	cda	cdb	dca	dcb	dda	ddb
ccc	ccd	cdc	cdd	dcc	dcd	ddc	ddd

Figura 6: Matriz de série temporal

É possível observar que o número de células da matriz do mapa de bits é exponencialmente proporcional à resolução desejada. O número de células φ é dado pela seguinte fórmula:

$$\varphi = \partial^{\rho}$$

Em cada uma das φ células, deve ser realizada a contagem de quantas ocorrências do(s) caractere(s) correspondente(s) da célula ocorreu na *string* resultante da série temporal (essa é a razão que $f = \partial$).

Realizando o mapeamento de cada uma das letras da sequência de *strings* resultantes da discretização das séries temporais C_1 e C_2 para o método TSB, é possível visualizar, na Figura 7, o mapa de bits resultante para essas duas séries temporais.

Série Temporal C_1

$\rho = 1$	
2	2
3	1

$\rho = 2$			
0	0	1	0
1	0	0	1
1	1	0	0
1	0	1	0

$\rho = 3$							
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0

Série Temporal C_2

$\rho = 1$	
2	2
1	3

$\rho = 2$			
1	1	0	1
0	0	0	0
0	0	1	0
0	1	1	1

$\rho = 3$							
0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0

Figura 7: Matrizes de C_1 e C_2

Como citado anteriormente, uma das vantagens do método é a possibilidade de análise visual das matrizes geradas.

Usando somente uma escala de cinza, é possível representar as matrizes apresentadas na Figura 7 em uma forma mais visual, como mostrado na Figura 8.

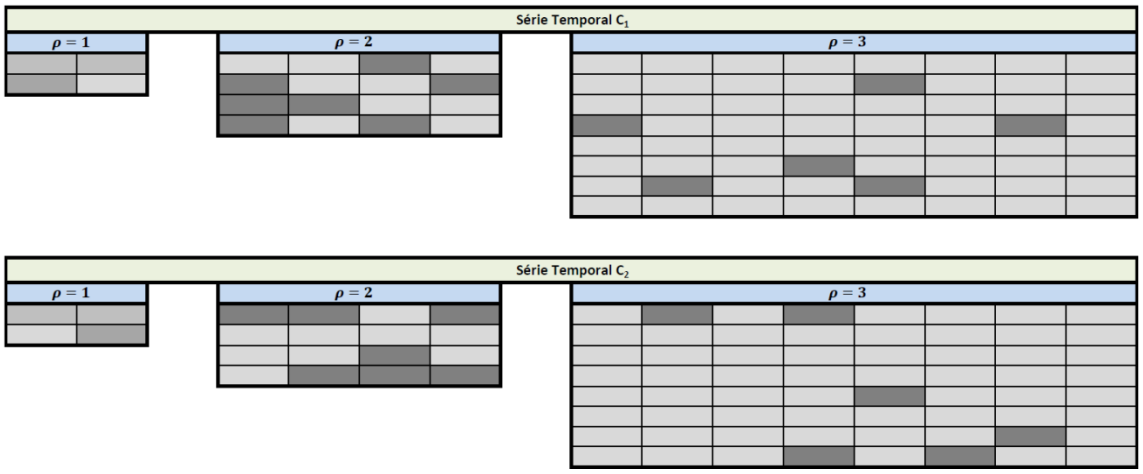


Figura 8: Apresentação visual das matrizes de C_1 e C_2

Como no exemplo de C_1 e C_2 a quantidade elementos das séries temporais é bastante reduzida, pode não ficar muito claro o potencial da possibilidade de visualização das matrizes que são geradas pelo método TSB.

Para exemplificar uma utilização prática dessa possibilidade, a Figura 9 mostra as matrizes resultantes do sequenciamento genético de quatro animais.

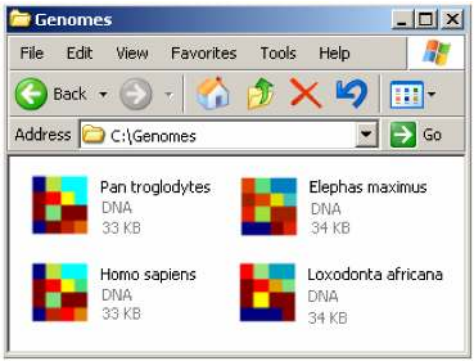


Figura 9: Apresentação visual de matrizes de genomas (Kumar, Lolla, Keogh, Lonardi, Ratanamahatana, & Wei, 2005)

Nesse exemplo da Figura 9 fica bem mais claro o potencial de visualização resultante da aplicação do método TSB, já que os dois primatas (*Pan Troglodytes* e *Homo Sapiens*) tem o mapa de bits de seus genomas com representações de cores muito próximas, assim como a representação para os dois paquidermes (*Elephas Maximus* e *Loxidonta Africana*).

Apesar da excelente possibilidade de classificação visual das séries temporais consideradas, pode ser desejável o reconhecimento automático dos padrões expostos através do uso do método TSB de forma massiva, especialmente quando se lida com uma grande quantidade de dados.

Para isso, podemos utilizar alguns dos diversos algoritmos de mineração de dados que existem atualmente. Muitos deles estão disponíveis em sistemas que permitem a

entrada do conjunto de dados a ser analisado através da utilização de diversos algoritmos diferentes para descoberta de conhecimento a partir desses dados.

2.6 Ferramenta de Análise Weka

Entre as ferramentas que permitem realizar o processo de descoberta de conhecimento em grandes conjuntos de dados, é possível citar a ferramenta *Waikato Environment Knowledge Analysis* ou mais comumente conhecida como Weka (Frank, Hall, Holmes, Kirkby, & Pfahringer, 2005).

A ferramenta Weka, bastante difundida no meio acadêmico, reúne um conjunto considerável de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Ela contém ferramentas para o pré-processamento dos dados, classificação, regressão, clusterização, regras de associação, bem como ferramentas de visualização dos resultados obtidos.

2.6.1 Pré-Processamento de Dados

A ferramenta Weka dispõe de uma série de filtros para realizar o pré-processamento dos dados de entrada. Os filtros são separados em categorias de filtros que utilizam técnicas de processamento dos dados de forma supervisionada e não supervisionada. Dentro de cada uma dessas categorias ainda existem os filtros que manipulam somente os atributos de cada uma das amostras do conjunto de dados (como por exemplo o filtro que discretiza os valores contínuos) ou que manipulam o conjunto de dados como um todo (como o filtro que gera uma sub-amostra randômica do conjunto de dados original).

Entre os diversos filtros disponíveis, podemos destacar o filtro de discretização. Na ferramenta Weka, esse filtro possui duas versões: um supervisionado e outro não supervisionado. O filtro de discretização supervisionado atua nos atributos do conjunto de dados, realizando uma transformação discreta no intervalo de atributos numéricos do conjunto de dados para atributos nominais. O método de discretização supervisionado tem como base o método explicado em (Fayyad & Irani, 1993).

Esse filtro mostra-se bastante útil para amostras de dados que contém, em sua grande parte, valores contínuos, já que a partir da aplicação desse filtro os algoritmos de mineração que utilizam alguns tipos de técnicas como, por exemplo, árvores binárias, tendem a ter mais facilidade para construir um modelo de classificação com maior acurácia.

Além do filtro de discretização, existem diversos outros tipos de filtros de pré-processamento, tais como:

- Re-ordenador de classe;
- Re-amostragem do conjunto de dados;
- Particionadores;

2.6.2 Mineração de Dados

O processo de mineração de dados na ferramenta Weka possibilita o uso de técnicas de classificação, associação e clusterização.

Em qualquer uma dessas técnicas, deve-se informar um conjunto de dados de entrada, o que pode ser ou um arquivo em formato texto ou uma tabela de um banco de

dados, que deverá conter a amostra de dados de um determinado domínio. Todos os exemplos dessa amostra devem conter atributos de interesse para a descoberta de conhecimento requerida. Em muitas situações, é possível, mas não obrigatório, definir um atributo classe que define o registro da amostra.

Estes dados são os mesmos que serviram para a fase de pré-processamento, e eventualmente passaram por um processo de filtragem nessa etapa. Agora estarão sendo submetidos à etapa de mineração de dados. Ambas as etapas estão contempladas dentro do processo de KDD tradicional.

```
@RELATION fraude_eletrica

@ATTRIBUTE idade                INTEGER
@ATTRIBUTE salario              REAL
@ATTRIBUTE situacao_civil       {CASADO,SOLTEI,DIVORC}
@ATTRIBUTE grau_instrucao       INTEGER
@ATTRIBUTE valor_consumo        REAL
@ATTRIBUTE situacao_instalacao  {Normal,Fraude}

@DATA
52,4312.43,'CASADO',1,120,Normal
38,2578.54,'SOLTEI',3,140,Normal
48,1635.96,'CASADO',1,290,Normal
47,3352.24,'CASADO',1,640,Normal
31,6575.54,'CASADO',3,310,Normal
55,3238.74,'CASADO',2,335,Normal
19,4685.83,'SOLTEI',3,050,Fraude
52,1811.04,'CASADO',3,340,Normal
31,4499.44,'DIVORC',3,440,Normal
44,4733.11,'CASADO',3,275,Normal
22,2122.34,'SOLTEI',0,260,Normal
48,4366.99,'SOLTEI',3,128,Normal
45,1948.34,'SOLTEI',2,140,Fraude
49,4928.64,'CASADO',3,360,Normal
59,2333.34,'CASADO',1,320,Normal
53,4646.45,'DIVORC',1,220,Normal
23,1344.30,'SOLTEI',0,270,Fraude
43,4894.43,'CASADO',1,090,Normal
46,4512.43,'DIVORC',2,100,Normal
31,2778.54,'SOLTEI',2,140,Normal
49,1535.96,'CASADO',1,300,Normal
47,3752.24,'CASADO',1,640,Normal
43,6475.54,'CASADO',2,300,Normal
52,3638.74,'DIVORC',3,300,Normal
21,4685.83,'SOLTEI',3,050,Fraude
```

Figura 10: Arquivo de entrada de dados Weka

A Figura 10 mostra dados no formato de um arquivo "arff" (***a**tribute-**r**elation **f**ile **f**ormat*), que exibe um exemplo de parte de um arquivo de entrada de dados para a ferramenta Weka. Esse exemplo refere-se a uma suposta mineração de dados de fraudes em instalações de energia elétrica. As tags do tipo *@ATTRIBUTE* nomeadas como "idade", "salário", "situação_civil", "grau_instrução", "valor_consumo" e "situacao_instalacao" fazem parte do conjunto de atributos que posteriormente serão utilizados para a etapa de mineração (se todos os atributos forem selecionados na etapa de pré-processamento). O atributo "situacao_instalacao" indica o resultado que será utilizado para validar o modelo gerado através de técnicas de validação, como a técnica de *cross-validation.*

Após a seleção do arquivo de entrada na ferramenta, pode-se então pré-processar os dados, selecionando atributos interessantes ao domínio e removendo outros atributos que não serão inclusos no processo de mineração. Após a seleção dos dados, a etapa de mineração pode ser iniciada selecionando-se um dos diversos algoritmos disponíveis na ferramenta, que são categorizados pelos processos de clusterização, associação e classificação de dados, sendo esta última a que será utilizada para o processo de mineração do arquivo mostrado parcialmente na Figura 10.

Após escolher um algoritmo de classificação e executar o processo de mineração, a ferramenta Weka dispõe de uma série de informações (métricas) que são utilizadas para a avaliação da qualidade do modelo obtido a partir dos dados minerados, como mostrado na Figura 11.

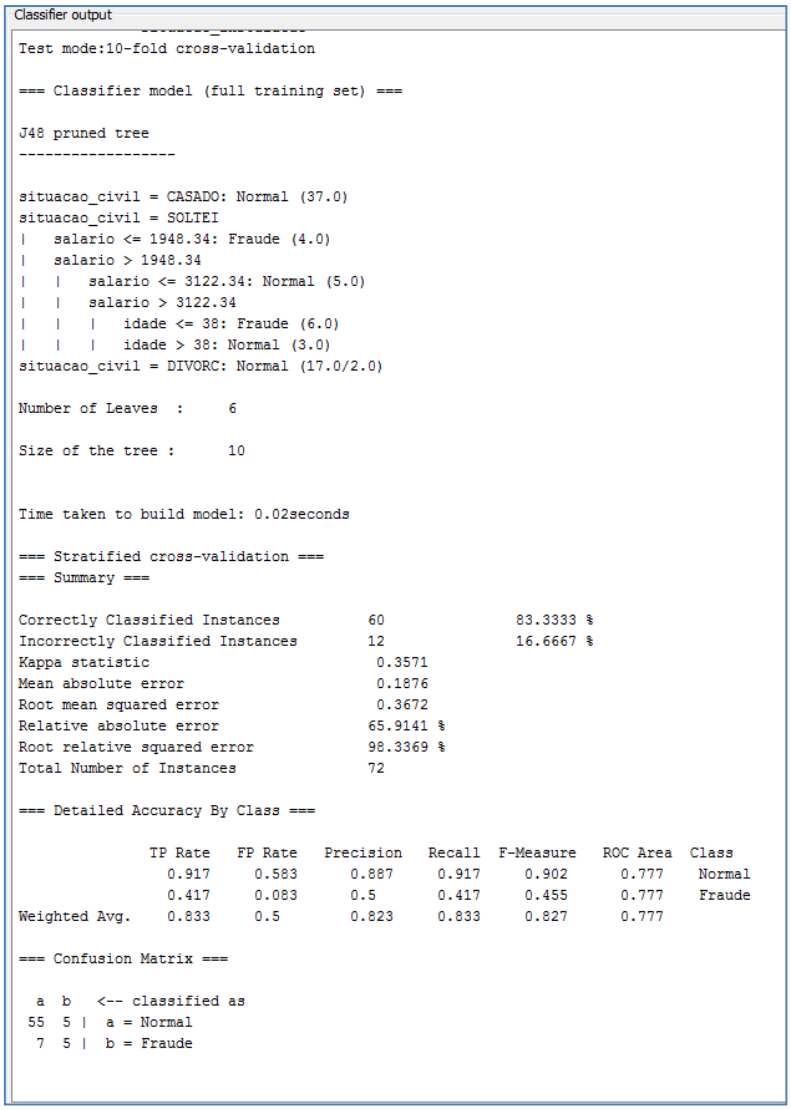


Figura 11: Resultado da etapa de mineração de dados

A Figura 11 mostra uma série de informações decorrentes do processo de mineração de dados. A primeira é o modelo de classificação gerado (*Classifier Model*), que nesse caso foi gerado por um algoritmo de classificação chamado *J48* (Quinlan, 1993) o qual produz um modelo de classificação em formato de árvore (*J48 prunned tree*). O tempo necessário para construir o modelo também é exibido, nesse caso 0,02 segundos. O formato e valor das informações referentes ao modelo de classificação gerado podem variar bastante, dependendo do tipo de algoritmo de classificação utilizado.

As informações exibidas abaixo do modelo de classificação são referentes as estatísticas de avaliação do modelo gerado. A primeira parte dessa informação é como foi realizado o processo de avaliação, que nesse caso foi realizado com o método de validação cruzada (*cross-validation*). Nesse método, o conjunto de dados original é dividido em n sub-conjuntos, dos quais um é utilizado para a validação do modelo gerado a partir dos $(n-1)$ sub-conjuntos restantes. O processo então é repetido n vezes, variando cada um dos restantes sub-conjuntos como o conjunto de teste e treinamento. O resultado do validação cruzada n-vezes pode então ser combinada para produzir um resultado geral da avaliação.

Logo após, é possível visualizar um sumário com as estatísticas de aplicação do modelo gerado no conjunto de dados de testes. Podemos destacar os seguintes campos e seus respectivos significados:

- *Correctly Classified Instances*: a acurácia do modelo, ou seja, quantas instâncias foram corretamente preditas a partir dos atributos de entrada.
- *Kappa statistic*: É a medida de concordância entre as classificações realizadas e a classe real. Um valor maior que zero significa que o classificador está realizando o processo de classificação melhor do que se houvesse uma escolha aleatória da classe da instância, enquanto um valor mais próximo de um estabelece um maior nível de acurácia do modelo gerado.
- *Mean absolute error*: Mede a magnitude média dos erros em um conjunto de previsões entre o que foi classificado e a classe real.
- *Root mean squared error*: O valor é obtido pela média do quadrado da diferença entre a classificação realizada e o valor real sobre o conjunto de dados.
- *Relative absolute error*: É o *mean absolute error* dividido pelo erro correspondente a aplicação do classificador ZeroR no conjunto de dados (ou seja, o classificador que prediz as probabilidades a priori das classes observadas no conjunto de dados).
- *Root relative squared error*: Faz o mesmo que *relative absolute error*, modificando a estatística de erro para *root mean squared error*.

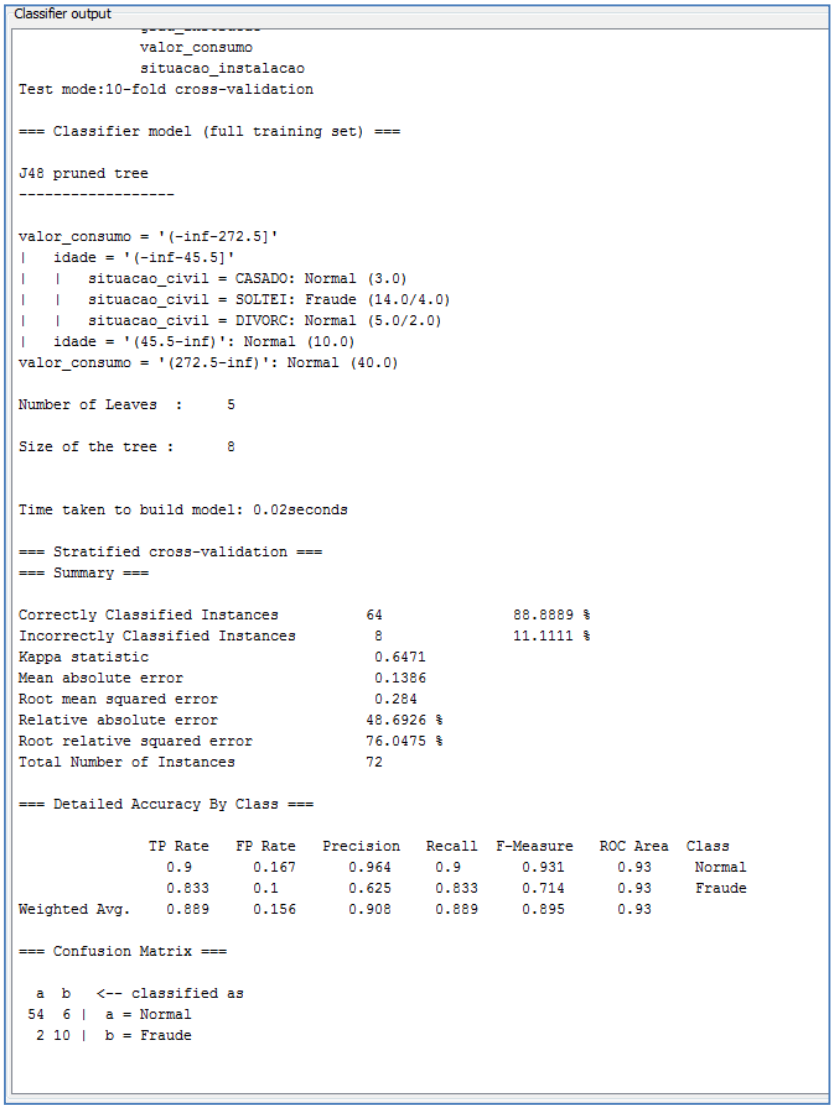
Os demais dados são todos relativos a matriz de confusão (*confusion matrix*), ou seja, a relação entre a classe predita e a classe real. As classes na horizontal representam as classes preditas, enquanto as classes na vertical representam as classes reais. As estatísticas derivadas da matriz de confusão são:

- *TP Rate*: taxa de verdadeiro positivo, ou seja, o número de instâncias corretamente classificadas como positivas pelo número total de instâncias daquela classe.
- *FP Rate*: taxa de falso positivo, ou o número de instâncias classificadas erroneamente como positivas pelo total de instâncias positivas.
- *Precision:* É a proporção de elementos que realmente possuem uma determinada classe dividida pela quantidade de elementos que foram classificados nessa mesma classe.
- *Recall*: É equivalente a *TP Rate*.
- *F-Measure*: É uma medida combinada entre precision e recall, calculado por 2**Precision***Recall*/(*Precision*+*Recall*).
- *ROC Area*: Representa a taxa de discriminação do modelo gerado. É desejável um valor maior que 0,5, já que esse é um valor que representa um modelo tão bom quanto à escolha da classe ao acaso.
- *Class*: É a classe que está sendo analisada para as respectivas métricas citadas.
- *Confusion Matrix*: é a matriz de confusão propriamente dita.

Podemos constatar claramente nesta matriz quando o modelo gerado classifica alguma instância de forma incorreta: na primeira linha da matriz, das 60 instâncias classificadas como instalação elétrica normal, cinco instâncias foram classificadas

incorretamente como instalações elétricas fraudulentas. E das 12 instâncias que eram fraudulentas, o modelo conseguiu classificar corretamente apenas cinco.

O processo KDD prevê, em casos como esse, onde o modelo não atinja os resultados esperados, voltar para alguma etapa anterior, como por exemplo, a de pré-processamento, e manipular o conjunto de dados. A Figura 12 mostra os resultados obtidos no mesmo conjunto de dados, com o mesmo algoritmo de mineração utilizado, mas com o filtro de discretização supervisionado aplicado antes da mineração dos dados.



Classifier output

```
              valor_consumo
              situacao_instalacao
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

valor_consumo = '(-inf-272.5]'
|   idade = '(-inf-45.5]'
|   |   situacao_civil = CASADO: Normal (3.0)
|   |   situacao_civil = SOLTEI: Fraude (14.0/4.0)
|   |   situacao_civil = DIVORC: Normal (5.0/2.0)
|   idade = '(45.5-inf)': Normal (10.0)
valor_consumo = '(272.5-inf)': Normal (40.0)

Number of Leaves  :     5

Size of the tree :      8


Time taken to build model: 0.02seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          64               88.8889 %
Incorrectly Classified Instances         8               11.1111 %
Kappa statistic                          0.6471
Mean absolute error                      0.1386
Root mean squared error                  0.284
Relative absolute error                 48.6926 %
Root relative squared error             76.0475 %
Total Number of Instances               72

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.9       0.167      0.964     0.9       0.931      0.93     Normal
                 0.833     0.1        0.625     0.833     0.714      0.93     Fraude
Weighted Avg.    0.889     0.156      0.908     0.889     0.895      0.93

=== Confusion Matrix ===

  a  b   <-- classified as
 54  6 |  a = Normal
  2 10 |  b = Fraude
```

Figura 12: Resultado da etapa de mineração de dados após discretização

A Figura 12 demostra o potencial de realizar a manipulação no conjunto de dados na etapa de pré-processamento. Após aplicar o filtro de discretização no conjunto de dados de entrada e submetê-lo ao processo de mineração, os resultados foram mais proveitosos para o objetivo da mineração nesse domínio de dados. Além do percentual de acerto geral ter melhorado, o maior ganho foi no aumento significativo da correta classificação de instalações fraudulentas como tal, expressado pela diminuição expressiva de *FP Rate* para instalações consideradas como “Normais”.

2.7 Trabalhos Relacionados

Para a classificação de dados convencionais, como no exemplo clássico da planta Iris, o conjunto de dados contém atributos descritivos da planta como o comprimento e largura da pétala e da sépala. Cada registro corresponde a um exemplo, com os dados de uma planta específica: os valores de suas características e a sua classe. Já para realizar a classificação de trajetórias, os dados fornecidos se resumem a uma sequência (id, x, y, z, t), onde cada identificador da trajetória id está associado a uma classe, o que dificulta bastante o processo.

Entre a base de dados de trajetórias e a construção do modelo de classificação final, algum tipo de processamento dos dados da trajetória deve ser realizado. Poucos trabalhos têm sido desenvolvidos visando à classificação de trajetórias. A grande maioria deles resume-se à extração de características relevantes para o domínio considerado, especializando-se em resolver um problema específico.

O trabalho proposto por (Panagiotakis, Pelekis, & Kopanakis, 2009) introduz um método de classificação de trajetórias baseado em uma função de votação definida pela distância entre trechos dessas trajetórias. A função de votação é baseada na similaridade entre esses trechos da trajetória, possibilitando a construção de um modelo de classificação a partir dessa métrica. Cada trajetória é segmentada em diversos trechos e a função de votação global é aplicada para cada um desses segmentos, formando um descritor local da trajetória. Através da análise desses descritores, é possível detectar trajetórias representativas, que são diferentes segmentos das trajetórias originais, de diversos objetos móveis, que percorreram tempo e espaço semelhantes, estabelecendo uma similaridade entre essas trajetórias. O método se mostra eficiente na detecção de padrões de congruência das trajetórias, possibilitando uma correlação eficiente entre elas, entretanto, caso seja necessário classificar trajetórias de classes que não tenham suas instâncias necessariamente relacionadas espacialmente, o método pode encontrar dificuldade em construir um modelo de classificação eficiente.

É também possível citar o trabalho desenvolvido por (Lee & Hoff, 2007) na descoberta da atividade esportiva desempenhada a partir da trajetória produzida pelos jogadores. A ideia desse método é dividir as trajetórias de uma mesma atividade desportiva em diversos segmentos de três segundos cada. A partir de cada um desses segmentos, são extraídos os valores de velocidade média e do comprimento do segmento. Esses valores são então separados por clusters em um eixo (x, y), sendo um dos eixos a velocidade média e o outro o comprimento do segmento. Após essa clusterização, cada segmento da trajetória é ligado em sequência, de acordo com os valores de suas características, entre os clusters, formando uma cadeia sequencial de clusters que possibilita a construção de um modelo de Markov. Um modelo de Markov é gerado para cada atividade. Com as trajetórias das atividades discretizadas, um modelo de classificação é gerado, possibilitando a classificação de trajetórias ainda não rotuladas, através do uso do mesmo processo de segmentação visto para a construção do modelo de classificação. Com o modelo de Markov gerado para essa trajetória desconhecida, é verificado qual modelo de Markov gerado anteriormente para as atividades maximiza a semelhança com o da trajetória desconhecida, permitindo o reconhecimento da atividade para a trajetória.

Apesar de o método ter obtido um bom resultado na identificação da atividade, ele acaba sendo muito dependente da correta parametrização da etapa de clusterização, pois uma vez que as características escolhidas para serem clusterizadas, ou o número de

clusters gerados não forem adequados, o modelo de classificação resultante pode ser ineficiente. Dessa forma o método depende da escolha das melhores características para o domínio considerado.

No trabalho de (Zheng, Liu, Wang, & Xie, 2008), o foco está baseado na descoberta do meio de transporte do usuário (carro, ônibus, moto, bicicleta, a pé, etc.) com base somente nas trajetórias que o usuário produziu durante um determinado período de tempo. A abordagem segmenta as trajetórias de um usuário utilizando a noção de parada ou de grande redução de velocidade ao mudar de meio de transporte. Para cada um desses segmentos, são extraídas características importantes tais como comprimento, velocidade média, covariância da velocidade, entre outras. A partir desses atributos é construído um modelo de classificação para a identificação do segmento de trajetória. Um modelo probabilístico é utilizado para corrigir eventuais discrepâncias de transição entre meios de transporte, como ônibus → carro → a pé → carro, situação essa que pode indicar um erro, já que é realizada a transição direta entre ônibus e carro. Nesses casos, é possível verificar a precisão de classificação do segmento com a tabela de probabilidade de transição e estabelecer uma ordem correta na transição entre os meios de transporte, ficando, por exemplo, como ônibus → a pé → carro.

O trabalho propõe um método que produz um bom resultado na classificação dos segmentos das trajetórias, identificando corretamente os meios de transporte. Entretanto, parte da acurácia do modelo é estabelecida pela possibilidade de sequência de transição entre os meios de transporte (segmentos de trajetória), o que acaba impossibilitando o uso desse mesmo método para conjuntos de dados que não tenham relação com esse tipo de domínio.

No trabalho de (García, Concha, Molina, & Miguel, 2006) é realizada a identificação do modo de voo de objetos móveis considerados no domínio de controle de tráfego aéreo. São utilizados três filtros de Kalman combinados para identificar manobras transversais, longitudinais e movimento uniformes dos objetos móveis. As classes dessas trajetórias acabam sendo um desses três tipos de modos de voo ou caem em uma quarta classe, que é uma manobra combinada, que contém mais de um modo de voo. Além de uma forte etapa de pré-processamento, o método também prevê uma etapa de pós-processamento para eliminar classificações realizadas em segmentos de trajetórias muito curtas, que não caracterizam corretamente um modo de voo, que faz sentido ser considerado somente para trajetos maiores. O método realiza com sucesso a classificação e predição off-line de modos de voo para trajetórias de tráfego de controle aéreo, mas os filtros de Kalman acabam sendo superespecializados na tarefa de identificar somente modos de voo de aviões, dificultando o seu possível aproveitamento na classificação de dados para outros domínios.

No recente trabalho publicado por (de Vries, van Hage, & van Someren, 2010) é apresentado um método que realiza a medição de similaridade entre trajetórias enriquecidas com conhecimento do domínio geográfico. Em uma abordagem semelhante à realizada por (Palma, Bogorny, Kuijpers, & Alvares, 2008), são aplicados rótulos geográficos em locais próximos ao percurso das trajetórias, proporcionando a introdução de semântica ao domínio de dados. Uma das diferenças entre os dois métodos é que no trabalho de (de Vries, van Hage, & van Someren, 2010) também são rotulados os “Moves” e não somente os “Stops”, como no trabalho de (Palma, Bogorny, Kuijpers, & Alvares, 2008). No primeiro trabalho também é utilizada uma ontologia para armazenar os locais (RDF), o que é feito através de uma base geográfica de locais no segundo. A primeira etapa desse método é a clusterização das trajetórias semelhantes

(foram utilizadas 1917 trajetórias de barcos em um raio de 50Km). Na segunda etapa é realizada a avaliação dos lugares pelos quais essas trajetórias semelhantes passam, através da avaliação dos locais geográficos, permitindo realizar a identificação do grupo de trajetórias. Com isso, é possível, através de uma tarefa de classificação de dados, identificar entre 18 tipos de barcos, qual realizou determinada trajetória. O percentual de acurácia de classificação das trajetórias ficou entre 52,8% e 75,4%, dependendo do parâmetro estipulado para o algoritmo.

A classificação de trajetórias em redes rodoviárias é o assunto do trabalho apresentado em (Han, Lee, Li, & Cheng, 2011). O método é fortemente baseado na análise de padrões sequenciais das trajetórias, que são transformados em um vetor de características utilizados para classificar a trajetória. Além dos locais pelos quais essas trajetórias passam, o método analisa também a ordem na qual os locais são visitados, utilizando dessa forma a componente espacial e temporal das trajetórias. A primeira etapa do algoritmo proposto é a geração do vetor de características através da mineração de padrões sequencias das trajetórias, que tenham como característica um alto poder discriminativo em seu conjunto, bem como um alto suporte. O comprimento das trajetórias a serem analisadas pelo minerador de padrões sequenciais é limitado através de uma parametrização feita a priori. Com isso, é obtido um conjunto inicial de padrões sequenciais, ou características, que são então selecionadas pelo critério de *F-Score*. O vetor das características selecionadas é então mapeado para cada uma das trajetórias, sendo então finalmente utilizadas para alimentar um classificador. O método tem um resultado muito bom para trajetórias sintéticas que tenham caminhos em comum (como pontes, rodovias e outros locais com convergência de trajetórias em rodovias), obtendo resultados com acurácia acima de 90%. Entretanto, quando alguns dados reais com pouca convergência são submetidos ao método, a acurácia acaba caindo sensivelmente, para em torno de 80%, expondo um dos pontos fracos do método: a análise de trajetórias pouco convergentes.

Na abordagem para classificação de trajetórias desenvolvida por (Lee, Han, Gonzalez, & Li, 2008) é desenvolvido um método que realiza a análise espacial de trajetórias em diferentes domínios de aplicação. Para isso, é realizado um processo que é dividido em duas grandes etapas: uma de clusterização e a outra de classificação. A primeira etapa, de clusterização, particiona e agrupa as trajetórias. O particionamento transforma as trajetórias originais em sub-trajetórias, gerando um conjunto de trajetórias com direção homogênea. Essas sub-trajetórias são então agrupadas (clusterizadas) em trajetórias representativas que definem o comportamento comum de um grupo maior de trajetórias. Na segunda etapa, de classificação, as sub-trajetórias particionadas e agrupadas na etapa anterior são utilizadas como entrada para o processo de classificação. Nessa etapa, são realizados duas tarefas de classificação: uma "*Region Based*" (baseado em regiões) e a outra chamada de "*Trajectory Based*" (baseado nas trajetórias). Na sub-etapa baseada em trajetória (TB) o objetivo é descobrir sub-trajetórias que indiquem um padrão de movimento comum para cada classe. Isso é feito ao se estender o método que particiona e agrupa as trajetórias na etapa de clusterização, incorporando rótulos de classe a essa tarefa. Já na sub-etapa baseada em regiões (RB), o objetivo é descobrir regiões que contenham a maioria das trajetórias de uma classe, independente do seu padrão de movimento, conforme é feito na etapa TB. Com isso, são gerados clusters altamente discriminativos que são utilizados para uma classificação eficiente. Assim, é possível converter cada trajetória em um vetor de características, onde cada característica é tanto um cluster TB ou RB. Essas características irão gerar os atributos de classificação. Os atributos de classificação gerados são então fornecidos

como entrada em um algoritmo de classificação (*Support Vector Machines*), e um modelo de dados é gerado para realizar a classificação de trajetórias ainda não rotuladas.

Como pode ser observado, a maioria dos trabalhos citados descreve métodos específicos para a aplicação que está sendo considerada no trabalho. O de (Lee, Han, Gonzalez, & Li, 2008) é que apresenta um método mais geral, permitindo uma grande flexibilidade na realização de classificação para diversos tipos de dados considerados. Entretanto, somente é utilizada a componente espacial da trajetória, sendo ignorada a sua componente temporal. O método *TRACTS*, proposto nessa dissertação, também é geral e considera a componente temporal para realizar a classificação das trajetórias.

3 MÉTODO *TRACTS*

3.1 Introdução

Diferentemente de como acontece na tarefa de classificação com bases de dados tradicionais, onde os dados já estão separados em atributos, a tarefa de classificação de trajetórias requer uma transformação completa dos dados antes que qualquer forma de classificação possa ser realizada. Isso se deve ao fato de que dificilmente um modelo de classificação tradicional possa ser construído diretamente a partir de uma base de dados espaço-temporal, que possui o formato (id, x, y, z, t), pelo menos não com os algoritmos de classificação tradicionais atualmente disponíveis.

Dessa forma, para possibilitar a classificação de um conjunto de dados espaço-temporal com os algoritmos clássicos atuais, devemos processar esses dados em diversas etapas de transformação, de modo que a sequência de pontos das várias trajetórias no formato (id, x, y, z, t) possa ser traduzido em uma sequência de atributos $(A_1, A_2, A_3, \ldots, A_n, A_c)$, onde A_n é o n-ésimo atributo e A_c é o atributo classe.

Para resolver esse problema, esse trabalho propõe o método *TRACTS* (***Tra****jectory* ***C****lassification using* ***T****ime* ***S****eries*). O método *TRACTS* contém quatro etapas principais, sendo algumas delas desdobradas em sub-etapas.

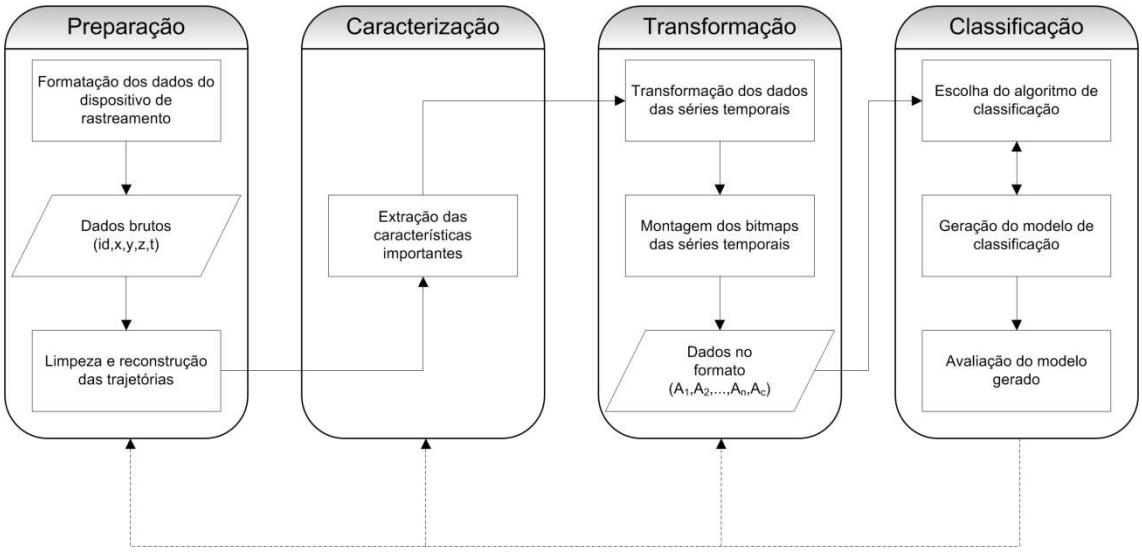


Figura 13: Método *TRACTS*

A primeira grande etapa consiste na preparação dos dados brutos obtidos a partir dos dispositivos de rastreamento de objetos móveis. Nessa etapa inicial, os dados são formatados após a sua obtenção, os ruídos são eliminados (limpeza) e os dados são adequados conforme a necessidade (reconstrução).

Na próxima etapa, características importantes são extraídas desse conjunto de dados, possibilitando identificar comportamentos específicos de cada uma das trajetórias. Essas características serão peças fundamentais para a formação dos atributos de classificação. A ideia proposta no método *TRACTS* é a utilização de características comuns a qualquer

trajetória, possibilitando uma generalização do método e utilização em domínios amplamente diferentes.

A terceira etapa realiza uma transformação mais profunda dos dados de entrada, modificando os valores contínuos das características obtidas em dados discretos utilizando técnicas de séries temporais. O método empregado para a discretização dos dados é o SAX, que gera uma cadeia de caracteres (*strings*) a partir de uma série temporal. Esses dados são então informados como entrada em um método de tratamento de sequência de caracteres, para formação das matrizes. A formação da matriz torna possível estabelecer um conjunto de atributos para cada trajetória do conjunto de dados.

Finalmente, na quarta e última etapa, é realizado o processo de classificação onde um modelo de classificação é gerado e avaliado. Conforme acontece no processo KDD, a partir da avaliação desse modelo, pode-se identificar uma eventual necessidade de se retornar a alguma das etapas anteriores, de forma a melhorar a qualidade do modelo de classificação gerado.

É possível ter uma visão geral do método *TRACTS*, bem como cada uma das etapas descritas anteriormente, na Figura 13.

Cada uma das etapas, estabelecidas no processo do método *TRACTS*, serão detalhadas nas seções a seguir.

3.2 Preparação

Para que qualquer processo de descoberta de conhecimento possa ser realizado a partir dos dados obtidos de um dispositivo de rastreamento móvel, eles devem ser primeiramente preparados de acordo com a necessidade das operações que serão realizadas nas etapas seguintes.

Dessa forma, duas sub-etapas devem ser realizadas para que os dados estejam preparados para as etapas subsequentes. A primeira é a formatação dos dados das trajetórias para a aplicação desejada. A segunda é a adequação e eliminação ou atenuação de ruídos. Ambas sub-etapas são descritas a seguir.

3.2.1 Formatação dos dados

Os dados obtidos a partir da grande maioria dos dispositivos de rastreamento de objetos móveis atuais, dificilmente encontram-se em um formato que permita a sua utilização direta por algum método de classificação tradicional.

Devido a isso, nessa primeira sub-etapa, a tupla (id, x, y, z, t) de cada ponto da trajetória deve ser extraída a partir dos dados brutos obtidos dos dispositivos de rastreamento. As componentes espaciais latitude, longitude, e altura (x, y, z), bem como sua respectiva componente temporal (t) e o identificador dessa trajetória (id), devem ser extraídas em um formato apropriado para formar o conjunto de dados de entrada do método *TRACTS*.

Em muitos casos, é desejável utilizar as componentes de referência espacial em um formato diferente do que aquele contido nos dados brutos originais. Um exemplo de ocorrência bastante comum é quando se obtém dados brutos originais com referencial no formato *Universal Transverse Mercator* (UTM), mas o formato desejado para estudo

é o de Coordenadas Geográficas (latitude e longitude). Pode também haver casos onde o referencial dos dados originais é o mesmo do desejado, mas existe uma diferença na projeção utilizada. Esse caso pode ser observado com bastante frequência para dados no formato UTM.

Em todos esses casos, podem ser utilizadas fórmulas matemáticas de conversão entre as trajetórias. Algumas são mais triviais, quando apenas existe alguma diferença de projeção entre um mesmo formato, outras são mais complexas, quando existe a necessidade de se modificar o formato de referência espacial (como por exemplo entre UTM e Coordenadas Geográficas).

De qualquer forma, o resultado do processo de formatação deve ser um conjunto de dados onde todas as trajetórias estejam expressas em uma sequência de pontos contendo as informações no formato da tupla (id, x, y, z, t). Essa formatação possibilita o processamento das trajetórias pelas etapas seguintes do método.

3.2.2 Reconstrução e limpeza das trajetórias

Com as trajetórias formatadas, pode haver necessidade de realizar uma transformação/reconstrução dessas trajetórias, associadas ou não com uma limpeza delas. Tais tarefas são descritas a seguir.

3.2.2.1 Segmentação das trajetórias

A primeira tarefa que pode ser necessária ser executada no conjunto de dados recém-formatado é a adequação e reconstrução das trajetórias de modo a possibilitar uma melhor análise do conjunto de dados nas etapas seguintes.

Um exemplo dessa situação é quando os pontos da trajetória referente a um objeto móvel estejam todos agrupados em uma única trajetória, mas a aplicação exige que uma segmentação seja feita para que a análise seja possível.

Um caso onde ocorre esse tipo de situação é quando se deseja realizar a análise de origens e destinos de um veículo a partir de uma trajetória contínua obtida através de um dispositivo GPS (situação bastante comum, pois a maioria dos dispositivos GPS não segmenta a trajetória). Dessa forma, como a trajetória não foi segmentada originalmente pelo dispositivo móvel, só é possível observar uma origem e um destino para o veículo (início e fim da trajetória, respectivamente), impossibilitando uma análise mais detalhada da situação desejada.

Uma possível solução para esse problema seria a segmentação da trajetória contínua do em veículo em sub-trajetórias, quando o tempo decorrido entre um ponto p_n e o seu seguinte, p_{n+1}, seja maior que um determinado período de tempo. O ponto p_n seria considerado o destino do segmento de trajetória S_t e o ponto p_{n+1} seria a origem do segmento de trajetória S_{t+1}.

É possível visualizar um exemplo de segmentação de trajetória, para adequação do conjunto de dados em situações como a exemplificada, na Figura 14. Nesse caso, uma trajetória única, com 20 pontos, que se estende durante o tempo, foi sub-dividida em três trajetórias menores.

O tempo decorrido entre a aquisição dos 7º e o 8º pontos, bem como entre os 15º e o 16º pontos, foi maior que o tempo estabelecido como limitador, causando a segmentação dessa trajetória. Considerando que essa trajetória fosse do veículo do exemplo anterior, seria possível estabelecer que o veículo percorreu três trajetos: o

primeiro trajeto entre os 1º e 7º pontos (origem e destino, respectivamente), o segundo entre os 8º e 15º pontos e o último trajeto entre os 16º e 20º pontos.

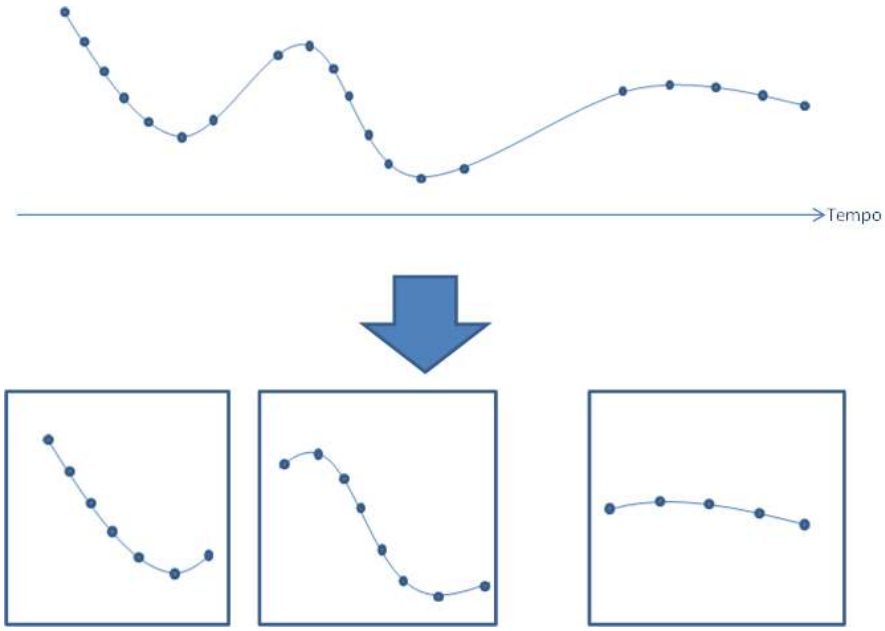


Figura 14: Segmentação de trajetória

3.2.2.2 Eliminação de ruídos

A segunda tarefa possível de ser realizada em um conjunto de dados formatos é a eliminação ou a atenuação de ruídos existentes na base de dados.

Grande parte dos algoritmos que possibilitam a descoberta de conhecimento assumem que os dados estão livres de ruídos (Zaiane, 1999). Entretanto, a maioria dos conjuntos de dados contém exceções, dados inválidos ou incompletos que podem complicar, e muitas vezes obscurecer, o processo de análise e em muitos casos comprometer a acurácia dos resultados.

Como consequência disso, a limpeza dos dados se torna vital. Muitas vezes, esse é um processo visto como "perda de tempo", mas a limpeza dos dados, por mais que consuma uma quantidade significativa de tempo, é uma das fases mais importantes de qualquer processo de descoberta de conhecimento. Em métodos de classificação de dados, como o proposto nesse trabalho, é fortemente recomendável o tratamento de ruído e de informações incompletas no conjunto de dados antes de proceder com qualquer tipo de análise.

Um dos tipos de ruído mais comumente observado em trajetórias de objetos móveis é o formado pela aquisição errônea de um ponto geográfico. Em certas ocasiões, os dispositivos de rastreamento, como o GPS, que dependem de triangulações matemáticas de acordo com satélites em órbita da Terra, podem perder a referência de um desses satélites (ou de um conjunto deles), prejudicando a precisão de aquisição da posição geográfica do objeto móvel.

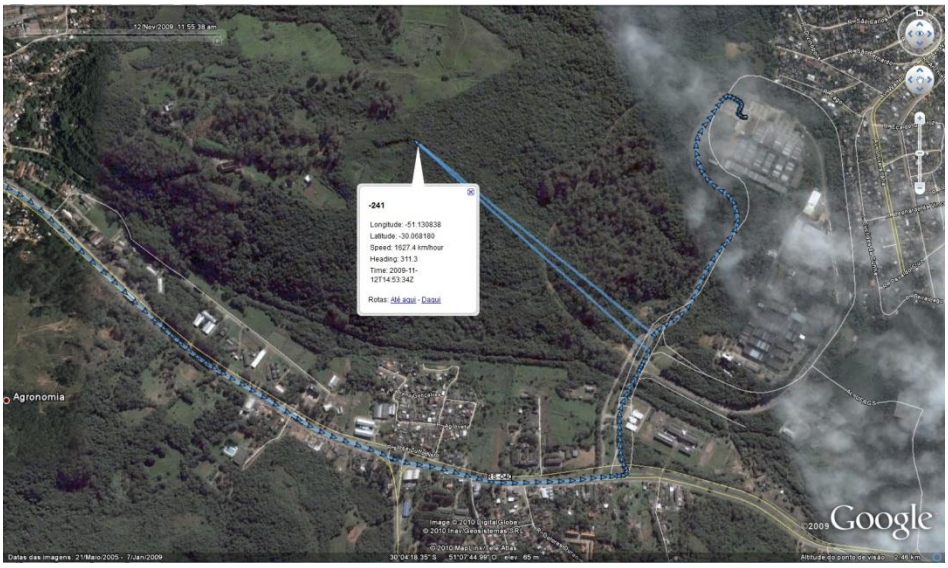


Figura 15: Ruído no conjunto de dados

Como visualizado na Figura 15, um dos pontos da trajetória considerada foi obtido de forma errônea pelo dispositivo de rastreamento, gerando um ponto incorreto na trajetória. A esse ponto foi atribuída uma velocidade de 1627,4Km/h, velocidade totalmente irreal em um trecho urbano percorrido por um carro. Dessa forma, se esse tipo de ruído não for tratado adequadamente, fatores tais como velocidade, aceleração e direção médias, serão afetados de forma significativa (e incorreta) para essa trajetória. Isso consequentemente alteraria o resultado da descoberta de conhecimento desse dado, e também a acurácia de precisão na classificação.

A detecção de ruídos como esse vão desde a estipulação de limites máximos de velocidade e aceleração, até demarcações geográficas de onde seria possível um ponto ocorrer (para casos onde se sabe qual o escopo geográfico de ocorrência dos pontos considerados).

Dessa forma, após a sua detecção, deve haver o tratamento desse ponto. No exemplo mostrado, ele poderia ser simplesmente eliminado, fazendo com que a trajetória seja remarcada entre os seu ponto anterior e o ponto posterior. Outra possível solução seria atenuar o próprio ponto de acordo a velocidade e posicionamento atribuídos aos seus pontos vizinhos.

3.3 Caracterização

Na etapa de caracterização dos dados é realizada a identificação e extração das características importantes a partir do conjunto de dados. Essa é uma etapa fundamental para o sucesso do método *TRACTS*, pois ela fornece a semântica necessária para a análise no processo de classificação.

Qualquer processo de classificação necessita de um ou mais atributos, que representarão as características da instância sendo classificada, e o atributo classe que identifica essa instância. No método *TRACTS* a classificação de trajetórias é realizada com os atributos, incluindo o atributo classe, sendo definidos utilizando-se as componentes (id, x, y, z, t) da trajetória. Entretanto, se esses atributos de classificação forem exatamente as componentes espaciais e a componente temporal, pouco mais que uma análise posicional em relação ao tempo poderá ser feita a partir dessa trajetória, o

que pode acabar não expressando diversos comportamentos significativos da trajetória, tais como comprimento, aceleração, etc.

Para que uma análise mais profunda possa ser realizada, um processo de caracterização mais elaborado deve ser realizado a partir dos dados das trajetórias dos objetos móveis. O primeiro passo nesse processo é a identificação e extração das características importantes das trajetórias desses objetos móveis.

A identificação das características que são relevantes para o domínio considerado não é uma tarefa trivial. Em muitos casos, essa definição de características pode ser um tanto quanto subjetiva, já que as possibilidades de extração de diferentes atributos, em diversos domínios espaço-temporais, podem ser muito grandes.

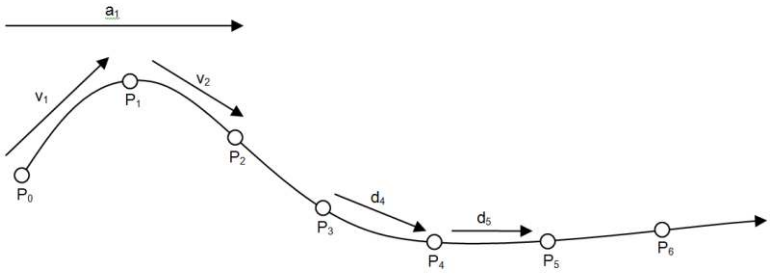


Figura 16: Trajetória e algumas de suas características

Entretanto, como estamos trabalhando com trajetórias de objetos em movimento, certas características serão sempre universais para todas as trajetórias, tais como velocidade, aceleração e direção vetorial absoluta, assim como a variação de cada uma dessas características em determinado intervalo. Além disso, essas características podem revelar comportamentos distintos de objetos móveis em suas trajetórias, o que é fundamental para o processo de classificação. Na Figura 16 é possível visualizar algumas dessas características, tais como as velocidades absolutas v_1 e v_2, a aceleração absoluta a_1, calculada a partir dessas duas velocidades, e as direções absolutas d_4 e d_5.

A forma de extração dessas características para utilização no processo de classificação também pode variar, conforme desejado. Uma das possibilidades mais comuns de extração das características é a obtenção dos valores geométricos globais das trajetórias, tais como distância percorrida, tempo de duração de cada trajetória e deslocamento realizado entre o ponto inicial e final da trajetória. Com isso, nenhuma etapa a mais de transformação é necessária para realizar o processo de mineração de dados através de algoritmos de classificação tradicionais, possibilitando a construção de modelos de classificação mais rapidamente.

Entretanto, existe uma grande desvantagem de se extrair as características somente dessa forma: a perda da análise de comportamentos locais da trajetória. Em trajetórias muito longas, é possível que o comportamento que define a trajetória aconteça em um curto período de tempo e espaço. Nessa situação, se tivermos para a análise somente dados globais da trajetória, tais como distância e deslocamento, pode ser muito difícil detectar essa variação pontual na trajetória, pois em uma trajetória muito extensa os valores que poderiam expressar esse diferencial estarão atenuados pelos valores dos demais pontos.

Um exemplo onde ocorre esse tipo de situação é na trajetória de furacões. Geralmente são trajetórias muito longas (que atravessam países inteiros), mas que atingem sua força máxima em curtas variações de tempo e espaço. Como se trata de um furacão, sua força máxima pode acabar sendo justamente o atributo classe da trajetória, mas como essa força máxima pode acontecer em um trecho muito reduzido da trajetória, os atributos globais podem não ajudar a caracterizar corretamente essa trajetória para processos de classificação tradicionais.

3.3.1 Extração de características no método *TRACTS*

Como foi visto anteriormente, a extração apenas de características globais para as trajetórias pode ocultar informações importantes relativas ao seu comportamento. Dessa forma, torna-se necessária uma análise mais detalhada das informações da trajetória.

Para isso, é possível considerar também as características geométricas locais de cada um dos pontos das trajetórias do conjunto de dados. Características tais como velocidade, aceleração e direção vetorial absolutas, assim como a variação de cada uma dessas características em determinado intervalo. Essas são as características utilizadas por *default* na aplicação do método em qualquer domínio.

Esse é um diferencial do método *TRACTS*, pois é o primeiro a considerar não apenas valores globais da trajetória como distância e deslocamento, mas também o valor das características em cada ponto da trajetória.

É possível extrair qualquer característica geométrica das trajetórias, formando assim um conjunto X de todas as características possíveis de se extrair de uma trajetória. Para uso no método *TRACTS*, é possível extrair das trajetórias do domínio considerado um subconjunto χ das características contidas em X. Seis características são mostradas aqui, mas qualquer característica geométrica das trajetórias pode ser extraída e utilizada no método *TRACTS*.

3.3.1.1 Comprimento

O comprimento é uma característica global da trajetória, calculada a partir do somatório da distância D formada entre todos os pares de pontos p_n e p_{n-1} consecutivos das trajetórias, as quais são compostas por $(i + 1)$ pontos.

$$Comp_\tau = \sum_{n=0}^{i} \Delta_{p_{n-1}}^{p_n}$$

3.3.1.2 Duração

A duração também é uma característica global da trajetória, calculada a partir do tempo transcorrido entro o primeiro e último pontos das trajetórias, as quais são compostas por $(i + 1)$ pontos.

$$Dur_\tau = \Delta t_{p_0}^{p_i}$$

3.3.1.3 Deslocamento

O deslocamento é uma característica global da trajetória, calculada a partir da distância D entre o primeiro e o último ponto da trajetória. Quando mais o valor do deslocamento se aproximar do comprimento de uma trajetória maior é a indicação que o objeto móvel se móvel em uma linha reta, podendo, dessa forma, evidenciar um

comportamento discriminatório importante para o objeto móvel sendo classificado. Nesse caso também é considerada uma trajetória com um total de i pontos.

$$Des_{\tau} = \Delta D_{p_0}^{p_i}$$

3.3.1.4 Velocidade entre dois pontos

A velocidade entre dois pontos é uma das características locais da trajetória, ou seja, calculada a cada par de pontos e não para a trajetória inteira, tais como as características globais. A partir da distância D e do tempo t que separam dois pontos p_n e p_{n-1} consecutivos na trajetória, onde n é o ponto da trajetória que está sendo caracterizado, é possível calcular a velocidade média v_n de deslocamento entre esses dois pontos e atribuí-la a p_n. Esse cálculo é feito para cada par consecutivo de pontos contidos na trajetória, utilizando a seguinte fórmula:

$$v_n = \frac{\Delta D_{p_{n-1}}^{p_n}}{\Delta t_{p_{n-1}}^{p_n}}$$

3.3.1.5 Aceleração entre duas velocidades

A característica de aceleração a_n da trajetória é uma característica local da trajetória calculada a cada três pontos da trajetória. Ela é calculada a partir de duas velocidades v_n e v_{n+1} consecutivas na trajetória. Como a aceleração necessita de duas velocidades já calculadas, ela é atribuída sempre ao ponto p_{n+1}.

$$a_n = \frac{v_{n+1} - v_n}{t_{n+1} - t_n}$$

3.3.1.6 Direção entre dois pontos e Variação da Direção entre duas direções

A direção absoluta d_n é obtida a partir de um par de pontos p_{n-1} e p_n de uma determinada trajetória, definida a partir do ângulo de inclinação entre esses pontos. O seu valor varia de 0º a 359º. O valor dessa característica é atribuído ao ponto p_n.

$$d_n = \theta(p_{n-1}, p_n)$$

A partir dos vetores de direção d_n formados entre dois pares de pontos de uma determinada trajetória, é possível calcular os fatores de variação de direção nessa trajetória.

O valor do vetor de direção é dado de acordo com o ângulo formado em relação ao norte geográfico, que tem um valor de 0º. A diferença ς entre um vetor de direção d_n e outro consecutivo d_{n+1}, varia com valores entre 0º e 180º. É importante observar que 180º é o valor máximo considerado na diferença entre dois vetores de direção absolutos, já que não são considerados valores positivos ou negativos de diferença, que poderiam gerar, por exemplo, valores como -200º ou 160º, que são na verdade a mesma diferença de direção, mas se tivessem valores diferentes poderiam afetar de forma negativa o processo de classificação.

Dessa forma, esse valor caracteriza a variação de direção vd_n. Da mesma forma como a aceleração, são necessárias duas direções absolutas já calculadas. Dessa forma, o valor da variação de direção é atribuído ao ponto p_{n+1}.

$$vd_n = \varsigma(d_n, d_{n+1})$$

3.3.1.7 Obtenção das características a partir das trajetórias

A forma de obtenção das características citadas a partir de cada uma das trajetórias do conjunto de dados é estabelecida pelo seguinte método:

Entrada: T //conjunto de trajetórias.
Λ_χ //fórmulas de cálculo para cada característica χ.

Saída: C_χ //conjunto de séries temporais para cada característica local χ.
G_χ //conjunto das características globais para cada característica
// global χ.

Método:

Para cada trajetória τ em T faça:
//inicialização das séries temporais dessa trajetória para cada
// característica χ como vazia.
$C_{\chi[\tau]}$ = null
Para cada ponto não processado n em τ faça:
//calcula o valor da característica local χ para o ponto n da trajetória τ
$Calcule(c_{\chi[\tau,n]})$
//insere o valor calculado no conjunto de séries temporais
$C_{\chi[\tau,n]} = c_{\chi[\tau,n]}$
Fim Para
//calcula o valor da característica global χ para cada trajetória τ
$Calcule\left(g_{\chi[\tau]}\right)$
//insere o valor calculado no conjunto de características globais para as
// trajetórias
$G_{\chi[\tau]} = g_{\chi[\tau]}$
Fim Para

3.3.1.8 Análise da abordagem

Em suma, um subconjunto χ de características que podem ser obtidas a partir das trajetórias do conjunto de dados de entrada é:

- Comprimento da trajetória ($Comp_\tau$);
- Duração da trajetória (Dur_τ);
- Deslocamento da trajetória (Des_τ);
- Velocidade entre dois pontos consecutivos (v_n);
- Aceleração entre duas velocidades consecutivas (a_n);
- Direção entre dois pontos consecutivos (d_n);

- Variação da direção entre duas direções consecutivas (vd_n).

Como foi citado anteriormente, o método *TRACTS*, ao contrário da abordagem adotada em diversos outros métodos, considera tanto as características globais quanto as características locais das trajetórias. A características globais, logo após a sua extração, já estão prontas para serem consideradas como entrada para um processo de classificação. Entretanto, as características locais das trajetórias não são utilizadas diretamente como atributos de entrada para o algoritmo de classificação. Como cada um dos valores das características locais são obtidos a partir de um par de pontos (v_n e d_n) ou dois pares de pontos consecutivos (a_n e vd_n), e não é realizada uma consolidação do valor desses atributos para a trajetória inteira, não é possível estabelecer, por exemplo, um único atributo representando a velocidade para uma dada trajetória do conjunto de dados.

Dessa forma, a trajetória antes representada somente por uma sequência de pontos $(p_1, p_2, p_3, \ldots, p_n)$, agora pode ser representada por um conjunto de séries temporais. Considerando χ como o subconjunto das características extraídas das trajetórias, é formado um conjunto $\mathrm{C}_{\chi[\tau]}$ de séries temporais para cada trajetória τ. Para as quatro características extraídas, que formam o subconjunto χ de características, teremos as seguintes séries temporais formadas para cada uma das τ trajetórias:

- $\mathrm{C}_{v[\tau]} = ([p_1, v_1], [p_2, v_2], [p_3, v_3], \ldots, [p_n, v_n])$
- $\mathrm{C}_{a[\tau]} = ([p_1, a_1], [p_2, a_2], [p_3, a_3], \ldots, [p_n, a_n])$
- $\mathrm{C}_{d[\tau]} = ([p_1, d_1], [p_2, d_2], [p_3, d_3], \ldots, [p_n, d_n])$
- $\mathrm{C}_{vd[\tau]} = ([p_1, vd_1], [p_2, vd_2], [p_3, vd_3], \ldots, [p_n, vd_n])$

Todavia, para que essa análise seja possível, é necessária uma transformação mais profunda dos dados, utilizando técnicas que reduzam os dados produzidos nesse formato de série temporal e possibilitem transformar os valores das características em atributos da trajetória, mantendo a capacidade de análise detalhada do seu comportamento.

A etapa que realiza essa transformação é detalhada na seção a seguir.

3.4 Transformação

Essa etapa realiza todas as operações necessárias para deixar os dados prontos para serem utilizados como entrada no algoritmo de mineração de dados, na etapa seguinte.

Como na classificação de dados tradicional os algoritmos utilizam diversos atributos com formato diferente da formatação espaço-temporal das trajetórias, ou mesmo das séries temporais formadas na etapa anterior, é necessária a transformação desses dados contínuos do conjunto de dados em atributos que possam ser informados como entrada para o processo tradicional de classificação. Dessa forma, os dados contínuos das séries temporais geradas na etapa anterior são processados nessa etapa de tal forma, que o conjunto de séries temporais seja resumido a um único registro com as características locais relevantes dessas séries, em formato adequado para os algoritmos tradicionais de classificação.

A primeira técnica utilizada é o método de agregação simbólica (SAX), introduzido na seção 2.4. Na aplicação deste método, o conjunto das séries temporais de cada característica é considerado, de tal forma que cada letra do alfabeto utilizado tenha o mesmo número de ocorrências no conjunto das séries temporais daquela característica.

Com as séries temporais transformadas em sequências de caracteres, é necessário utilizar um método de tratamento de *strings* para possibilitar o processo de classificação dessas trajetórias. Para isso é utilizado o método TSB, explicado na seção 2.5, que realiza a transformação das sequências de caracteres produzidas pelo método SAX em matrizes.

A aplicação de ambos os métodos é detalhada nas subseções a seguir.

3.4.1 Aplicação do método SAX

Conforme explicado anteriormente, o método SAX permite a redução dimensional e a discretização de uma série temporal. Nas etapas anteriores do método *TRACTS*, as trajetórias foram transformadas em séries temporais, onde cada ponto da série temporal possui um conjunto de valores numéricos e contínuos, representando as características daquele ponto. Dessa forma, é possível realizar a aplicação do método SAX nessas séries temporais.

Como cada uma das características locais extraídas das trajetórias pode expressar um comportamento diferente das demais, para cada uma delas (aceleração, velocidade, direção absoluta e variação da direção), será aplicada a função de conversão para o método SAX.

O único parâmetro que deve ser informado para o método SAX é o número de faixas f a serem utilizadas pelo método. Como explicado anteriormente, essa definição estabelecerá o tamanho do alfabeto que será observado na *string* resultante da função de conversão do método SAX.

Para que a distribuição das letras do alfabeto seja equiprovável, a função calcula inicialmente, para cada característica, a quantidade de elementos que deve haver em cada uma das f faixas. Isso é feito considerando-se todas as trajetórias contidas no conjunto C_χ geradas na etapa de Caracterização para cada característica. A forma de realizar essa distribuição é explicada na subseção 2.4.1.

De forma geral, o método de construção das *strings* no método *TRACTS*, utilizando as técnicas do método SAX é o seguinte:

Entrada: C_χ //conjunto de séries temporais para cada característica local χ
// para todas as trajetórias.
f //número de faixas a serem utilizadas no método SAX

Saída: S_χ //conjunto de *strings* SAX para cada característica local χ para
// todas as séries temporais.

Método:

$A = [a, b, c, \ldots, z]$ //Alfabeto disponível para o SAX

Para cada característica χ em X faça:

$e_\chi = \frac{n_\chi}{f}$ //estabelece o número de elementos em cada faixa para a característica.

Se o resto da divisão $\frac{n_\chi}{f} > 0$

$e_\chi = e_\chi + 1$ //forçará a adição de um elemento nas primeiras faixas para
//ajustar a equiprobabilidade

```
        Fim Se
        iteração = 0 //contador de iteração
        Para cada elemento λ_χ[τ] ordenado crescentemente por valor no conjunto das
        series temporais C_χ faça:
                n = índice(λ_χ[τ]) //recupera a posição do elemento λ_χ[τ] em C_χ
                iteração = iteração + 1
                i = inteiro(iteração / e_χ) //considera somente a parte inteira da divisão
                S_χ[c,n] = A_i          //a i-esima letra do alfabeto A é colocada na n-esima
                                        //posição da string SAX da série temporal c_χ[τ] em C_χ para a
                                        //característica χ
        Fim Para
Fim Para
```

Como resultado do processo de conversão através do método SAX, é obtido um conjunto de $S_{\chi[\tau]}$ para cada τ trajetória, para cada característica local de χ.

Após a obtenção desse conjunto de sequência de caracteres, representantes das séries temporais (trajetórias), torna-se necessário utilizar algum método para o tratamento de *strings*, para que seja possível o processo de classificação das trajetórias na etapa seguinte. Um método para esse fim deve ser utilizado na próxima subetapa de transformação do método *TRACTS*.

3.4.2 Aplicação do método TSB

Um dos métodos disponíveis atualmente para tratamento de sequência de caracteres é o método *Time Series Bitmaps* (TSB). Conforme explicado na seção 2.5, esse método tem a capacidade de transformar uma sequência de caracteres qualquer em um mapa de bits.

Uma das funcionalidades nativas desse método é a possibilidade de visualização das matrizes resultantes, permitindo a sua classificação visual. Entretanto, quando diversas séries temporais são consideradas ao mesmo tempo no conjunto de dados, como no método *TRACTS*, a tarefa de classificação visual pode-se tornar muito onerosa.

Sendo assim, a utilização desse método no *TRACTS* irá até o ponto onde as matrizes TSB são geradas com a graduação de cada uma de suas células valoradas. A partir desse momento, são gerados atributos de classificação a partir de cada uma das células das matrizes geradas, possibilitando dessa forma a sua utilização em um processo de classificação automatizado.

O método de construção das matrizes TSB e a extração dos atributos de cada uma dessas matrizes são detalhados a seguir.

```
Entrada:    S_χ     //conjunto de strings SAX para cada característica local χ.
            ∂       //dimensão da matriz TSB
            ρ       //profundidade da matriz TSB

Saída:      Φ_χ     //conjunto de matrizes TSB para cada característica local χ.

Método:
```

```
Para cada célula ω em Ω faça:
        Ω[∂,ρ] = InicializaSubstringTSB(ω)      //matriz modelo com todas as substrings
                                                // possíveis da matriz TSB
Fim Para
Para cada característica χ em X faça:
        Para cada string SAX correspondente a cada série temporal s_χ[c] em S_χ faça:
                Φ_χ[c] = null     //inicializa a matriz TSB dessa série temporal para
                                  // todas as características locais
                Para cada célula φ_χ[c][∂,ρ] em Φ_χ[c] faça:
                        φ_χ[c][∂,ρ] = calculaOcorrencias(Ω[∂,ρ], s_χ[c])      //Realiza a contagem das
                                                                              // ocorrências da substring
                                                                              // Ω[∂,ρ] na string s_χ[τ].
                Fim Para
                Φ_χ[c] = normaliza(Φ_χ[c])      //Normaliza os valores absolutos em Φ_χ[c] para
                                                // valores percentuais
        Fim Para
Fim Para
```

Com isso, temos no final do processo de aplicação do método TSB um conjunto $\Phi_{\chi[c][\partial,\rho]}$ de matrizes, representando as *strings* SAX de todas as séries temporais, com a contagem de ocorrências das *substrings* em cada uma das $(\partial * \rho)$ células de cada matriz TSB.

Cada uma dessas células, para cada uma das características, será então transformada em um par (*atributo*, *valor*). Os atributos serão a composição da característica χ com cada *substring* do conjunto $\Omega_{[\partial,\rho]}$ e os seus respectivos valores será o valor contido em $\Phi_{\chi[c][\partial,\rho]}$ para cada série temporal.

É possível constatar que o número de atributos α que serão gerados a partir do conjunto Φ de matrizes TSB no método *TRACTS* é dado por:

$$\alpha = \chi * \partial^{\rho}$$

Dessa forma, já é possível iniciar a última etapa do método: a etapa de classificação de dados.

3.5 Classificação

A classificação é a última etapa do método *TRACTS*. Nessa etapa, é realizada a construção de um modelo de classificação capaz de classificar trajetórias do conjunto de dados ainda não rotuladas.

Após a geração do modelo de classificação, esse modelo deve ser avaliado através de métricas tais como acurácia e precisão. O desempenho do modelo de classificação gerado nessas métricas estabelece a necessidade de ou se manter esse modelo, ou ajustar os parâmetros em uma etapa anterior. Com isso, um novo modelo será gerado a partir da nova configuração.

3.5.1 Ferramenta de classificação

O processo de classificação pode ser realizado com o auxílio de qualquer ferramenta computacional que implemente algum algoritmo de classificação. Existem algumas ferramentas que reúnem diversas funcionalidades úteis para a tarefa de classificação, facilitando a realização do processo. Uma das ferramentas mais conhecidas e utilizadas ferramentas para esse fim é a ferramenta Weka. As funcionalidades principais dessa ferramenta foram detalhadas na seção 2.6.

Como descrito anteriormente, a ferramenta Weka possibilita a entrada de dados tanto via um banco de dados, como PostgreSQL, quanto a partir de um arquivo texto no formato "arff". Ambos os formatos possuem estrutura de dados semelhante para entrada na ferramenta.

Nos experimentos do capítulo 4, foi utilizado a formato de entrada de dados em arquivo. Dessa forma, será detalhado na subseção a seguir o processo de montagem desse arquivo.

3.5.2 Montagem do arquivo de classificação

O arquivo "arrf" contém a informação de atributos que caracterizam um determinado conjunto de dados com seus respectivos valores. Um dos atributos é considerado o atributo classe, que é aquele que identifica o elemento do conjunto de dados.

No método *TRACTS*, os atributos principais expressam o comportamento do subconjunto χ das características extraídas de uma trajetória. Essas características são:

- Comprimento
- Duração
- Deslocamento
- Velocidade
- Aceleração
- Direção
- Variação da direção

Já o atributo classe será o atributo que expressa a característica objeto do processo de classificação. Pode ser considerado, por exemplo, o tipo de objeto móvel que percorre a trajetória, tal como um animal, ou a força de um furacão.

Os valores dos atributos principais derivados das características locais (velocidade, aceleração, direção e variação da direção) são obtidos através do método TSB, que produz a matriz $\Phi_{\chi[c][\partial,\rho]}$, a qual contém os valores de ocorrência de cada *substring* possível do alfabeto SAX nas diversas *strings* produzidas pelo método SAX para cada uma das séries temporais representantes das trajetórias. Para cada uma das quatro características consideradas, deve ser construída uma matriz TSB, possibilitando obter os valores para todas as características em cada conjunto de matrizes TSB.

De forma geral, cada célula da matriz TSB para cada característica pode ser considerada como um atributo distinto do conjunto de entrada a ser classificado. Em cada uma dessas células haverá um valor, o totalizador das ocorrências distintas das *substrings* do alfabeto SAX nas *strings* SAX das séries temporais, que será considerado o respectivo valor do atributo. Com isso é formada uma sequência de atributos obtidos

pela combinação das χ características com todas as *substrings* contidas em $\Omega_{[\partial,\rho]}$, gerando conjuntos de atributos como por exemplo:

- Velocidade_aa
- Velocidade_ab
- Velocidade_ba
- Velocidade_bb
- Aceleracao_aa
- Aceleracao_ab
- ...

Os últimos atributos são os derivados das características globais das trajetórias (comprimento, duração e deslocamento). Entretanto, o valor $G_{\chi_{[\tau]}}$ de cada uma dessas características locais para cada trajetória τ não é atribuído diretamente a cada instância do arquivo de classificação para cada uma das trajetórias, é realizado um cálculo percentual entre todos os valores absolutos da mesma característica para a mesma série temporal. Por exemplo, se para o atributo velocidade tivéssemos o seguinte conjunto de valores absolutos (ocorrências na string SAX) na matriz TSB em uma mesma série temporal:

- Velocidade_aa: 21
- Velocidade_ab: 15
- Velocidade_ba: 5
- Velocidade_bb: 30

O arquivo "arff" teria os seguintes valores para cada atributo:

- Velocidade_aa: 29,58
- Velocidade_ab: 21,13
- Velocidade_ba: 7,04
- Velocidade_bb: 42,25

Esse procedimento é muito importante para o processo de classificação, já que sem a transformação não seria possível comparar de forma equivalente as diversas séries temporais/trajetórias, pois nem sempre existe o mesmo número de pontos para todas as instâncias do conjunto de dados.

Assim sendo, o arquivo de classificação "arff" conterá uma quantidade total α_t de atributos, onde α_l representa o número de características locais consideradas, α_g as características globais e α_c o atributo de classificação:

$$\alpha_t = (\alpha_l * \partial^{\rho}) + \alpha_g + \alpha_c$$

Como teremos sempre $\alpha_c = 1$ e o número de atributos derivados de características globais é, geralmente, reduzido (no caso apresentado, apenas três características globais são extraídas), o número de atributos é definido de forma mais significativa pela profundidade (ρ) escolhida para a matriz TSB. Apesar da possibilidade de haver muitos atributos disponíveis como entrada no arquivo de dados para realizar a tarefa de classificação, a maioria dos algoritmos tende a ignorar os atributos menos significativos, dando preferência para os atributos que realmente possam fazer diferença na classificação dos elementos do conjunto de dados.

3.5.3 Pré-processamento e escolha do algoritmo de classificação

Com a geração do arquivo de entrada para o algoritmo de classificação, contendo os atributos e o rótulo de classe, é possível começar a tarefa de classificação.

Como explicado anteriormente na seção 2.6, é possível manipular os dados de entrada com diversos filtros de pré-processamento. Como o conjunto de dados possui essencialmente números contínuos, um filtro que se mostra interessante é o filtro de discretização supervisionado, que categoriza os valores contínuos de acordo com a distribuição das classes nos dados de entrada.

Após a etapa de pré-processamento, é realizada a mineração de dados propriamente dita, através da escolha de um algoritmo para a tarefa de classificação. Nessa etapa é escolhido um algoritmo tradicional de classificação, tal como, por exemplo, *Naive Bayes* ou C4.5. Para cada um dos algoritmos escolhidos deve ser executado o processo de mineração de dados e colhidas as métricas de desempenho para avaliação do modelo gerado.

A escolha do melhor algoritmo de classificação é difícil de ser estabelecida a priori. A forma mais utilizada para essa escolha é a experimentação de vários algoritmos em busca das melhores métricas de classificação.

3.5.4 Geração do modelo de classificação

A cada execução de um algoritmo de classificação, será gerado um modelo de classificação distinto. O formato do modelo de classificação gerado (árvore, estatístico, etc.) também irá depender do algoritmo de classificação escolhido na sub-etapa anterior.

Entretanto, qualquer que tenha sido o algoritmo escolhido, o modelo de classificação gerado poderá ser utilizado para encontrar o rótulo de classe de registros ainda não rotulados. Com isso, trajetórias ainda não classificadas, que não entraram no conjunto original de dados, podem agora receber um rótulo de classe com base no modelo gerado.

Após a geração do modelo de classificação para trajetórias, o método *TRACTS* é finalizado. Entretanto, como estabelecido no modelo KDD, as métricas de performance podem ser avaliadas para o modelo gerado, evidenciando a necessidade de se retomar alguma das etapas anteriores do método.

3.5.5 Avaliação do modelo gerado

Todos os modelos gerados a partir da classificação de trajetórias do conjunto de dados podem e devem ser avaliados em relação a sua performance. Nesse momento, é avaliada a acurácia na predição na classificação realizada a partir desse modelo de classificação.

Conforme explicado na subseção 2.6.2, podem ser utilizadas diversas métricas para avaliação do modelo de classificação. A partir da matriz de confusão é realizada a extração de métricas tais como acurácia, taxa de erro, positivos verdadeiros e positivos falsos. Outras medidas também podem ser utilizadas, de acordo com o domínio específico da aplicação, mas as métricas extraídas a partir da matriz de confusão podem ser utilizadas como universais para cada domínio considerado.

Essas métricas permitem traçar bases comparativas entre os diversos algoritmos de classificação ou também com outros métodos de classificação de trajetórias. Como em

todo processo de descoberta de conhecimento, se os resultados estiverem aquém do esperado, o processo deve ser refeito a partir de alguma das etapas anteriores.

4 EXPERIMENTOS REALIZADOS

O método *TRACTS* foi testado e validado utilizando os mesmos dados do trabalho realizado pelo grupo de Jiawei Han (Lee, Han, Gonzalez, & Li, 2008), que introduziu o método *TraClass*, também para classificação de trajetórias. Isso permitiu uma base comparativa para os resultados obtidos.

Três bases de dados foram utilizadas:

1. Um conjunto de dados contendo trajetórias de três tipos de animais distintos, rastreados através de coleiras RFID, disponível em (Pacific Northwest Research Station, 2005).
2. Dados de trajetórias de navegação de dois barcos. Os dados podem ser encontrados em (Monterey Bay Aquarium Research Institute, 2001).
3. Uma base de dados contendo trajetórias de rastreamento de furacões, disponível em (Unisys, 2009).

Para os três experimentos foram utilizadas características locais e globais das trajetórias, formando o seguinte conjunto de características:

- *Características locais*: velocidade (v_n), aceleração (a_n), direção (d_n) e variação da direção (vd_n).
- *Características globais*: comprimento ($Comp_\tau$), duração (Dur_τ) e deslocamento (Des_τ).

As fórmulas e métodos de cálculo de cada uma dessas características são detalhadas na subseção 3.3.1. É importante observar que qualquer característica espaço-temporal pode ser utilizada no método *TRACTS*. Para demonstração de utilização do método, foram utilizadas essas sete características.

Como foi citado a partir da seção 3.3, as características globais não necessitam passar pela etapa de transformação e já são utilizadas diretamente na tarefa de classificação. Entretanto, as características locais necessitam de uma transformação adicional antes de serem utilizadas para a classificação de trajetórias. Dessa forma, cada uma delas gerou uma série temporal para cada trajetória, gerando dessa forma quatro séries temporais para cada uma das trajetórias do conjunto de dados.

Na etapa de transformação, cada um dos valores das características locais obtidos na etapa anterior foi mapeado para um símbolo, através do método SAX (como as trajetórias eram relativamente pequenas, com uma média de 433 pontos em 2033 trajetórias, não houve necessidade de redução de dimensionalidade) e, logo após, inserido em uma matriz de mapa de bits de série temporal. Para realizar a agregação simbólica com o método SAX, todos os n valores, para cada uma das quatro características obtidas de cada domínio, são considerados juntamente. Para cada caraterística, é formada uma lista ordenada do menor para o maior valor, que é dividida em um número equiprovável de elementos, de acordo com o número de faixas estabelecido. Conforme explicado na subseção 3.4.1, essas faixas irão formar os símbolos SAX que serão mapeados para cada uma das trajetórias do domínio.

As séries temporais geradas pelo método SAX foram então submetidas ao método TSB para a geração das matrizes e, consequentemente, dos atributos utilizados para a classificação. Essa etapa da transformação é detalhada na subseção 3.4.2.

Com o arquivo de classificação contendo os atributos e seus respectivos valores gerados a partir das características globais e locais, foi utilizada a ferramenta Weka (Frank, Hall, Holmes, Kirkby, & Pfahringer, 2005), que possibilitou a utilização de diversos algoritmos de classificação, bem como a análise comparativa da acurácia de cada um dos modelos gerados. Em todas as execuções foi utilizada validação cruzada com 10 subconjuntos. O computador utilizado em todos os experimentos estava equipado com um processador intel CORE 2 Quad e 4GB de memória.

Toda a implementação dos algoritmos para formação das séries temporais, transformação das mesmas em uma *string* SAX e construção das matrizes TSB foram construídas em PG/SQL dentro de um banco de dados Postgres rodando na máquina citada anteriormente.

Com isso, a partir da análise dos dados dos três domínios de aplicação citados, o resultado na eficácia de classificação do método poderá ser validado sob domínios inteiramente distintos.

4.1 Experimentação – Trajetórias de Animais

Com relação aos dados das trajetórias dos animais, foram coletadas informações do projeto Starkey, que estuda a variação populacional de animais selvagens existentes em parques nacionais dos Estados Unidos. A principal motivação deste projeto reside na administração destes parques, com o controle de sua fauna. Os dados foram coletados entre os anos de 1993 até 1996 e as trajetórias dos animais analisados em questão são: veados, gado e alces, que são justamente as classes definidas para este domínio de aplicação. Todos os animais foram rastreados a partir de coleiras RFID. A base de dados com a trajetória de animais pode ser obtida a partir de (Pacific Northwest Research Station, 2005).

Inicialmente dispunha-se de um conjunto de dados com trajetórias dos animais com tuplas no formato bruto (id, x, y, z, t). Nesse conjunto de dados, houve necessidade de eliminação de ruídos, já que foi observado que a velocidade média entre alguns pontos era maior que a velocidade possível para o respectivo animal que originava essa trajetória. Também foi possível observar certa irregularidade no intervalo de obtenção das coordenadas geográficas, o que eventualmente ocasionava grande tempo decorrido entre a aquisição de dois pontos consecutivos. Dessa forma uma segmentação das trajetórias em sub-trajetórias menores foi realizada.

A partir dessas trajetórias, foi executada a função de transformação do conjunto de dados, criando um conjunto de séries temporais a partir das características extraídas dessas trajetórias. O tempo de geração das séries temporais pode ser visualizado na Tabela 2. É possível observar que o número de séries temporais geradas é o número de características locais utilizadas (quatro), multiplicada pelo número de trajetórias.

Número de trajetórias	Número de pontos	Séries temporais geradas	Tempo de geração (segundos)
253	287134	1012	634,06

Tabela 2: Geração das séries temporais para os animais

Com as sérias temporais geradas, elas foram então transformadas em *strings* SAX com alfabetos de tamanhos 3, 4, 5 e 7. O tempo de transformação das séries temporais em strings SAX é mostrado na Tabela 3.

Tamanho do alfabeto SAX	Tempo de geração da string SAX (segundos)
3	5260,02
4	5896,67
5	5405,61
7	5327,01

Tabela 3: Tempo de geração da *string* SAX para os animais

Para cada tamanho de alfabeto SAX, foi construída uma matriz TSB com profundidades variando de 1 a 6, dependendo do tamanho do alfabeto SAX, de forma a não gerar uma matriz TSB com quantidade muito grande de células, o que tornaria o processo computacional bastante demorado. O tempo de geração das matrizes TSB pode ser visto na Tabela 4.

Tamanho alfabeto SAX	Profundidade da matriz TSB	Tempo geração matriz TSB (segundos)
3	1	88,06
3	2	95,63
3	3	118,24
3	4	119,72
3	5	136,59
3	6	675,7
4	1	208,68
4	2	240,33
4	3	279,31
4	4	299,64
4	5	309,6
5	1	142,2
5	2	155,37
5	3	222,43
5	4	539,4
7	1	92,83
7	2	171,77
7	3	194,18
7	4	1180,94

Tabela 4: Tempo de geração das matrizes TSB para os animais

A partir do conjunto de matrizes *TSB* geradas, construiu-se o arquivo “arff” para entrada na ferramenta Weka. É possível visualizar parte do arquivo gerado para a configuração *TRACTS*.3.1 na Figura 17. Foram utilizadas as característica locais de velocidade, aceleração, direção e variação da direção para cada uma das três letras do alfabeto SAX dessa configuração. As três características globais que podem ser observadas são o comprimento, deslocamento e duração. Por fim, o atributo classe, que identifica a trajetória (diz qual tipo de animal gerou a trajetória) é utilizado.

Como explicado na subseção 3.5.2, os valores para as características locais são os valores percentuais, onde a soma de todas as ocorrências de uma mesma característica é 100%. Isso permite a comparação das diversas séries temporais/trajetórias seja realizada utilizando valores proporcionais e equivalentes, e não absolutos, o que dificultaria muito o processo de classificação para um conjunto de dados com quantidade heterogênea de pontos nas trajetórias da amostra.

```
@RELATION animais.0.3.1
@ATTRIBUTE Velocidade_A REAL
@ATTRIBUTE Aceleracao_A REAL
@ATTRIBUTE Direcao_A REAL
@ATTRIBUTE VariacaoDirecao_A REAL
@ATTRIBUTE Velocidade_B REAL
@ATTRIBUTE Aceleracao_B REAL
@ATTRIBUTE Direcao_B REAL
@ATTRIBUTE VariacaoDirecao_B REAL
@ATTRIBUTE Velocidade_C REAL
@ATTRIBUTE Aceleracao_C REAL
@ATTRIBUTE Direcao_C REAL
@ATTRIBUTE VariacaoDirecao_C REAL
@ATTRIBUTE Comprimento REAL
@ATTRIBUTE Deslocamento REAL
@ATTRIBUTE Duracao REAL
@ATTRIBUTE Classe {VEADO,ALCE,GADO}
@DATA
17.84,40.57,33.33,24.53,35.21,14.62,35.68,34.91,46.95,44.81,30.99,40.57,53065.81,885.89,1100860.00,VEADO
36.73,35.83,36.41,29.34,35.09,29.86,29.05,36.27,28.17,34.31,34.53,34.39,411182.85,1376.41,39229063.00,VEADO
44.44,20.96,33.43,30.51,37.19,58.09,35.35,37.13,18.37,20.96,31.22,32.35,344469.03,1047.85,42745710.00,VEADO
40.57,28.32,33.89,29.76,35.95,41.85,31.74,34.73,23.49,29.83,34.37,35.51,713082.91,2545.94,74402565.00,VEADO
33.75,32.98,31.32,34.89,33.41,34.89,32.17,34.33,32.85,32.13,36.51,30.78,623252.76,2549.12,38137197.00,ALCE
24.36,38.82,36.21,33.59,33.14,23.37,31.41,37.62,42.49,37.81,32.37,28.79,647501.22,9180.69,71758656.00,ALCE
24.30,39.01,34.96,36.99,31.74,21.64,31.47,30.86,43.95,39.36,33.57,32.14,826854.23,7821.95,40244713.00,ALCE
23.89,40.14,37.82,34.74,32.05,19.59,28.93,29.09,44.06,40.26,33.25,36.18,229697.77,540.83,5732065.00,ALCE
30.05,32.41,34.17,34.94,35.71,34.20,32.57,35.24,34.24,33.39,33.25,29.82,544201.74,432.67,39946612.00,ALCE
22.24,43.12,35.93,30.80,32.10,16.53,30.84,31.43,45.67,40.35,33.23,37.77,347702.63,2875.62,8718750.00,ALCE
23.43,41.30,39.26,31.68,34.02,17.34,28.79,31.80,42.54,41.36,31.95,36.52,345791.41,1779.13,8718273.00,ALCE
31.20,33.49,29.31,35.66,34.02,33.40,34.97,33.49,34.78,33.11,35.72,30.85,378148.79,0.00,11232509.00,ALCE
28.99,36.16,33.18,28.84,36.04,26.96,30.87,34.20,34.97,36.88,35.95,36.96,335312.00,3409.06,9071611.00,ALCE
28.94,38.68,33.09,31.96,33.14,24.16,33.91,33.84,37.93,37.17,33.00,34.20,497974.47,3771.54,39871339.00,ALCE
34.55,33.33,28.91,35.18,32.60,32.51,36.03,32.15,32.85,34.15,35.06,32.67,588472.62,1544.34,40015424.00,ALCE
26.98,41.57,31.53,29.33,33.09,16.56,36.21,33.53,39.93,41.87,32.25,37.13,339268.81,366.20,8720569.00,ALCE
35.18,30.13,27.40,37.48,30.24,41.38,39.52,34.93,34.58,28.49,33.08,27.59,359422.82,6920.19,40003375.00,ALCE
24.29,43.32,33.73,32.09,33.31,14.32,34.14,32.03,42.40,42.36,32.13,35.89,348183.09,2284.73,8717751.00,ALCE
33.07,27.03,29.64,40.78,34.01,43.59,41.03,35.47,32.92,29.38,29.33,23.75,329829.63,3167.24,39990080.00,ALCE
25.05,40.08,30.70,30.79,35.86,17.29,36.34,31.61,39.09,42.63,32.97,37.60,283089.38,2078.68,8713570.00,ALCE
26.07,37.70,32.32,35.44,32.11,24.54,37.72,32.90,41.82,37.76,29.96,31.66,1074428.28,1942.83,103405402.00,ALCE
34.19,30.10,30.32,36.57,34.19,36.89,38.39,33.33,31.61,33.01,31.29,30.10,129503.01,6541.44,9076841.00,ALCE
36.26,28.10,27.48,31.41,34.96,43.80,40.58,37.90,28.78,28.10,31.94,30.69,271868.06,4760.18,8347381.00,ALCE
32.25,30.69,29.70,36.55,35.56,38.29,39.89,34.76,32.20,31.02,30.41,28.69,705049.29,1806.24,41729142.00,ALCE
31.21,35.94,32.54,32.65,35.11,29.02,35.28,31.77,33.69,35.05,32.18,35.58,303670.74,6236.10,9057158.00,ALCE
21.53,41.65,29.68,35.75,30.97,17.31,36.03,30.16,47.51,41.04,34.29,34.09,366036.80,4590.00,8711988.00,ALCE
42.26,26.56,26.49,31.98,35.69,46.71,37.38,33.73,22.06,26.73,36.12,34.28,439526.58,524.79,74397414.00,VEADO
17.88,40.59,31.80,36.99,29.89,14.71,38.23,33.31,52.23,44.70,29.96,29.70,389924.06,3999.12,8700123.00,ALCE
40.06,27.81,26.45,35.16,34.58,45.21,36.90,32.31,25.36,26.98,36.65,32.52,531542.36,939.63,42770084.00,VEADO
39.55,29.52,31.97,33.53,38.89,40.38,34.09,30.10,21.56,30.10,33.94,36.37,296108.21,4438.78,35869961.00,VEADO
30.29,31.79,29.94,29.09,33.25,33.28,36.55,33.10,36.47,34.93,33.51,37.80,318932.54,1141.58,39053861.00,VEADO
30.27,34.10,29.97,38.00,34.77,32.10,34.77,31.10,34.97,33.80,35.26,30.90,295736.22,2094.42,9073052.00,ALCE
33.13,33.25,33.54,38.32,32.22,35.02,37.08,33.35,34.65,31.73,29.38,28.33,832812.83,391.15,42777071.00,ALCE
27.67,38.27,29.35,35.22,30.79,22.95,37.36,30.81,41.54,38.78,33.29,33.97,785065.21,2876.40,40247442.00,ALCE
21.55,39.79,35.35,31.14,31.87,20.54,31.10,31.14,46.58,39.66,33.55,37.73,253129.72,10693.56,6910557.00,ALCE
20.34,42.60,35.45,32.70,31.36,14.74,32.62,31.06,48.30,42.66,31.93,36.23,374929.81,2760.65,8597552.00,ALCE
20.57,44.01,37.39,33.54,30.61,13.62,27.59,28.51,48.82,42.37,35.03,37.95,365973.85,2880.62,8718471.00,ALCE
24.17,38.10,29.30,36.44,32.46,23.92,36.68,30.82,43.37,37.99,34.01,32.74,689590.20,780.58,40238382.00,ALCE
30.58,35.38,32.45,32.17,33.58,29.97,33.91,32.42,35.84,34.65,33.65,35.41,846592.60,6445.46,71781534.00,ALCE
32.75,36.40,37.00,32.36,35.05,27.00,28.58,32.85,32.20,36.60,34.42,34.79,371219.35,5248.71,9073010.00,ALCE
27.48,39.18,37.17,35.11,34.17,23.27,31.28,31.29,38.34,37.55,31.55,33.61,1077444.42,2588.53,103411228.00,ALCE
```

Figura 17: Arquivo ARFF para os animais (configuração *TRACTS*.3.1)

Utilizou-se um filtro de pré-processamento de discretização supervisionada e então houve a mineração de dados. O resultado de aplicação da tarefa de classificação de dados no Weka gerou os dados como os exibidos na Figura 18. Nesse exemplo,

utilizou-se o algoritmo de classificação *SMO* do Weka (baseado em SVM), o qual produziu os melhores resultados para a classificação de trajetórias do método *TRACTS*.

```
=== Summary ===

Correctly Classified Instances         246               97.2332 %
Incorrectly Classified Instances         7                2.7668 %
Kappa statistic                          0.9552
Mean absolute error                      0.2301
Root mean squared error                  0.2858
Relative absolute error                 55.8055 %
Root relative squared error             62.9695 %
Total Number of Instances              253

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.922     0.015      0.94      0.922     0.931      0.935    VEADO
                 0.977     0.024      0.977     0.977     0.977      0.984    ALCE
                 1         0.006      0.987     1         0.993      0.997    GADO
Weighted Avg.    0.972     0.017      0.972     0.972     0.972      0.978

=== Confusion Matrix ===

   a   b   c   <-- classified as
  47   3   1 |   a = VEADO
   3 125   0 |   b = ALCE
   0   0  74 |   c = GADO
```

Figura 18: Mineração de dados de trajetórias de animais

Para esse exemplo, foi utilizado um alfabeto SAX de quatro letras, e a profundidade da matriz TSB utilizada foi de um. O tempo de geração do modelo de dados foi bastante rápido, produzindo o resultado em 0.17 segundos. Em uma das classes, a que identifica as trajetórias de Gado, a *TP Rate* foi de 100%, seguida pelas classe de Alce e por fim Veado, que foi a classe com maior dificuldade de classificação, mas mesmo assim com uma taxa de acerto de mais de 90%.

Com a combinação do tamanho do alfabeto do método SAX e da profundidade utilizada na matriz TSB, é possível utilizar diversos parâmetros diferentes para geração de arquivos de classificação “arff” pelo método *TRACTS*. A partir dessa combinação de parâmetros, utilizaram-se diversos algoritmos de mineração de dados disponíveis na ferramenta Weka para cada uma dessas combinações, gerando diversos modelos de dados para cada configuração. A quantidade de atributos utilizados e o melhor resultado para cada uma das configurações utilizadas no método *TRACTS* podem ser conferido na Tabela 5.

Cada uma das linhas da tabela onde o método *TRACTS* é citado representa um parâmetro de entrada para o método, onde o primeiro algarismo representa o tamanho do alfabeto SAX e o algarismo seguinte representa a profundidade de análise no método

TSB. Foram executados experimentos para alfabetos SAX de tamanhos 3, 4, 5 e 7, com profundidade da matriz TSB variando de 1 a 6.

Configuração do método	Total de atributos utilizados	Acurácia de classificação	Algoritmo utilizado	Tempo geração do modelo de classificação (segundos)
TRACTS.3.1	16	94,07%	Kstar	0,01
TRACTS.3.2	40	94,47%	Adaboost+BayesNet	0,6
TRACTS.3.3	112	95,26%	Bagging+SMO	3,6
TRACTS.3.4	328	95,65%	SMO	0,31
TRACTS.3.5	976	95,26%	SMO	0,42
TRACTS.3.6	2920	95,26%	SMO	0,69
TRACTS.4.1	**20**	**97,23%**	**SMO**	**0,17**
TRACTS.4.2	68	95,26%	Bagging+RandomForest	2,8
TRACTS.4.3	260	96,05%	Bagging+SMO	5,5
TRACTS.4.4	1028	95,65%	Bagging+SMO	16,5
TRACTS.4.5	4100	92,89%	SMO	0,89
TRACTS.5.1	24	94,47%	Adaboost+J48	1,6
TRACTS.5.2	104	95,65%	Adaboost+NaiveBayes	1,6
TRACTS.5.3	504	96,05%	SMO	0,34
TRACTS.5.4	2504	95,26%	SMO	0,66
TRACTS.7.1	32	94,86%	Bagging+SMO	2,8
TRACTS.7.2	200	96,44%	Bagging+SMO	5,5
TRACTS.7.3	1376	95,65%	SMO	0,47
TRACTS.7.4	9608	94,07%	SMO	2,01
Média	-	95,24%	-	2,45

Tabela 5: Resultado geral do método *TRACTS* para os animais

Como é possível observar, o método teve boa resposta geral na classificação das trajetórias para todas as classificações. Os resultados de acurácia, de forma geral, mostraram-se relativamente próximos a média final, demonstrando uma boa estabilidade no resultado final alcançado independente dos parâmetros e métodos de classificação utilizados. O tempo de geração do modelo de classificação foi reduzido.

O resultado comparativo da aplicação do método *TRACTS*, juntamente com os do método *TraClass*, para as trajetórias de animais pode ser visto na Tabela 6.

Método	Acurácia	Algoritmo de Classificação Utilizado
TRACTS.4.1	97,23%	*SVM (SMO)*
TRACTS.7.2	96,44%	*Bagging+SVM (SMO)*
TRACTS.4.3	96,05%	*Bagging+SVM (SMO)*
TRACTS.5.3	95,65%	*SVM (SMO)*
TRACTS.3.4	95,65%	*SVM (SMO)*
TraClass RB-TB	83,30%	*SVM*
TraClass TB-Only	50,00%	*SVM*

Tabela 6: Resultado comparativo para as trajetórias de animais

Nas tabelas de resultados comparativos mostrados nesse trabalho, são exibidos os cinco melhores resultados obtidos com o método *TRACTS*. Os resultados dos dois métodos introduzidos no método *TraClass* são colocados como base de comparação.

TraClass RB-TB corresponde ao método *TraClass* aplicado com as duas opções de geração de características, *region-based* e *trajectory–based*. *TraClass TB-only* corresponde a execução do método *TraClass* apenas considerando a característica baseada na trajetória. Não realizamos a implementação do método *TraClass*. Nas tabelas com os resultados dos experimentos, no que se refere ao método *TraClass,* reproduzimos os valores constantes no artigo que descreve o método (Lee, Han, Gonzalez, & Li, 2008).

Apesar de apresentarmos nesse trabalho o tempo de execução do melhor algoritmo do método *TRACTS*, não foram considerados como base de comparação os valores de tempo de execução dos algoritmos, uma vez que não teríamos condições de replicar exatamente o mesmo ambiente computacional utilizado pelo método *TraClass*, além do que os tempos de execução do método *TraClass* não são informados no artigo correspondente.

Como pode ser observado na Tabela 6, o método *TRACTS* obteve resultados sensivelmente melhores que o *TraClass* nas diversas configurações de parâmetros experimentadas para a base de dados de animais.

4.2 Experimentação – Trajetórias de Barcos

Para o segundo experimento foi utilizada a base disponível em (Monterey Bay Aquarium Research Institute, 2001), cuja finalidade é desenvolver instrumentos marítimos e aprimorar métodos de pesquisa científica em águas profundas. Os dados foram coletados no ano de 2000 e são informações de trajetórias advindas de dois barcos: PointSur e PointLobos. O objetivo então é predizer da qual destes dois barcos uma determinada trajetória pertence.

Como os dados de entrada eram duas trajetórias contínuas, uma para cada um dos barcos, para evitar a geração de um modelo super-especializado, com somente duas amostras, as trajetórias dos barcos foram divididas em 404 sub-trajetórias diferentes, 202 para PointSur e 202 para PointLobos, de forma com que cada uma das sub-trajetórias ficasse exatamente com o mesmo número de pontos. As 404 sub-trajetórias foram divididas então em duas classes, que identificam cada um dos barcos: PointSur ou PointLobos.

Da mesma forma que na base de dados dos animais, as mesmas características foram extraídas formando séries temporais (Tabela 7), as *strings* SAX (Tabela 8) foram formadas e após transformadas em matrizes *TSB* (Tabela 9) e por fim a ferramenta Weka foi utilizada para a etapa de classificação.

Número de trajetórias	Número de pontos	Séries temporais geradas	Tempo de geração (segundos)
404	56622	1616	61,92

Tabela 7: Geração das séries temporais para os animais

Tamanho do alfabeto SAX	Tempo de geração da string SAX (segundos)
3	218,21
4	201,09
5	202,00
7	208,62

Tabela 8: Tempo de geração da *string* SAX para os barcos

Tamanho alfabeto SAX	Profundidade da matriz TSB	Tempo geração matriz TSB (segundos)
3	1	19,58
3	2	29,98
3	3	35,56
3	4	69,52
3	5	133,5
3	6	699,92
4	1	25,97
4	2	32,02
4	3	70,19
4	4	198,5
4	5	648,89
5	1	20,84
5	2	53,87
5	3	235,25
5	4	649,03
7	1	29,08
7	2	84,21
7	3	337,45
7	4	1097

Tabela 9: Tempo de geração das matrizes TSB para os barcos

A aplicação dos algoritmos de classificação da ferramenta Weka mostrou uma situação distinta nesse domínio. Como explicado na subseção 3.3, foram extraídas tanto características locais, quanto características globais das trajetórias. Como aparentemente os dois barcos possuíam rotinas muito específicas de seus itinerários, principalmente em relação ao tempo de permanência no mar, ao segmentar as trajetórias, foi possível observar a situação demonstrada na Figura 18.

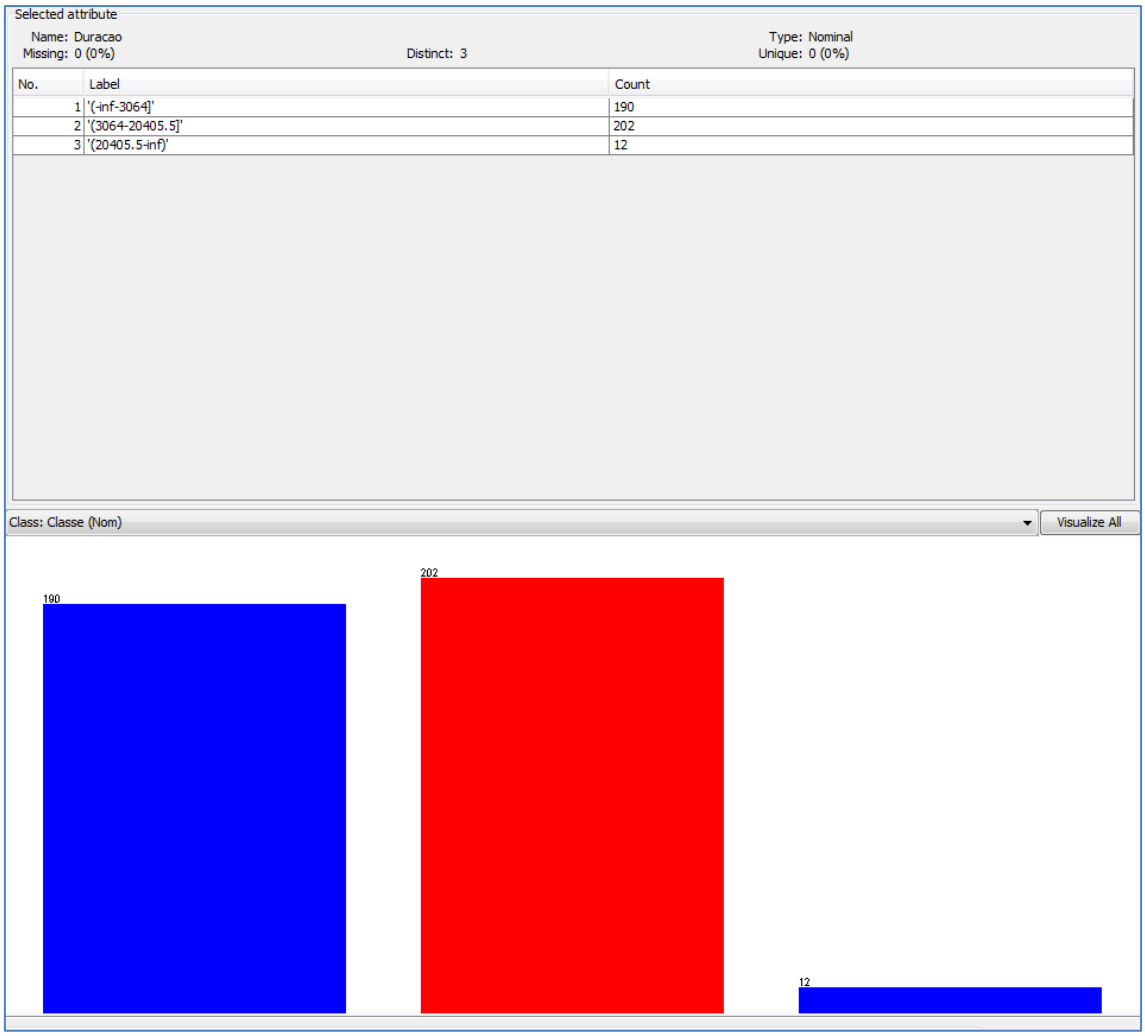


Figura 19: Atributo "Duração" para as trajetórias de barcos

É possível verificar na Figura 19 que o atributo “Duração” acaba discriminando completamente as trajetórias do domínio de barcos. Todas as 202 trajetórias do barco PointSur, em vermelho (coluna do meio), tem sua duração no intervalo entre 3064 e 20405,5 segundos, e as trajetórias do barco PointLobos estão abaixo ou acima desse intervalo (colunas em azul a direita e a esquerda). Isso torna o processo de classificação uma tarefa trivial, bastando o algoritmo de classificação basear-se somente nesse atributo para atingir 100% de acurácia no modelo gerado. De qualquer forma, isso demonstra o potencial de análise proporcionado pela análise combinada de características locais e globais.

Entretanto, para possibilitar uma análise dos demais atributos utilizados, que não são discriminatórios, em todas as análises e resultados mostrados nesse trabalho para as trajetórias dos barcos, não será utilizado o atributo “Duração”.

Dessa forma, um exemplo de resultado de aplicação da ferramenta Weka para este domínio produziu o resultado exibido na Figura 20.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         404               100      %
Incorrectly Classified Instances         0                 0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0.0007
Relative absolute error                  0.0098 %
Root relative squared error              0.1423 %
Total Number of Instances              404

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        PointLobos
                 1         0          1         1         1          1        PointSur
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

   a   b   <-- classified as
 202   0 |   a = PointLobos
   0 202 |   b = PointSur
```

Figura 20: Mineração dos dados de trajetórias dos barcos

Nessa execução foi utilizado o Weka com o algoritmo de mineração de dados *NaiveBayes* com alfabeto SAX 3 e profundidade 2. Como se pode observar na Figura 19, o método teve excelente resposta na classificação das trajetórias, chegando a 100% de acerto na classificação, mesmo havendo sido removido o atributo “Duração”. O resultado geral de aplicação do método TRACTS para as trajetórias dos barcos é mostrado na Tabela 10.

Configuração do método	Total de atributos utilizados	Acurácia de classificação	Algoritmo utilizado	Tempo geração do modelo de classificação (segundos)
TRACTS.3.1	15	99,76%	Adaboost+SimpleCart	9
TRACTS.3.2	**39**	**100,00%**	**NaiveBayes**	**0,01**
TRACTS.3.3	111	100,00%	Logistic	5,3
TRACTS.3.4	327	100,00%	Logistic	8,8
TRACTS.3.5	975	99,76%	NaiveBayes	0,06
TRACTS.3.6	2919	99,76%	NaiveBayes	0,06
TRACTS.4.1	19	99,26%	SMO	0,14
TRACTS.4.2	67	100,00%	NaiveBayes	0,01
TRACTS.4.3	259	100,00%	Logistic	0,84
TRACTS.4.4	1027	99,76%	NaiveBayes	0,04
TRACTS.4.5	4099	100,00%	Logistic	23
TRACTS.5.1	23	99,76%	SMO	0,16
TRACTS.5.2	103	100,00%	BayesNet	0,04
TRACTS.5.3	503	99,76%	NaiveBayes	0,03
TRACTS.5.4	2503	100,00%	Logistic	19
TRACTS.7.1	31	100,00%	NaiveBayes	0,01
TRACTS.7.2	199	100,00%	SMO	2
TRACTS.7.3	1375	100,00%	Logistic	18
TRACTS.7.4	9607	99,76%	NaiveBayes	0,04
Média	-	99,87%	-	4,55

Tabela 10: Resultado geral do método TRACTS para os barcos

A Tabela 10 nos mostra uma excelente média de acurácia nos resultados, com uma estabilidade bastante consistente em todas as parametrizações utilizadas. O tempo de execução do algoritmo de classificação também foi bastante rápido.

O resultado comparativo entre as cinco melhores execuções no método *TRACTS* e o método *TraClass* pode ser visualizado na Tabela 11.

Método	Acurácia	Algoritmo de Classificação
TRACTS.3.2	100,00%	*NaiveBayes*
TRACTS.4.2	100,00%	*NaiveBayes*
TRACTS.7.1	100,00%	*NaiveBayes*
TRACTS.5.2	100,00%	*BayesNet*
TRACTS.7.2	100,00%	*SMO*
TraClass RB-TB	98,20%	*SVM*
TraClass TB-Only	84,40%	*SVM*

Tabela 11: Resultado comparativo para as trajetórias de barcos

Como pode ser observado, o resultado nesse domínio também foi melhor que o do método *TraClass* em todas as configurações de parâmetros testadas, obtendo acurácia de 100% nas cinco melhores configurações, utilizadas para essa comparação.

4.3 Experimentação – Trajetórias de Furacões

Os dados de rastreamento de furações, disponível em (Unisys, 2009), foram coletados a partir de 1851 na região do Golfo do México. A Figura 21 mostra todas as rotas desse conjunto de dados em relação ao mapa dos Estados Unidos da América.

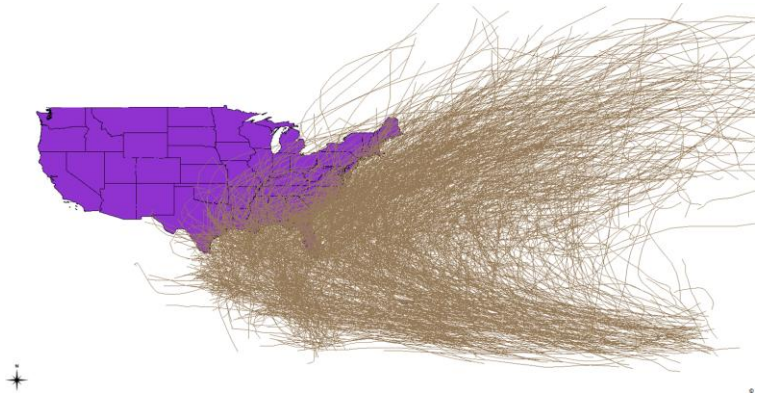


Figura 21: Trajetórias de furacões

Nessa base de dados, cada ponto contém a informação de força do seu respectivo furacão, que vai de -3 até +5, como pode ser observado na tabela 4. A partir do valor +0, é possível considerar o trajeto como um furacão com potencial de dano intenso, estabelecendo dessa forma um paralelo com a escala Fujita (F-Scale) que vai de F0 (mais fraco) até o F5 (mais forte).

De forma a possibilitar o processo de classificação, o atributo classe de cada trajetória desse domínio é a força máxima que o furacão atingiu ao longo de sua trajetória inteira. No conjunto de dados utilizado, esse atributo é categórico em formato descritivo. Para facilitar a classificação pela maior força, foi atribuído um valor para cada um dos valores existentes, da seguinte forma:

Classificação de Furacão	Valor (classe)
Onda tropical (*Tropical wave*)	-3
Depressão subtropical (*Subtropical depression*)	-2
Depressão extratropical (*Extratropical depression*)	-2
Depressão tropical (*Tropical depression*)	-2
Tempestade tropical (*Tropical storm*)	-1
Tempestade subtropical (*Subtropical storm*)	-1
Tempestade extratropical (*Extratropical storm*)	-1
Tempestade subtropical-1 (*Subtropical storm-1*)	0
Tempestade extratropical-1 (*Extratropical storm-1*)	0
Furacão-1 (*Hurricane-1*)	1
Furacão-2 (*Hurricane-2*)	2
Furacão-3 (*Hurricane-3*)	3
Furacão-4 (*Hurricane-4*)	4
Furacão-5 (*Hurricane-5*)	5

Tabela 12: Classes dos furacões

Alguns dos valores de classe foram agrupados nos mesmos valores de força devido a serem eventos climáticos de intensidade semelhante, que tem como diferença principal o local de ocorrência (tropical, subtropical ou extratropical).

Os atributos gerados para a classificação são os mesmos atributos gerados na experimentação da seção 4.1, bem como foram executadas as etapas de caracterização, transformação e classificação. De forma a permitir uma comparação equivalente ao trabalho realizado por (Lee, Han, Gonzalez, & Li, 2008), foram consideradas somente as classes 2 e 3.

Número de trajetórias	Número de pontos	Séries temporais geradas	Tempo de geração (segundos)
1367	39777	5468	161,89

Tabela 13: Geração das séries temporais para os furacões

Tamanho do alfabeto SAX	Tempo de geração da string SAX (segundos)
3	395,12
4	483,11
5	447,48
7	433,95

Tabela 14: Tempo de geração da *string* SAX para os furacões

Tamanho alfabeto SAX	Profundidade da matriz TSB	Tempo geração matriz TSB (segundos)
3	1	4,12
3	2	5,13
3	3	7,04
3	4	19,31
3	5	124,81
3	6	869,79
4	1	4,33
4	2	6,74
4	3	18,92
4	4	105,31
4	5	410,36
5	1	5,13
5	2	10,67
5	3	35,13
5	4	497,03
7	1	6,95
7	2	18,28

7	3	92,99
7	4	1132,36

Tabela 15: Tempo de geração das matrizes TSB para os furacões

Da mesma forma como nos domínios anteriores, após a geração da matriz TSB foi realizada a etapa de classificação com a ferramenta Weka. É possível observar um exemplo de execução na Figura 22. Nessa execução, foi utilizada a configuração do método *TRACTS* com tamanho de alfabeto SAX 4 e profundidade da matriz TSB 3. O algoritmo de classificação utilizado foi o AODEsr, disponível na ferramenta Weka.

O tempo de execução do processo de mineração também foi bastante rápido, levando cerca de três segundos na mesma máquina onde foram realizados os experimentos anteriores. O resultado da melhor execução do método *TRACTS* pode ser visualizado na Figura 22.

```
=== Summary ===

Correctly Classified Instances         270               71.2401 %
Incorrectly Classified Instances       109               28.7599 %
Kappa statistic                          0.4218
Mean absolute error                      0.3761
Root mean squared error                  0.4531
Relative absolute error                 76.4905 %
Root relative squared error             91.3768 %
Total Number of Instances              379

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.706     0.279      0.766     0.706     0.735      0.746    2
                 0.721     0.294      0.654     0.721     0.686      0.746    3
Weighted Avg.    0.712     0.286      0.717     0.712     0.713      0.746

=== Confusion Matrix ===

   a   b   <-- classified as
 151  63 |   a = 2
  46 119 |   b = 3
```

Figura 22: Mineração dados de trajetórias de furacões

O tempo de execução do algoritmo, assim como nos domínios anteriores, foi bastante rápido, levando menos de um segundo para concluir a tarefa de classificação. Entretanto, como é possível observar, o método *TRACTS* teve uma dificuldade bastante elevada para classificar os furacões de ambas as classes.

O resultado geral de classificação do método *TRACTS* para o domínio dos furacões pode ser observado na Tabela 16.

Configuração do método	Total de atributos utilizados	Acurácia de classificação	Algoritmo utilizado	Tempo geração do modelo de classificação (segundos)
TRACTS.3.1	16	66,22%	BayesNet	0,02
TRACTS.3.2	40	66,75%	Bagging+BayesNet	0,01
TRACTS.3.3	112	68,34%	Bagging+BayesNet	0,3
TRACTS.3.4	328	67,43%	RandomTree	0,2
TRACTS.3.5	976	65,34%	NaiveBayes	0,4
TRACTS.3.6	2920	65,25%	NaiveBayes	0,5
TRACTS.4.1	20	65,44%	NaiveBayes	0,5
TRACTS.4.2	68	66,75%	NaiveBayes	0,1
TRACTS.4.3	**260**	**71,24%**	**AODEsr**	**0,44**
TRACTS.4.4	1028	69,31%	Bagging+RandomForest	35,3
TRACTS.4.5	4100	69,31%	NaiveBayes	0,7
TRACTS.5.1	24	65,96%	NaiveBayes	0,1
TRACTS.5.2	104	67,55%	NaiveBayes	0,2
TRACTS.5.3	504	69,13%	Logistic	0,16
TRACTS.5.4	2504	66,93%	NaiveBayes	0,6
TRACTS.7.1	32	65,44%	NaiveBayes	0,1
TRACTS.7.2	200	68,60%	BayesNet	0,6
TRACTS.7.3	1376	66,23%	NaiveBayes	0,4
TRACTS.7.4	9608	65,87%	BayesNet	21,9
Média	-	67,22%	-	3,29

Tabela 16: Resultado geral do método TRACTS para os furacões

Realizando a comparação dos cinco melhores resultados obtidos pelo método *TRACTS* com os dois resultados do método *TraClass*, é possível obter os valores exibidos na Tabela 5.

Método	Acurácia	Algoritmo de Classificação
TraClass RB-TB	73,10%	*SVM*
TRACTS.4.3	71,24%	*AODEsr*
TRACTS.4.4	69,31%	*Bagging+RandomForest*
TRACTS.4.5	69,31%	*NaiveBayes*
TRACTS.5.3	69,13%	*Logistic*
TRACTS.7.2	68,60%	*BayesNet*
TraClass TB-Only	65,40%	*SVM*

Tabela 17: Resultado comparativo para as trajetórias de furacões

Apesar do melhor resultado do método *TRACTS* ter ficado próximo ao melhor resultado do método *TraClass*, é possível observar que a acurácia geral do método *TRACTS* no domínio dos furacões foi bem menor do que nos experimentos anteriores, em grande parte porque a trajetória dos furacões não está fortemente correlacionada com a sua intensidade. O método que obteve os melhores resultados foi o *TraClass RB-TB*, pois analisa a região de ocorrência das trajetórias (*Region Based*), já que furacões

mais fortes tendem a atingir locais mais afastados do que furacões mais fracos. No método *TRACTS*, os melhores resultados do método foram a partir de análises mais profundas do alfabeto SAX (com três ou quatro *substrings* sendo avaliadas por vez).

Uma das principais causas do baixo valor de acurácia do método *TRACTS* nesse domínio em relação aos outros dois domínios deve-se ao grau de aleatoriedade das trajetórias climáticas. Como não se observa facilmente um padrão de comportamento, a extração das características não é tão bem sucedida na tarefa de extrair comportamentos tão significativos como no domínio dos animais e barcos.

Outra possível causa da dificuldade de classificação nesse domínio foi estabelecer o atributo classe como a força máxima atingida pelo furacão, já que a força máxima pode ter sido atingida somente em um breve período de tempo, enquanto o método considera a trajetória inteira. Isto acaba afetando a qualidade dos dados obtidos pela etapa de extração de características, já que estes dados acabam sendo distorcidos pelos diferentes estágios que um furacão atinge durante uma trajetória.

Uma das ideias para se contornar este problema, seria a segmentação dos furacões através de suas diferentes categorias obtidas no decorrer da trajetória. Fazendo assim com que as características extraídas não sejam distorcidas pela mudança da classe no decorrer de uma trajetória. Entretanto, como o método *TraClass*, usado na comparação, não tratou o conjunto de dados dessa forma, essa formatação não foi considerada, já que não teríamos subsídios para indicar que uma eventual melhora na performance de classificação seria relevante ou não, devido a falta de dados para comparação.

4.4 Resultado Comparativo

Considerando os melhores resultados de cada uma das execuções dos experimentos, é possível compilar um resultado geral para os experimentos realizados, através da média de acurácia dos resultados. A tabela 6 mostra a performance do método *TRACTS* face ao trabalho desenvolvido em (Lee, Han, Gonzalez, & Li, 2008), que criou o método *TraClass*, em cada uma de suas configurações:

Método	Acurácia			
	Animais	Barcos	Furacões	Média
TRACTS	97,23%	100,00%	71,24%	89,49%
TraClass RB-TB	83,30%	98,20%	73,10%	84,87%
TraClass TB-Only	50,00%	84,40%	65,40%	66,60%

Tabela 18: Comparação dos resultados

Dessa forma, é possível observar que mesmo com a dificuldade de classificação no domínio de dados dos furacões, quando é considerada a regularidade de acurácia nos diversos experimentos realizados, o método *TRACTS* apresenta resultado superior. De forma geral, é possível observar que o método demonstra resultados semelhantes em domínios caóticos ou com comportamento aleatórios, enquanto demonstra uma clara superioridade nos resultados em domínios com comportamentos cognitivos.

Esse resultado permite afirmar a possibilidade de utilização do método *TRACTS*, com bons resultados, em muitos domínios de aplicação.

5 CONCLUSÃO

Nesse trabalho foi apresentado o método *TRACTS*, um novo método para classificação de trajetórias, utilizando séries temporais e técnicas de mineração de dados clássicas para esse tipo de representação, possibilitando a criação de modelos de classificação com boa acurácia.

O método *TRACTS* demonstrou-se eficaz na tarefa de criar um modelo de classificação com boa acurácia, no geral superior a um método proposto para o mesmo fim, sem comprometimento da propriedade de independência de domínio considerado.

O domínio com maior índice de erros foi um composto por trajetórias formadas por elementos caóticos da natureza, não possuindo qualquer tipo de raciocínio para tomada de decisão quanto ao seu comportamento. Isso, juntamente com a dificuldade de predizer uma característica momentânea (força máxima) a partir da trajetória inteira, acabou dificultando muito a tarefa de classificação.

Entretanto, quando considerados domínios compostos por trajetórias obtidas a partir de seres vivos (ou de objetos que tenham um comportamento que segue alguma lógica de movimentação), onde existe uma decisão para cada ação de movimentação realizada, pode-se observar uma melhora significativa na acurácia de predição, chegando a quase acurácia total em alguns domínios, como foi possível observar, por exemplo, no conjunto de dados dos barcos.

Na comparação com o trabalho desenvolvido por (Lee, Han, Gonzalez, & Li, 2008), considerando a técnica *Trajectory Based*, que realiza uma classificação semelhante ao método *TRACTS*, apesar do exemplo dos furacões ter sido somente um pouco mais eficaz, nos dois outros domínios houve uma clara melhora de acurácia de predição, indicando uma das grandes potencialidades do método *TRACTS*, que é a detecção de comportamento a partir de trajetórias. Quando comparado a técnica de *Region Based*, apesar do resultado ter sido um pouco menos eficaz nas trajetórias formadas por furacões, o resultado foi superior no domínio dos barcos e claramente superior no conjunto de trajetórias de animais.

Deve ser observado, entretanto, que o método *TRACTS* não utilizou nenhuma outra técnica além da análise pura das características das trajetórias, mantendo a independência da região ou objetos ao redor delas. Entretanto, o método proposto pelo *TraClass* utilizou uma técnica mais apurada para pré particionamento das regiões da trajetória, necessitando analisar outros elementos do domínio, ao invés de somente realizar a análise das trajetórias dos conjuntos de dados.

5.1 Contribuições

Como contribuições principais desse trabalho, é possível citar:

- A construção de um método eficaz de classificação de trajetórias, utilizando técnicas de construção de séries temporais, e discretização das mesmas;
- Utilização de algoritmos clássicos de classificação para realizar a construção de modelos de classificação;
- Manter uma boa independência quanto ao domínio considerado no conjunto de dados; possibilitando uma análise pura das características da trajetória.

5.2 Artigo Publicado

Essa dissertação resultou no seguinte artigo publicado:

Santos, I.P., & Alvares, L.O. (2011). TRACTS: Um método para a classificação de trajetórias de objetos móveis usando séries temporais. *8º Encontro Nacional de Inteligencia Artificial (ENIA) – CSBC, Proceedings* (pp. 800-808). Natal, Brasil: Springer.

5.3 Trabalhos Futuros

Será realizando um estudo mais aprofundado, considerando outros domínios de aplicação, sobre a influência do tamanho do alfabeto utilizado no método SAX e do tamanho da *substring* avaliada no método TSB na acurácia de classificação. Como o método *TRACTS* prevê a utilização de qualquer característica geométrica da trajetória como característica global ou local, é possível prospectar novas características para compor os atributos de classificação.

Também será investigado o uso de outros métodos de tratamento de *strings*, além do TSB, que também possibilitem a detecção de padrões interessantes a partir da *string* gerada pelo método SAX. Entre esses métodos podemos citar o Hot SAX (Keogh & Lin, 2005), contrastes de conjunto (Lin & Keog, 2006), clusterização (Ratanamahatana, Keogh, Bagnall, & Lonardi, 2005) e medidas de distância baseadas em compressão (Keogh, Lonardi, & Ratanamahatana, 2004).

REFERÊNCIAS

ASHBROOK, D.; STARNER, T. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. **Personal and Ubiquitous Computing Journal**, p. 275 - 286. 2003.

CHAKRABARTI, K.; KEOGH, E.; MEHROTRA, S.; PAZZANI, M. Locally adaptive dimensionality reduction for indexing large time series databases. **ACM Transactions on Database Systems**, p. 151–162, jun. 2002.

CHAN, K.-P.; & FU, A. W.-C. Efficient Time Series Matching by Wavelets. **ICDE 15th International Conference on Data Engineering, Proceedings**, Sydney, Australia, IEEE Computer Society, p. 126–133, 1999.

DE VRIES, G. K.; VAN HAGE, W. R.; VAN SOMEREN, M. Comparing Vessel Trajectories Using Geographical Domain Knowledge and Alignments. **IEEE International Conference on Data Mining Workshops (ICDMW), Proceedings**, Sydney, Australia, IEEE Computer Society, p. 209 - 216, 2010.

FALOUTSOS, C.; RANGANATHAN, M.; MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. **SIGMOD international conference on Management of data, Proceedings**, **Proceedings**, New York, NY, EUA, ACM, p. 419–429, 1994.

FAYYAD, U. M.; IRANI, K. B. Multi-interval discretization of continuousvalued attributes for classification learning. **Thirteenth International Joint Conference on Artificial Intelligence**, **Proceedings**, Chambéry, França, Morgan Kaufmann, p. 1022-1027, 1993.

FRANK, E.; HALL, M.; HOLMES, G.; KIRKBY, R.; PFAHRINGER, B. WEKA - A Machine Learning Workbench for Data Mining. In: **Collection of The Data Mining and Knowledge Discovery Handbook**, Springer, p. 1305-1314, 2005.

GARCÍA, J.; CONCHA, O. P.; MOLINA, J. M.; MIGUEL, G. Trajectory classification based on machine-learning techniques over tracking data. **9th International Conference on Information Fusion**, **Proceedings**, Florence, Italia, p. 1-8, 2006.

GEURTS, P. Pattern Extraction for Time Series Classification. **European Conference on Principles of Data Mining and Knowledge Discovery PKDD, Proceedings**, London, UK, Springer, p. 115–127, 2001.

GUDMUNDSSON, J.; KREVELD, M. V.; SPECKMANN, B. Efficient Detection of Patterns in 2D Trajectories. **GeoInformatica**, p. 195-215, 2007.

HAN, J.; LEE, J.-G.; LI, X.; CHENG, H. Mining Discriminative Patterns for Classifying Trajectories on Road Networks. **IEEE Transactions on Knowledge and Data Engineering**, p. 713 - 726, 2011.

HARIHARAN, R.; TOYAMA, K.. Project Lachesis: Parsing and Modeling Location Histories. **3rd International Conference on Geographic Information Science**, **Proceedings**, Adelphi, EUA, Springer, p. 106–124, 2004.

KAMBER, M.; HAN, J. *Data Mining: Concepts and Techniques.* San Francisco, CA, EUA, Morgan Kaufmann, 2006.

KEOGH, E.; LIN, J. Hot sax: Efficiently finding the most unusual time series subsequence. **5th IEEE International Conference on Data Mining**, **Proceedings**, Houston, TX, EUA, Computer Society, p. 226-233, 2005.

KEOGH, E.; CHAKRABARTI, K.; PAZZANI, M.; MEHROTRA, S. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. **Knowledge Inform Systems**, **Proceedings**, p. 263–286, 2000.

KEOGH, E.; LONARDI, S.; RATANAMAHATANA, C. A. Towards parameter-free data mining. **10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, **Proceedings**, Seattle, WA, EUA,: ACM, p. 206-215, 2004.

KUMAR, N.; LOLLA, V. N.; KEOGH, E.; LONARDI, S.; RATANAMAHATANA, C. A.; WEI, L. Time-series Bitmaps: A Practical Visualization Tool for working with Large Time Series Databases. **5th SIAM International Conference on Data Mining - SDM'05**, **Proceedings**, Newport Beach, CA, EUA, SIAM, p. 531-535, 2005.

LARSEN, R. J.; MARX, M. L. *An introduction to mathematical statistics and its applications.* Englewood Cliffs, N.J., EUA, Prentice-Hall International, 1986.

LAUBE, P.; KREVELD, M. V.; IMFELD, S. Finding REMO - Detecting Relative Motion Patterns in Geospatial Lifelines. **12th ACM international workshop on Geographic information systems**, **Proceedings**, Washington, DC, EUA, ACM, p. 250-257, 2004.

LEE, J. Y.; HOFF, W. Activity Identification Utilizing Data Mining Techniques. **IEEE Workshop on Motion and Video Computing**, **Proceedings**, Austin, Texas, EUA, IEEE Computer Society, p. 12, 2007.

LEE, J.-G.; HAN, J.; GONZALEZ, H.; LI, X. TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. **VLDB Endowment**, Auckland, Nova Zelândia, VLDB Endowment, p. 1081-1094, 2008.

LIAO, L.; PATTERSON, D.; FOX, D.; KAUTZ, H. Building Personal Maps from GPS Data. **Annals of the New York Academy of Sciences**, p. 249 - 265, 2006.

LIN, J.; KEOG, E. Group SAX: Extending the Notion of Contrast Sets to Time Series and Multimedia Data. **10th European Conference on Principles and Practice of Knowledge Discovery in Databases**, **Proceedings**, Berlin, Alemanha, ECML PKDD, p. 284-296, 2006.

LIN, J.; KEOGH, E.; LONARDI, S.; CHIU, B. A symbolic representation of time series, with implications for streaming algorithms. **8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Proceedings**, San Diego, California, EUA, ACM, p. 2-11, 2003.

LIN, J.; KEOGH, E.; WEI, L.; LONARDI, S. Experiencing SAX: a novel symbolic representation of time series. **Data Mining and Knowledge Discovery**, p. 107-144, Out. 2007).

LONARDI, S. *Global detectors of unusual words: design implementation and applications to pattern discovery in biosequences.* Department of Computer Sciences, Purdue University, 2001.

MONTEREY BAY AQUARIUM RESEARCH INSTITUTE. **Monterey Bay Aquarium Research Institute**. Disponível em MUSE Project: http://www.mbari.org/MUSE/platforms/ships.htm. Acesso em 16 de Julho de 2010.

PACIFIC NORTHWEST RESEARCH STATION. **US Forest Service**. Disponível em The Starkey Project: http://www.fs.fed.us/pnw/starkey/data/tables/. Acesso em 16 de Julho de 2010.

PALMA, A. T.; BOGORNY, V.; KUIJPERS, B.; ALVARES, L. O. A Clustering-based Approach for Discovering Interesting Places in Trajectories. **23rd Annual Symposium on Applied Computing**, **Proceedings,** Fortaleza, Ceara, Brasil, p. 863-868, 2008.

PANAGIOTAKIS, C.; PELEKIS, N.; KOPANAKIS, I. Trajectory Voting and Classification based on Spatiotemporal Similarity in Moving Object Databases. **8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII**, **Proceedings,** Lyon, France, Springer, p. 131-142, 2009.

QUINLAN, R. *C4.5: Programs for Machine Learning.* San Mateo, CA, EUA, Morgan Kaufmann Publishers, 1993.

RATANAMAHATANA, C.; KEOGH, E.; BAGNALL, A.; LONARDI, S. A Novel Bit Level Time Series Representation with Implications for Similarity Search and Clustering. **9th Pacific-Asia Conference Advances in Knowledge Discovery and Data Mining**, **Proceedings,** Hanoi Vietnam, Springer, p. 771-777, 2005.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining.* Boston: Addison-Wesley, 2006.

TIAKAS, E.; PAPADOPOULOS, A. N.; NANOPOULOS, A.; MANOLOPOULOS, Y. P.; STOJANOVIC, D.; KAJAN, S. D. Searching for similar trajectories in spatial networks. **Journal of Systems and Software**, p. 772-788, mai. 2009.

TIESYTE, D.; JENSEN, C. Similarity-based prediction of travel times for vehicles traveling on known routes. **16th ACM SIGSPATIAL International Symposium on Advances**, **Proceedings,** Irvine, California, EUA, ACM, p. 14, 2008.

UNISYS. **Atlantic Tropical Storm Tracking by Year**. Disponível em Unisys Weather: http://weather.unisys.com/hurricane/atlantic/. Acesso em 16 de Julho de 2010

YI, B.-K.; FALOUTSOS, C. Fast Time Sequence Indexing for Arbitrary Lp Norms. **26th international conference on very large databases**, **Proceedings**, Cairo, Egito: Morgan Kaufmann Publishers Inc., p. 385–394, 2000.

ZAIANE, O. R. **Principles of Knowledge Discovery in Databases**. Disponível em Departament of Computer Science - Universty of Alberta: http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/index.html. Acesso em 6 de Fevereiro de 2011

ZHENG, Y.; LIU, L.; WANG, L.; XIE, X. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. **17th international conference on World Wide Web, Proceedings,** Pequim, China, ACM, p. 247-256, 2008.

ZHOU, C.; BHATNAGAR, N.; SHEKHAR, S.; & TERVEEN, L. Mining Personally Important Places from GPS Tracks. **IEEE 23rd International Conference on Data Engineering Workshop**, **Proceedings,** Istambul, Turquia, IEEE Computer Society, p. 517-526, 2007.