



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA - INE
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

DANIEL HENRIQUE KOCK

UMA ABORDAGEM BASEADA EM ALGORITMO GENÉTICO PARA OTIMIZAÇÃO
DE SELEÇÃO DE SUBTRAJETÓRIAS RELEVANTES PARA CLASSIFICAÇÃO

Volume 1

Florianópolis

2019

DANIEL HENRIQUE KOCK

UMA ABORDAGEM BASEADA EM ALGORITMO GENÉTICO PARA OTIMIZAÇÃO
DE SELEÇÃO DE SUBTRAJETÓRIAS RELEVANTES PARA CLASSIFICAÇÃO

Volume 1

Trabalho de Conclusão de Curso apresentada ao curso de graduação em Sistemas de informação, como parte dos requisitos necessários à obtenção do título de Bacharel em Sistemas de Informação. Orientador: Luis Otavio Campos Alvares. Coorientadora: Vania Bogorny.

Florianópolis: 2019

Florianópolis

2019

DANIEL HENRIQUE KOCK

UMA ABORDAGEM BASEADA EM ALGORITMO GENÉTICO PARA OTIMIZAÇÃO
DE SELEÇÃO DE SUBTRAJETÓRIAS RELEVANTES PARA CLASSIFICAÇÃO

Volume 1

Trabalho de Conclusão de Curso apresentada ao curso de graduação em Sistemas de informação, como parte dos requisitos necessários à obtenção do título de Bacharel em Sistemas de Informação. Orientador: Luis Otavio Campos Alvares. Coorientadora: Vania Bogorny.

Florianópolis, 2019

BANCA EXAMINADORA

Luis Otavio Campos Alvares

Universidade Federal de Santa Catarina

Vania Bogorny

Universidade Federal de Santa Catarina

Rafael de Santiago

Universidade Federal de Santa Catarina

Camila Leite da Silva

Universidade Federal de Santa Catarina

AGRADECIMENTOS

Encerro esta etapa da graduação, tão importante na minha vida, e seria injusto não destacar pessoas a quem serei eternamente grato, e que foram inseparáveis do meu sucesso nesta fase.

Agradeço a minha mãe e meu pai, bases fundamentais da pessoa que sou hoje, e que são a razão do sucesso nesta e em todas as fases da minha vida, que sempre deram atenção especial a educação dos seus filhos, e sempre batalharam para dar a mim e minha irmã a melhor condição possível.

Deixo também meus agradecimentos a todos os meus amigos, pessoas maravilhosas que tenho a oportunidade de ter ao meu lado. Apesar de não citar o nome de todos aqui, quero que saibam que tem minha eterna gratidão e amizade, por me apoiarem nessa fase, compreendendo meus momentos de ausência, e me fortalecendo todos os dias com seu companheirismo.

Gostaria de deixar um agradecimento especial ao meu orientador Luis Otávio Campos Alvares, que me orientou em todos os momentos da construção desse trabalho sem medir esforços para tirar todas as minhas dúvidas; a professora Vania Bogorny, que acreditou em mim em todos os momentos e prestou toda a ajuda que eu necessitava. Por fim, agradeço a esta Universidade, seu corpo docente, direção e administração, que forneceram a estrutura necessária para que tudo isso fosse possível.

RESUMO

Com o atual avanço de tecnologias para a coleta de dados de trajetória, tais como GPS e smartphones, temos cada dia uma maior quantidade de dados relacionados à movimentação de pessoas e objetos, e devido ao crescente uso de informações neste contexto, é importante a análise deste conjunto de dados espaço-temporais a fim de agregar valor a estes dados. Neste trabalho, estudou-se um método de seleção de subtrajetórias relevantes chamado Movelets, que procura determinar subtrajetórias frequentes estritamente em uma classe específica. Em trabalhos anteriores, isso foi realizado para determinar a classe a que pertence cada trajetória através de uma busca exaustiva. O objetivo final foi propor e implementar um algoritmo genético que obtivesse acurácia semelhante a busca exaustiva utilizando um menor tempo de processamento. Os resultados obtidos mostram que a implementação proposta conseguiu apresentar bons resultados de acurácia, e de tempo de execução, superando o Movelets em acurácia para 18 dos 30 testes executados e em tempo de execução para grandes datasets, demonstrando que a implementação proposta foi muito bem sucedida.

Palavras-chave: Trajetórias; Classificação; Movelets; Inteligência Artificial; Algoritmo Genético

ABSTRACT

With the current advancement of trajectory data collection technologies such as GPS and smartphones, we have more and more data related to the movement of people and objects, and the analysis of this information is very important. One type of analysis is trajectory classification. The objective of this work is to study a new method called Movelets that discover relevant subtrajectories for trajectory classification, based on an exhaustive search, and implement and evaluate an alternative approach based on genetic algorithm. The goal was to propose and implement a genetic algorithm that would obtain similar accuracy to exhaustive search using a shorter processing time. The results obtained show that the proposed implementation was able to deliver good accuracy and runtime results, surpassing Movelets in accuracy for 18 out of 30 tests performed and runtime for large datasets, demonstrating that the proposed implementation was very successful.

Keywords: Trajectories; Classification; Movelets; Artificial Intelligence; Genetic Algorithm

LISTA DE ILUSTRAÇÕES

Figura 1 —	Abordagem geral para criação de um modelo de classificação	18
Figura 2 —	Representação de uma trajetória	19
Figura 3 —	Exemplos de trajetórias: (1) bruta e (2) e (3) semânticas	20
Figura 4 —	Exemplificação de Gene, Indivíduo e População	22
Figura 5 —	Exemplo de crossover em (1) um ponto e (2) dois pontos	24
Figura 6 —	Exemplo de mutação	25
Figura 7 —	Exemplo da linha de ordenação de um candidato a subtrajetória . .	28
Figura 8 —	Descrição dos datasets	30
Figura 9 —	Comparativo entre Movelets e outros métodos de classificação de trajetórias	32
Figura 10 —	Diagrama de diferença crítica entre os quatro algoritmos avaliados	35
Figura 11 —	Diagrama de execução do algoritmo proposto	37
Figura 12 —	Diagrama de classes do algoritmo	39
Figura 13 —	Exemplo de gene	40
Figura 14 —	Exemplo de geração do fitness	42
Figura 15 —	Exemplo de gráfico de fitness do melhor indivíduo x gerações	45
Gráfico 1 —	Acurácia e Tempo de execução variando o tamanho da população	48
Gráfico 2 —	Acurácia e Tempo de execução variando o tamanho do indivíduo. .	49
Gráfico 3 —	Acurácia e Tempo de execução variando a quantidade de gerações	50
Gráfico 4 —	Acurácia e Tempo de execução variando a taxa de mutação	50
Tabela 1 —	Avaliação de holdout utilizando Bayes	53
Tabela 2 —	Avaliação de holdout utilizando C4.5	53
Tabela 3 —	Avaliação de holdout utilizando SVM	53
Tabela 4 —	Tempo de execução da avaliação com holdout	54
Tabela 5 —	Avaliação de cross-validation utilizando Bayes	55
Tabela 6 —	Avaliação de cross-validation utilizando C4.5	55

Tabela 7 — Avaliação de cross-validation utilizando SVM.....	55
Tabela 8 — Tempo de execução da avaliação com cross-validation.....	56
Figura 16 — Gráfico de comparação da escalabilidade do Movelets e do AG utilizando datasets sintéticos.....	57

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
CV	Cross-validation
DBSCAN	Density-based spatial clustering of applications with noise
FS	Fast Shapelets
GPS	Sistema de Posicionamento Global
HO	Holdout
IA	Inteligência Artificial
ID	Identificador
KDD	descoberta de conhecimento em bancos de dados
LTS	Learning Timeseries Shapelets
ST	Shapelet Transform
SVM	Support Vector Machine

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVO GERAL	12
1.2	OBJETIVOS ESPECÍFICOS	12
1.3	JUSTIFICATIVA	13
1.4	METODOLOGIA	13
1.5	ESTRUTURA DO TRABALHO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	MINERAÇÃO DE DADOS	15
2.1.1	Classificação	16
2.2	TRAJETÓRIAS	18
2.3	ALGORITMO GENÉTICO	20
2.3.1	Conceitos básicos	21
2.3.2	Seleção	22
2.3.3	Reprodução (Crossover)	23
2.3.4	Mutação	24
3	TRABALHOS RELACIONADOS	26
3.1	MOVELETS: EXPLORING RELEVANT SUBTRAJECTORIES FOR ROBUST TRAJECTORY CLASSIFICATION	26
3.2	GENDIS: GENETIC DISCOVERY OF SHAPELETS	32
4	MÉTODO PROPOSTO	36
4.1	ESTRUTURA DO ALGORITMO	36
4.2	IMPLEMENTAÇÃO DO ALGORITMO	38
4.2.1	Diagrama de classes	38
4.2.2	Inicialização do AG	41
4.2.3	Fitness	41
4.2.4	Elitismo	43
4.2.5	Seleção	43
		43

4.2.6	Reprodução	43
4.2.7	Mutação	43
4.2.8	Critério de Parada	44
4.2.9	Visualização da evolução do melhor indivíduo	44
5	EXPERIMENTOS	46
5.1	VARIAÇÃO DA ACURÁCIA E TEMPO DE EXECUÇÃO EM FUNÇÃO DOS PARÂMETROS	46
5.1.1	Tamanho da população	47
5.1.2	Tamanho dos indivíduos	48
5.1.3	Gerações	49
5.1.4	Taxa de mutação	50
5.1.5	Definição dos parâmetros utilizados nas comparações	51
5.2	COMPARATIVO COM OUTROS MÉTODOS	52
5.2.1	Holdout	52
5.2.2	Cross-validation	54
5.3	ANÁLISE DE ESCALABILIDADE	56
6	CONCLUSÕES E TRABALHOS FUTUROS	59
	REFERÊNCIAS	61
	APÊNDICE A — CÓDIGO-FONTE	63
	APÊNDICE B — ARTIGO	64

1 INTRODUÇÃO

Trajетórias são formadas por sequências de pontos registrados para cada indivíduo correspondentes a sua localização em determinado momento do tempo (BOGORNY, 2012). Os pontos representam informações a respeito do objeto móvel, e normalmente são representados por coordenadas geográficas, formando o que é conhecido como trajetória bruta.

Os dados de movimento de objetos móveis só serão realmente úteis se forem analisados e essa análise resultar em conhecimento. Para a análise de trajetórias, uma das formas mais utilizadas é a classificação, que corresponde a identificar a classe de trajetórias. A classe pode ser o meio de transporte utilizado na trajetória, como por exemplo, ônibus, carro ou táxi, a identificação do usuário de uma trajetória de redes sociais, a qual tipo de animal corresponde uma trajetória, etc (BOGORNY, 2012).

Um dos métodos de classificação de trajetórias existente na literatura é o método Movelets (FERRERO et al., 2018), que faz uso da comparação exaustiva de subtrajетórias para obter as partes da trajetória que melhor identificam a classe em comparação com as outras classes. Esse método se mostrou muito eficiente para a classificação de dados de trajetória, superando outros métodos utilizados para a mesma finalidade. Porém sua implementação exige um processamento computacional muito grande, tornando praticamente inviável a sua execução em grandes conjuntos de dados.

Para resolver este problema, uma possível solução seria a utilização de meta-heurísticas para a otimização da busca destas subtrajетórias relevantes. O objetivo deste trabalho é a análise e implementação de uma destas técnicas de modo a reduzir o consumo de recursos computacionais em relação ao método Movelets, e analisar os resultados obtidos em comparação a implementação atual do método.

A abordagem escolhida para a resolução do problema é a técnica de Algoritmo Genético, um método que busca encontrar a melhor solução para os

problemas, utilizado um processo de busca iterativa da melhor solução para o problema em questão, que parte de uma população inicial e tende a manter os melhores resultados para as populações posteriores. (FERNANDES, 2005).

O trabalho de Vandewiele, Ongena e De Turck (2019) por exemplo, utiliza algoritmos genéticos para encontrar shapelets em séries temporais, que é um problema similar ao de encontrar as melhores subtrajetórias para a classificação de trajetórias. Os resultados mostraram que a solução proposta apresenta uma complexidade computacional inferior quando comparada com métodos de busca que utilizam força bruta, e com resultados muito próximos do melhor método de busca comparado, o que demonstra a viabilidade do uso desta abordagem para a solução do problema em questão.

1.1 OBJETIVO GERAL

O método Movelets de seleção de subtrajetórias relevantes para a classificação de trajetórias, proposto em 2018 (FERRERO et al.), tem uma complexidade de tempo muito alta, cúbica em relação ao tamanho das trajetórias, o que o torna inviável para grandes volumes de dados. O objetivo geral deste trabalho é propor um método alternativo para a seleção de subtrajetórias relevantes utilizando algoritmos genéticos.

1.2 OBJETIVOS ESPECÍFICOS

Os principais objetivos específicos deste trabalho são:

- Modelar o problema de forma adequada ao uso de Algoritmo Genético.
- Definir os principais componentes de um algoritmo genético para o problema em questão: a função objetivo, o indivíduo, a população inicial, a seleção e a reprodução;
- Testar e parametrizar a execução de modo com que ela seja compatível

com qualquer dataset analisado;

- Avaliar os resultados e tempo de execução afim de comparar com os resultados do método proposto com o Movelets;

1.3 JUSTIFICATIVA

Segundo Ferrero et al. (2018), o método Movelets é um dos melhores métodos para classificação de trajetórias, ele executa a comparação de todas as subtrajetórias possíveis para encontrar as melhores subtrajetórias, fazendo com que seu custo de tempo de execução seja igual a $O(n^2 \times m^3)$.

O foco deste trabalho é reduzir o tempo de execução do método Movelets utilizando o conceito de algoritmo genético, sem perder a boa acurácia nos resultados.

1.4 METODOLOGIA

Para este trabalho a primeira etapa foi o estudo de conceitos básicos sobre trajetórias de objetos móveis e o estudo do método Movelets, além dos conceitos sobre classificação de dados em geral e de classificação de trajetórias em particular.

A segunda etapa consistiu no estudo geral de algoritmos genéticos, seguido da modelagem do problema em questão (a escolha de subtrajetórias relevantes para a classificação) sob a forma de um algoritmo genético. Também nesta etapa foi feita a definição de seus principais componentes: o indivíduo, a função objetivo, o método de seleção, de reprodução e de mutação.

A terceira etapa corresponde à implementação do método, testes preliminares para a definição do melhor conjunto de parâmetros, coleta de resultados utilizando exemplos diferentes de *datasets*.

A última etapa consistiu na comparação dos resultados obtidos pelo algoritmo

genético em relação ao método Movelets. As comparações foram feitas com base no tempo de processamento e uso de memória, quanto em relação a acurácia obtida, considerando vários conjuntos de dados.

1.5 ESTRUTURA DO TRABALHO

O presente trabalho se divide em cinco capítulos. No Capítulo 2 são apresentados conceitos básicos para o entendimento geral do contexto do trabalho e as soluções selecionadas para este trabalho, com suas devidas justificativas. No Capítulo 3, duas obras correlatas ao tema proposto são analisadas. A parte de implementação do trabalho está no Capítulo 4 e contempla a parte técnica, com o ferramental e tecnologias utilizadas, o plano de implementação e sua execução. O Capítulo 5 apresenta os experimentos realizados para validar o método proposto e comparativos com outros métodos com finalidade semelhante e finalmente o Capítulo 6 apresenta as conclusões do trabalho e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos e contextualizações necessárias para o entendimento do trabalho. As Seções 2.1 e 2.2 apresentam conceitos básicos sobre mineração de dados e sobre trajetórias, enquanto a Seção 2.3 apresenta uma visão geral sobre algoritmos genéticos e seus parâmetros de execução.

2.1 MINERAÇÃO DE DADOS

Cada vez mais temos instrumentos e sensores que coletam dados de temperatura, localização, clima, humor, etc., com o intuito de saber cada vez mais sobre tudo que está acontecendo ao nosso redor. Do aumento no volume de dados coletados, surgem volumosos repositórios de dados. De modo a não perder nenhuma informação, precisamos analisar estes dados de modo a identificar padrões e prever acontecimentos (BOGORNÝ, 2012).

A mineração de dados é a área que utiliza técnicas de análise para processamento de dados, descobrindo informações úteis e padrões novos provindos desses dados. A mineração de dados é parte integrante do processo geral de conversão de dados brutos em informações úteis, também conhecido como KDD. Esse processo consiste em uma série de etapas de transformação, que são o pré-processamento, mineração e pós-processamento dos dados (TAN; STEINBACH; KUMAR, 2006).

Pré-processamento: etapa onde um conjunto de dados é analisado com intuito de identificar os atributos deste conjunto que serão utilizados na mineração. Em seguida, os dados são processados e convertidos para um formato adequado para a análise. Podem ser utilizados métodos como fusão de dados de várias fontes, remoção de ruído e duplicatas dos dados, transformação de dados através de técnicas como generalização, normalização e discretização de atributos.

Mineração (data mining): etapa onde os dados são processados utilizando métodos de mineração, para classificação (ID3, C4.5, SVM, Naive Bayes, etc), agrupamento (K-means, DBSCAN, etc.), associação (Apriori, Closet, etc.), padrões sequenciais, etc.

Pós-processamento: etapa onde os padrões resultantes são analisados, visualizados e interpretados.

2.1.1 Classificação

A tarefa de classificação consiste em atribuir um valor de classe para um objeto (exemplo, instância, registro) a partir das características (*features*, atributos) desse exemplo. Em outras palavras, classificar um objeto é determinar com que grupo de entidades já classificadas anteriormente, este objeto apresenta mais semelhança. Exemplos da tarefa de classificação são classificar um tumor como benigno ou maligno, uma transação de cartão de crédito como legítima ou fraudulenta, um cogumelo como comestível ou venenoso. (BOGORNY, 2012)..

Os dados de entrada para uma tarefa de classificação são uma coleção de registros, onde cada registro é caracterizado por uma tupla (a,b) , onde a é um conjunto de atributos e b designa um rótulo de classe.

Os métodos de classificação são divididos em dois grupos principais: os que não geram um modelo (preguiçosos) e os que geram um modelo de classificação (espertos).

Um exemplo de modelo de classificação preguiçoso é o KNN, que compara o exemplo a ser classificado com cada uma das instâncias rotuladas e atribui ao exemplo a classe da instância mais parecida (BOGORNY, 2012). Para métodos de classificação espertos, podemos citar os métodos:

Baseados em árvore de decisão: os algoritmos que geram árvores de decisão, como o ID3 e C4.5, buscam dividir um problema complexo em sub-problemas mais simples de forma recursiva, de forma que os nodos da árvore

correspondem a atributos e os arcos são os valores possíveis dos atributos presentes nos exemplos e as folhas são as classes. Vários critérios podem ser usados para a escolha dos atributos a serem testados nos nodos na construção da árvore, como o ganho de informação e o *índice de Gini*.

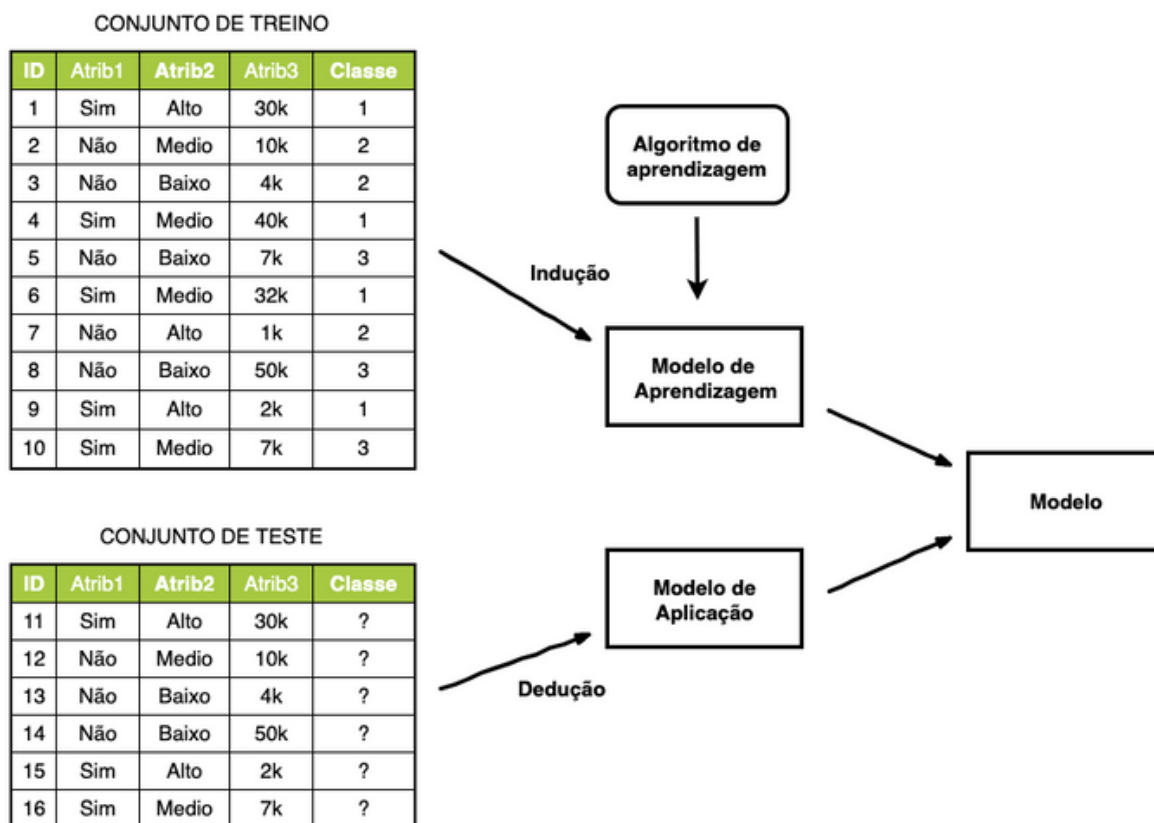
Naive Bayes: se baseia no teorema de Bayes. Ele estima a probabilidade de cada classe supondo que os atributos sejam condicionalmente independentes, desconsiderando completamente a correlação entre os atributos, daí advindo a denominação *naive* (ingênuo). (TAN; STEINBACH; KUMAR, 2006).

SVM: baseada na aprendizagem estatística, funciona muito bem com dados de alta dimensionalidade. Esta técnica traça um hiperplano de margem máxima do seu centro para a margem que separa as classes dos elementos do conjunto (TAN; STEINBACH; KUMAR, 2006).

Os métodos que geram um modelo de classificação são esquematizados na Figura 1, onde um conjunto de dados com os rótulos de classe (*training set*) é usado para induzir o modelo pelo algoritmo de classificação. Este modelo será usado para atribuir classe a instâncias para as quais queremos determinar a classe.

Para a validação do método é aplicado um conjunto de testes, que consiste em registros onde se conhece sua classe, porém ela é omitida ao modelo para que ele pressuponha as classes dos registros da coleção e depois possa se fazer uma comparação com a classe real, medindo assim a sua assertividade (TAN; STEINBACH; KUMAR, 2006).

Figura 1 - Abordagem geral para criação de um modelo de classificação



Fonte: Adaptado de Tan, Steinbach e Kumar (2006)

Algumas abordagens são utilizadas para a definição dos grupos de treino e de testes, sendo delas:

holdout: técnica onde é selecionada uma porcentagem do conjunto de dados para treino e outra parte para testes (TAN; STEINBACH; KUMAR, 2006).

cross-validation: técnica onde todos os registros são usados tanto para treino quanto para testes, de modo que o *dataset* é dividido em grupos, e a execução do treino/teste é feita várias vezes selecionando um conjunto como teste e os outros conjuntos como treino (TAN; STEINBACH; KUMAR, 2006).

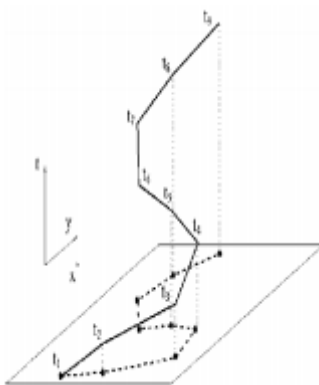
2.2 TRAJETÓRIAS

Com a evolução e propagação de tecnologias de captura de dados de

localização, como o GPS, a cada dia que passa, temos *Terabytes* de dados sendo coletados sobre o trajeto de veículos, animais e pessoas. Esses dados correspondem a uma sequência de pontos ordenados pelo tempo, formando a trajetória de um objeto móvel (BOGORNY, 2012), como exemplificado na Figura 2.

Para a formação de uma trajetória, o dispositivo coletor gera pontos com as coordenadas geográficas em um determinado momento, permitindo dessa forma que além da localização do indivíduo no espaço, seja possível também determinar outras medidas como a velocidade e a direção que o indivíduo rastreado estava se locomovendo.

Figura 2 - Representação de uma trajetória



Fonte: Giannotti e Pedreschi (2008)

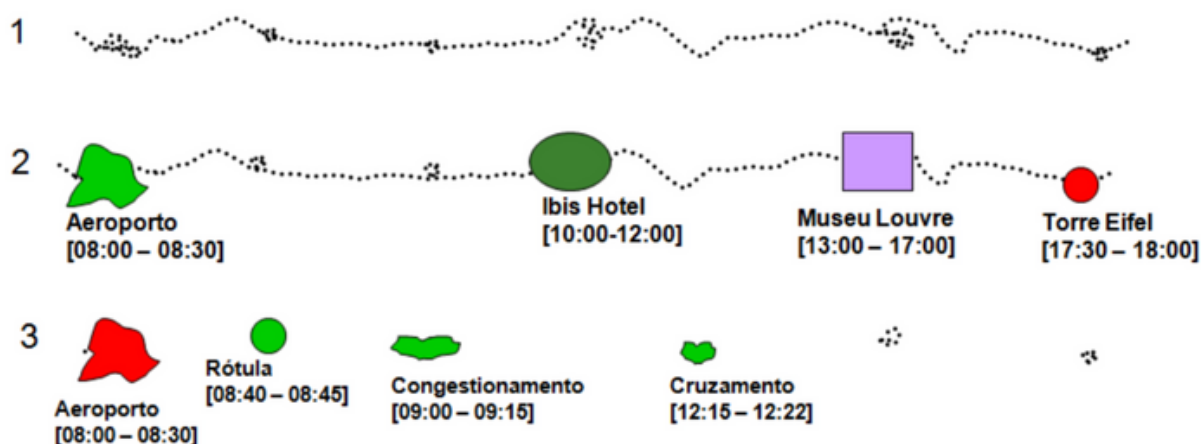
As trajetórias podem ser divididas em dois grupos, as trajetórias brutas e trajetórias semânticas (ou multi-aspecto), que são definidos de forma que:

Trajétória bruta: diz respeito a trajetórias que possuem somente informações espaço-temporais. Tratam-se de um conjunto de pontos compostos por um ID, coordenadas geográficas (x, y) , correspondentes à localização espacial do objeto no instante de tempo (t) .

Trajétória semântica: correspondem a trajetórias brutas enriquecidas com dados provindos de outras fontes, com o intuito de trazer mais informações a estes dados, tornando possível colher mais informações a respeito dos acontecimentos,

como o que o indivíduo fez, onde ele parou, que rotas tomou, etc.

Figura 3 - Exemplos de trajetórias: (1) bruta e (2) e (3) semânticas



Fonte: BOGORNÝ (2012)

Com esses dados, é possível tomar várias conclusões a respeito do indivíduo analisado, como padrões de locais que frequenta, qual meio de transporte utiliza regularmente, onde trabalha, entre outras inúmeras informações que são possíveis de obter agregando dados de outras fontes aos dados da trajetória (BOGORNÝ, 2012).

2.3 ALGORITMO GENÉTICO

A teoria evolucionista é um conceito da biologia que trata da alteração da carga genética dos seres vivos no decorrer de gerações, e pode ser definida como o processo de variação e adaptação de populações por meio da reprodução, onde a seleção natural faz com que somente os indivíduos mais propícios sobrevivam e deem continuidade a espécie (RIDLEY, 2006). Neste cenário, também deve-se considerar a possibilidade de mutações nestes indivíduos, por meio de alterações genéticas ocasionadas por inúmeros fatores.

Com base nesse conceito de evolução Darwiniano, algoritmos genéticos

buscam simular esse processo evolutivo com o intuito de executar a busca de uma solução para o problema proposto.

Outra característica de algoritmos genéticos é o fato de apresentarem uma solução meta-heurística, que permite a resolução do problema para conjuntos de dados variados, sem a necessidade de conhecer o problema antes da execução.

2.3.1 Conceitos básicos

Esta seção apresenta os conceitos básicos necessários ao entendimento de algoritmos genéticos: gene, indivíduo, população e geração. Esses conceitos são exemplificados na Figura 4.

Gene: Para os Algoritmos Genéticos (AGs) a estrutura mais básica é o gene, que é responsável por controlar uma ou mais características de um indivíduo.

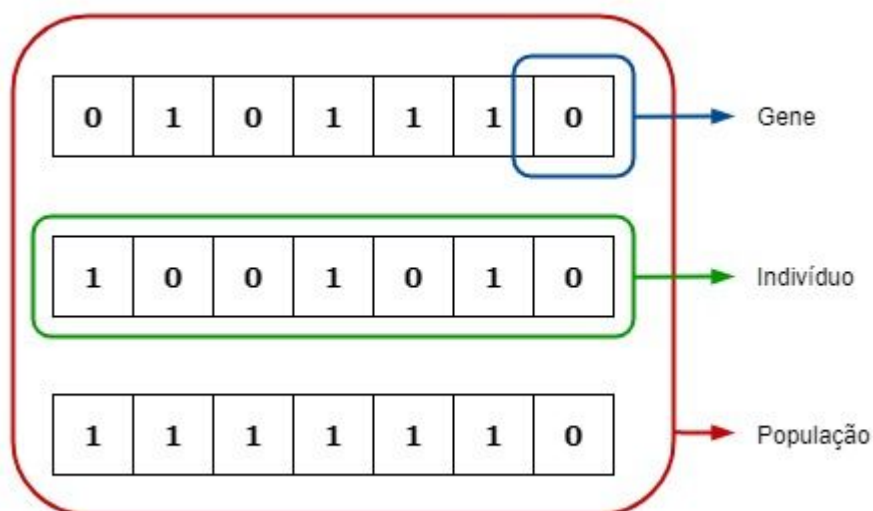
Indivíduo: é formado por um conjunto de genes que representam todas as características de um indivíduo dentro de uma população.

População: é um conjunto de indivíduos que estão dentro de um mesmo ecossistema.

Geração: corresponde a uma população em determinado momento do tempo.

Fitness: função de avaliação que define os indivíduos com maior aptidão para a resolução do problema.

Figura 4 - Exemplificação de Gene, Indivíduo e População



Fonte: O autor (2019)

2.3.2 Seleção

Na natureza, os membros de uma população e os membros de diferentes espécies competem entre si em busca da sobrevivência. Essa competição resulta das condições mais ou menos propícias para a continuidade da espécie no meio ambiente, também referida por Darwin como “a luta pela sobrevivência”. Ela é o pilar principal da seleção natural (RIDLEY, 2006). Essa luta pela sobrevivência em que somente os mais fortes sobrevivem e, portanto, tem maior chance de procriar e passar as suas características genéticas para as gerações seguintes corresponde à seleção nos AGs.

O método de seleção utilizado em AGs simula essa evolução, onde os indivíduos de uma população são ranqueados com o intuito de enfatizar os indivíduos mais aptos, na esperança de que seus filhos tenham uma aptidão maior.

Diversos métodos lidam com esse processo, tais como:

Elitismo: Consiste em um método que obriga que indivíduos melhores

avaliados sejam mantidos na próxima geração, impedindo que estes indivíduos diminuam sua nota de avaliação durante o cruzamento ou mutação. (MITCHELL, 1999).

Roleta: este método de seleção de indivíduos para a reprodução privilegia os indivíduos com função de avaliação mais alta, sem desprezar indivíduos com função de avaliação extremamente baixa, tendo como intuito manter nas populações características encontradas em indivíduos com baixa avaliação que podem não estar presentes em nenhum outro indivíduo (LINDEN, 2012).

Torneio : No método de seleção por torneio, seleciona-se de forma aleatória um conjunto de indivíduos que irão competir entre si com as suas avaliações. Aquele indivíduo que estiver participando no torneio e for melhor avaliado será o indivíduo selecionado para a reprodução. Uma diferença importante deste processo de seleção comparado com o método da roleta é que os melhores indivíduos terão como vantagem o seu alto valor de fitness. Isto significa que os mesmos não serão favorecidos na escolha de quais indivíduos irão participar do torneio, fazendo com que todos os cromossomos tenham chances iguais de participar (LINDEN, 2012).

2.3.3 Reprodução (*Crossover*)

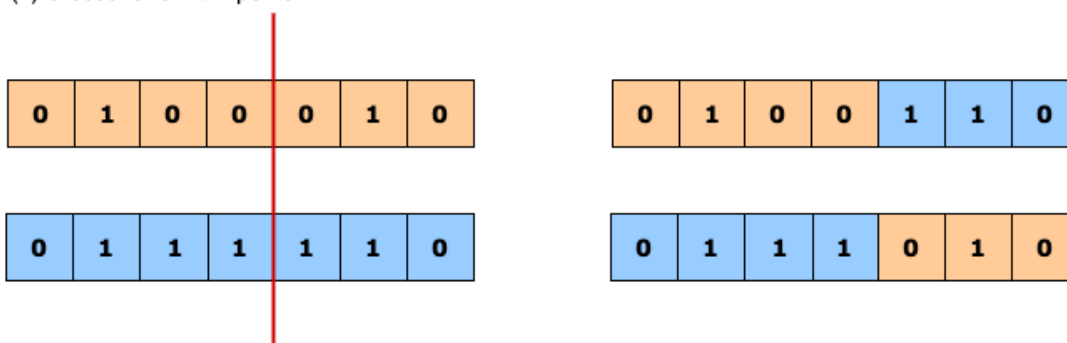
Na natureza, todos os seres vivos geram novos indivíduos para dar continuidade à espécie. Além disso, é através da reprodução que o indivíduo transmite suas características genéticas para seus descendentes. Nesse processo, os genes dos indivíduos mais aptos para a seleção tem maior chance de se manterem presentes no código genético dos sucessores, garantindo assim a evolução da espécie (RIDLEY, 2006).

A reprodução de um AG é dada por uma função de reprodução conhecida também por *crossover*, que simula a reprodução sexuada que acontece na natureza. Através desta função indivíduos trocam material genético, gerando assim novos indivíduos com as características combinadas de seus pais. (MITCHELL, 1999).

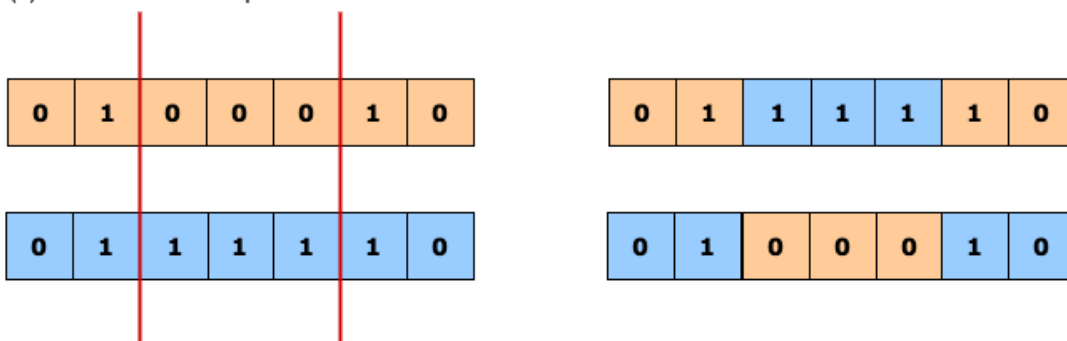
São duas as formas de se executar a reprodução com troca de material genético. No **crossover de um ponto** existe um ponto de divisão nos pais que determina onde e como será feita a troca do material genético dos pais para gerar os filhos, como mostrado na Figura 5 (1). No **crossover de dois pontos**, dois pontos de divisão são selecionados e a troca genética é realizada como mostrado na Figura 5 (2) (LINDEN, 2012).

Figura 5 - Exemplo de crossover em (1) um ponto e (2) dois pontos

(1) Crossover em um ponto



(2) Crossover em dois pontos



Fonte: O autor (2019)

2.3.4 Mutação

As células são unidades estruturais que compõem todos os indivíduos vivos e

são responsáveis por múltiplas funcionalidades do organismo. Parte das células se reproduzem por meio de mitose, que corresponde a um processo de divisão celular, onde a célula se divide em duas novas células com cargas genéticas idênticas, porém um erro durante este processo pode ocorrer, gerando uma divergência na carga genética da célula, e essas falhas são chamadas de mutação (RIDLEY, 2006).

Em AGs, a mutação representa uma alteração aleatória na carga genética de um indivíduo, de modo que os dados de um determinado gene são modificados, como exemplificado na Figura 6.

Figura 6 - Exemplo de mutação

Mutação



Fonte: O autor (2019)

Esta função é aplicada em AGs porque somente a aplicação do *crossover* não garante a diversidade de genes nas populações das gerações posteriores, então a mutação de genes selecionados aleatoriamente ajuda a manter a diversidade nas populações. Desta forma, a *mutação* amplia o espaço de busca do algoritmo

3 TRABALHOS RELACIONADOS

Este capítulo apresenta um resumo de dois artigos que foram usados como fundamentação para este trabalho. O primeiro artigo se refere ao método Movelets, que corresponde ao método que este trabalho se propõe a otimizar. O segundo artigo descreve o método GENDIS, que utiliza AG para encontrar os melhores Shapelets de um conjunto de séries temporais. O método Movelets também foi inspirado nos Shapelets e por isso estudar o método GENDIS é importante para o objetivo do nosso trabalho.

3.1 MOVELETS: EXPLORING RELEVANT SUBTRAJECTORIES FOR ROBUST TRAJECTORY CLASSIFICATION

Ferrero et al. (2018) propõem em seu artigo um novo método para a classificação e descoberta de subtrajetórias relevantes baseado no conceito de Shapelets de séries temporais que não necessita da pré-definição de nenhum critério de partição de trajetória. Este se mostrou muito eficaz para a classificação de trajetórias, visto que ele superou todos os outros métodos da mesma finalidade em grande parte dos experimentos efetuados.

Para chegar nestes resultados, os autores averiguaram os trabalhos existentes a respeito de classificação de trajetórias, onde constataram que estes consistiam em extrair características globais, que consideram a trajetória como um todo, ou características locais, que consideram características de partes da trajetória, para construir o classificador, de modo que a análise de características globais leva em consideração toda a trajetória para a captura de informações de movimento, e a análise de características locais leva em consideração somente partes da trajetória para a análise. Também foi feita uma análise descritiva destes métodos, a fim de enfatizar as características levadas em consideração em cada abordagem.

Os autores utilizam as seguintes definições formais, que são base para este trabalho:

Trajatória: uma trajetória $T = \langle p_1, p_2, \dots, p_m \rangle$ é a uma sequência de pontos ordenados no tempo $p_i = (x, y, t)$, onde x, y correspondem à localização espacial do objeto no instante de tempo t . (FERRERO et al., 2018).

Subtrajetória: uma subtrajetória s é uma subsequência contígua de uma trajetória começando no ponto p_a e terminando no ponto p_b , onde $1 \leq a < m$ e $a < b < m$. O comprimento da subtrajetória é definido como $w = |s|$.

Distância entre dois pontos de trajetórias: Dados dois pontos de trajetórias, p_i e p_j , a distância entre esses pontos é dada pela função $Dist(p_i, p_j)$.

Nos experimentos realizados foi utilizada a distância euclidiana como função de distância.

Distância entre duas subtrajetórias de igual tamanho: Dadas duas subtrajetórias s_1 e s_2 de igual comprimento, $Dist(s_1, s_2)$ calcula a distância entre seus pontos sequenciais, retorna um valor não negativo e respeita a propriedade de simetria, $Dist(s_1, s_2) = Dist(s_2, s_1)$ (FERRERO et al., 2018).

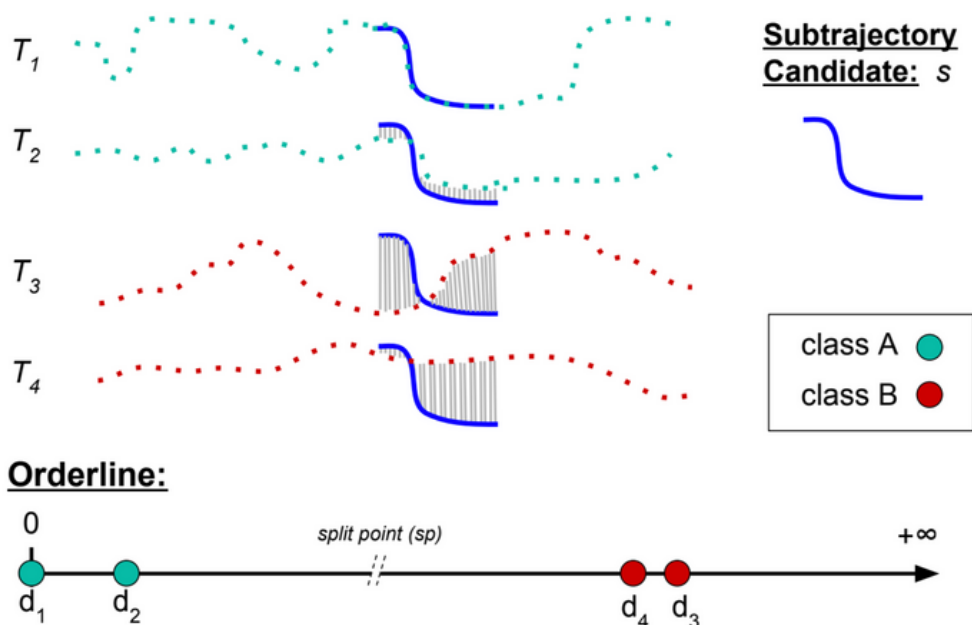
Distância entre trajetória e subtrajetória: Dada uma trajetória T e uma subtrajetória s de comprimento $w = |s|$, esta é a distância no melhor alinhamento de s em T , que é definido por $SubDist(s, T) = \min(Dist(s, r) \mid r \in S^w_t)$, onde S^w_t é o conjunto de todas subtrajetórias de comprimento w em T e $\min()$ retorna a menor distância entre s e todas as subtrajetórias em S^w_t (FERRERO et al., 2018).

Candidato a subtrajetória: corresponde a uma subtrajetória com (i) *identificação* da trajetória que deu origem a este candidato, (ii) o ponto de início (iii) tamanho da subtrajetória, (iv) um vetor com as distâncias entre a subtrajetória e todas as trajetórias do *dataset* e (v) a qualidade, que é o seu grau de relevância. Para calcular o ganho de informação da subtrajetória (Figura 7), o método ordena as distâncias da subtrajetória para as trajetórias em uma linha de ordenação de ordem crescente, e um ponto de corte é utilizado para dividir a linha em dois conjuntos

disjuntos, onde do lado esquerdo ficam somente distâncias que pertencem a trajetórias da mesma classe da subtrajetória analisada, e do lado direito ficam o restante das distâncias. Desta forma, o ponto de corte é definido entre o primeiro elemento de classe diferente da subtrajetória e último elemento anterior da mesma classe. O valor normalizado do ponto de corte corresponde a relevância do candidato.

Movelet: Dada uma trajetória T e uma subtrajetória candidata $s \in T$, esta subtrajetória é uma *movelet* se para cada subtrajetória $r \in T$ que intersecta s em pelo menos 1 ponto, $s.\text{qualidade} > r.\text{qualidade}$.

Figura 7 - Exemplo da linha de ordenação de um candidato a subtrajetória



Fonte: Ferrero et al. (2019)

Para melhor compreensão do funcionamento do método, os autores apresentam o algoritmo do método Movelets em três etapas, que correspondem a:

(1) descoberta: consiste em selecionar todas as subtrajetórias com maior relevância (movelets) do *dataset* analisado. Nesta etapa, são analisadas todas as possíveis subtrajetórias de tamanho maior que um (mais do que um único ponto) até

de tamanho da trajetória, para todas as trajetórias do *dataset*.

(2) poda: com base nas subtrajetórias mais relevantes encontradas, as que forem consideradas redundantes são removidas. A poda tem como intuito diminuir o conjunto de dados que irá para o classificador (movelet). Uma subtrajetória k é considerada redundante caso existam subtrajetórias de maior qualidade que cubram todas as trajetórias que estão à esquerda do *split-point* de k .

(3) transformação: esta etapa consiste em converter os movelets em uma matriz onde as linhas correspondem às trajetórias, as colunas correspondem aos movelets e os valores das células correspondem à distância da trajetória para o movelet. Essa transformação ocorre para que os dados gerados estejam compatíveis com o formato esperado para a análise de um classificador.

Após explicado o funcionamento do método, os autores analisam a sua complexidade. Em termos de espaço de memória, ele utiliza $O(n \times m^2)$, onde n é o número de trajetórias e m é o comprimento da trajetória mais longa, visto que ele armazena três matrizes de distância em memória de modo a diminuir o custo computacional. Além disso, armazena no máximo $O(m \times \log m)$ candidatos para cada trajetória. Portanto, a complexidade do espaço é $O(n \times m^2)$. Em termos de tempo, o processo geral requer $O(n^2 \times m^3)$.

Para avaliação da acurácia do Movelets, os autores propõem três experimentos, cada um com tipos de trajetórias diferentes, para demonstrar a adaptabilidade do método. Foram utilizados três algoritmos de classificação (SVM, C4.5 e *Naive Bayes*).

Foram utilizados 5 *datasets* para a avaliação do desempenho do método, que estão descritos na Figura 8, onde as colunas representam respectivamente o nome, número de trajetórias, comprimento médio e desvio padrão, número de classes e proporção das classes nos *datasets*.

Figura 8 - Descrição dos datasets

Dataset		# Traj	Length avg (sd)		Classes	
D_1	Hurricane _{2,3}	135	42.11	(17.21)	Scale 2 (46%) Scale 3 (64%)	
D_2	Hurricane _{1,4}	210	34.84	(17.33)	Scale 1 (71%) Scale 4 (29%)	
D_3	Hurricane _{0,4,5}	354	27.44	(17.03)	Scale 0 (76%) Scale 4,5 (24%)	
D_4	Animals	102	146.96	(62.51)	Elk (37%) Deer (30%) Cattle (33%)	
D_5	Vehicle	421	467.98	(250.53)	Bus (34%) Truck (66%)	

Fonte: Ferrero et al. (2018)

Os três primeiros conjuntos de dados (D_1 , D_2 e D_3) são relacionados a trajetórias de furacões do Atlântico. Sendo classificados usando o Escala de Simpson de 0 a 5, onde 0 é o mais fraco e 5 é o mais forte. O conjunto de dados D_1 contém as trajetórias das escalas 2 e 3, D_2 contém as trajetórias das escalas 1 e 4 e D_3 trajetórias das escalas 0, 4 e 5. O primeiro *dataset* é o menor dos *datasets* analisados em quantidade de pontos (tamanho médio * quantidade de trajetórias).

O conjunto de dados D_4 contém trajetórias relacionadas ao movimento de três espécies de animais: alce, veado e bovinos. E o conjunto de dados D_5 contém duas categorias de veículos que se deslocam em áreas metropolitanas. As trajetórias coletadas são de ônibus e caminhões escolares. Este último é o maior *dataset* do conjunto, tanto em quantidade de trajetórias quanto em tamanho médio das trajetórias.

Nesta experiência, o método Movelets foi comparado com outros métodos de classificação de trajetórias, que são considerados estado da arte para a classificação de trajetórias. Os três principais métodos são:

O método proposto por Zheng et al. (2010) utiliza somente atributos globais

para a classificação das trajetórias, que demonstraram bons resultados para a classificação de trajetórias, porém esta abordagem possui limitações, visto que um objeto móvel pode variar sua velocidade durante o percurso, e essa variação não é abordada por este método.

O método de Dodge, Weibel e Forootan (2009), que corresponde a uma proposta para extrair características locais transformando uma trajetória em uma série temporal com atributos para cada ponto. Estas séries temporais são discretizadas, a fim de diminuir a complexidade dos dados, porém a limitação desta abordagem é a perda de informações neste processo de discretização.

A proposta de Xiao et al. (2017), que corresponde a um método que utiliza características globais gerados por um método estatístico e características locais extraídos por segmentação das trajetórias analisadas, que são combinadas para obter melhores resultados de classificação.

A Figura 9 apresenta na esquerda, os resultados da classificação utilizando *cross-validation*, e a coluna da direita apresenta resultados utilizando *holdout* (60% dos dados para treino e 40% para teste). Os resultados obtidos demonstram que o Movelets se mostrou superior na classificação de todos os *datasets* em quase todas as análises executadas.

Figura 9 - Comparativo entre Movelets e outros métodos de classificação de trajetórias

Dataset	SVM					MOVELETS
	TB-RB	TCPR	Dodge	Zheng	Xiao	
Hurricane _{2,3}	0.46	0.55	0.50	0.50	0.52	0.75
Hurricane _{1,4}	0.72	0.78	0.72	0.77	0.78	0.86
Hurricane _{0,45}	0.71	0.87	0.85	0.85	0.86	0.90
Animals	0.79	0.89	0.74	0.79	0.87	0.97
Vehicle	0.94	0.99	0.94	0.84	0.98	0.98

Dataset	SVM				MOVELETS
	Dodge	Zheng	Xiao		
Hurricane _{2,3}	0.56	0.59	0.53		0.60
Hurricane _{1,4}	0.79	0.75	0.77		0.85
Hurricane _{0,45}	0.87	0.87	0.86		0.88
Animals	0.67	0.68	0.81		0.90
Vehicle	0.89	0.78	0.98		0.99

Dataset	C4.5					MOVELETS
	TB-RB	TCPR	Dodge	Zheng	Xiao	
Hurricane _{2,3}	0.53	0.56	0.47	0.49	0.60	0.62
Hurricane _{1,4}	0.69	0.73	0.72	0.76	0.74	0.78
Hurricane _{0,45}	0.71	0.83	0.83	0.83	0.81	0.85
Animals	0.80	0.89	0.74	0.83	0.81	0.96
Vehicle	0.94	0.98	0.85	0.94	0.92	0.98

Dataset	C4.5				MOVELETS
	Dodge	Zheng	Xiao		
Hurricane _{2,3}	0.39	0.62	0.51		0.62
Hurricane _{1,4}	0.81	0.71	0.76		0.85
Hurricane _{0,45}	0.84	0.85	0.82		0.83
Animals	0.67	0.76	0.74		0.93
Vehicle	0.73	0.90	0.94		0.96

Dataset	Bayes					MOVELETS
	TB-RB	TCPR	Dodge	Zheng	Xiao	
Hurricane _{2,3}	0.35	0.45	0.53	0.55	0.48	0.76
Hurricane _{1,4}	0.74	0.79	0.70	0.70	0.66	0.86
Hurricane _{0,45}	0.71	0.85	0.80	0.82	0.78	0.87
Animals	0.70	0.77	0.51	0.70	0.77	0.91
Vehicle	0.92	0.97	0.71	0.60	0.74	0.99

Dataset	Bayes				MOVELETS
	Dodge	Zheng	Xiao		
Hurricane _{2,3}	0.54	0.56	0.54		0.60
Hurricane _{1,4}	0.72	0.68	0.68		0.80
Hurricane _{0,45}	0.76	0.86	0.81		0.84
Animals	0.63	0.64	0.76		0.93
Vehicle	0.76	0.47	0.81		0.97

Fonte: Ferrero et al. (2018)

3.2 GENDIS: GENETIC DISCOVERY OF SHAPELETS

Shapelets são séries temporais relevantes para a classificação de séries temporais. O conceito e um método de implementação de shapelets foram propostos em Ye e Keogh (2009).

Vandewiele, Ongena e De Turck (2019) propõem em seu artigo a criação de um modelo para a descoberta de shapelets baseado em algoritmo genético, trazendo como vantagens a facilidade de encontrar candidatos adequados, redução de complexidade computacional de execução e definição de quantidade e comprimento de shapelets em tempo de execução.

Como motivação deste trabalho, os autores citaram a proposta inicial para descoberta do método *shapelets*, onde a complexidade computacional é muito alta ($O(n^2 \times m^3)$, onde n corresponde ao número de séries temporais e m o tamanho da menor série temporal no conjunto de dados). Alguns trabalhos citados no artigo buscaram a redução da complexidade do método, porém necessitam da especificação de parâmetros para a execução do método ou pioram o resultado da classificação no conjunto de dados testados.

O GENDIS se propõe como um método de descoberta de um conjunto de *shapelets* com base em um *dataset* de trajetórias baseado em Algoritmos Genéticos. O objetivo do algoritmo proposto é alcançar um desempenho preditivo de última geração mantendo uma baixa complexidade computacional, permitindo assim que qualquer função proposta consiga chegar em um bom resultado de busca. Em segundo lugar, a quantidade total de *shapelets* e o comprimento de cada um desses *shapelets* não precisa ser definido antes da descoberta, aliviando a necessidade de ajuste, o que poderia ser computacionalmente caro.

O método se baseia na implementação de um AG padrão, contando com operadores de reprodução, mutação e seleção. Na implementação, o gene corresponde a um ponto de uma trajetória de referência, fazendo com que o indivíduo corresponda a uma subtrajetória. A quantidade de genes por indivíduo não é fixa, fazendo com que os indivíduos possam apresentar diferentes tamanhos.

inicialização: duas estratégias podem ser utilizadas neste método para seleção dos candidatos iniciais: aplicação do método *K-means* de agrupamento em um subconjunto de séries temporais selecionadas aleatoriamente, onde os centróides resultantes formam um conjunto de candidatos, ou a geração de um conjunto de candidatos de comprimentos aleatórios. As duas abordagens são aceitas visto que a primeira resulta em indivíduos iniciais fortes, e a segunda aumenta a diversidade populacional e tem o tempo necessário para inicializar a população inferior quando comparado com a primeira abordagem.

elitismo: para garantir que o candidato mais apto nunca seja descartado da

população, ou seja, alterado por mutações que prejudicam a sua aptidão o método utiliza o elitismo, onde a função de *fitness* analisa o vetor de distâncias com o melhor desempenho preditivo quando fornecido a um classificador.

reprodução: esta é feita por meio de duas abordagens, a primeira divide os conjuntos de *shapelets* em um ou dois pontos, de modo a criar novos conjuntos de *shapelets* que são compostos por *shapelets* de ambos os pais, e a segunda abordagem utiliza uma média dos pontos pertencentes ao conjunto de *shapelets*, buscando os mais similares para formar o novo *shapelet*.

mutação: são utilizadas três abordagens diferentes para a mutação, onde a primeira remove aleatoriamente uma quantidade variável de pontos do começo ou fim do *shapelet*, a segunda remove um *shapelet* aleatório e a terceira cria um novo *shapelet* e o adiciona ao indivíduo.

torneio: após cada geração, um número fixo de conjuntos de candidatos é escolhido com base em sua adequação para a próxima geração, onde a técnica utilizada foi a seleção por meio de torneios com tamanhos pequenos.

De modo a determinar o número ideal de gerações que o AG deve executar, o método proposto termina sua execução quando não encontra mais nenhum candidato melhor do que na execução anterior, encerrando sua execução quando não obtém mais convergência entre as gerações.

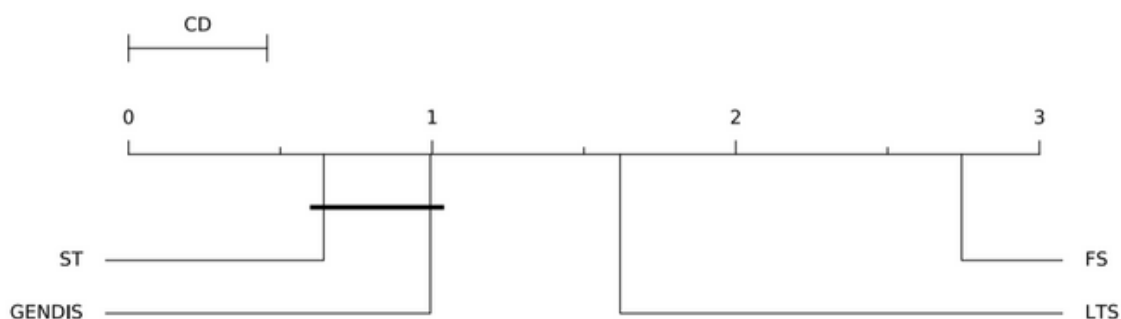
Para a avaliação de performance os autores comparam a acurácia com 3 técnicas de descoberta de *shapelets* três técnicas de descoberta de *shapelet*: *Shapelet Transform* (ST) de (LINES et al., 2012); *Learning Timeseries Shapelets* (LTS) de (GRABOCKA et al., 2014); e *Fast Shapelets* (FS) (RAKTHANMANON; KEOGH, 2016), que são utilizadas para fins comparativos com o GENDIS.

Foram utilizados 85 *datasets* para a comparação dos resultados, utilizando a mesma configuração de parâmetros: o tamanho da população de 100; no máximo 100 iterações; parada precoce após 10 iterações sem convergência; probabilidade de cruzamento 4%; e mutação de 1%. O único parâmetro que foi ajustado para cada

conjunto de dados separadamente era um comprimento máximo para cada shapelet, onde foi utilizada uma função que leva em consideração o tamanho da série temporal. A matriz de distância obtida através dos shapelets extraídos é então inserida em um classificador e o resultado é coletado.

A fim de demonstrar os resultados, a Figura 10 representa a precisão média do GENDIS em comparação com a média das medições dos outros três algoritmos.

Figura 10 - Diagrama de diferença crítica entre os quatro algoritmos avaliados



Fonte: Vandewiele, Ongena e De Turck (2019)

Estes resultados demonstram que ST possui melhor precisão, porém o GENDIS apresentou um resultado muito aproximado, demonstrando que a acurácia resultante dos dois métodos são estatisticamente equivalentes, levando em consideração que a distancia critica (CD) indica que a probabilidade da diferença de performance entre os classificadores ocorrer por acaso é menor do que 1%.

Quanto ao tempo de execução, o artigo relata que a técnica apresentada apresenta a complexidade computacional igual a $G P K N M^2$, que é relatada como uma complexidade inferior ao ST, que apresenta a complexidade de $O(N^2 M^3)$, porém não são apresentas evidências desta melhora no tempo de processamento, pois não é apresentada nenhuma comparação de tempo de processamento.

4 MÉTODO PROPOSTO

Ao fazer análises e compreender a dinâmica de conceito do método Movelets mostrado na Seção 3.1, é possível observar que sua execução é muito custosa, tornando-o inviável em *datasets* muito grandes, ou com grande número de pontos por trajetória.

Este trabalho tem como objetivo o desenvolvimento de um método que utiliza AG para encontrar subtrajetórias relevantes. O método proposto usa os conceitos básicos do Movelets, e se propõe a ser uma alternativa à este trabalho, com o intuito de otimizar o custo computacional de execução mantendo a acurácia na classificação.

4.1 ESTRUTURA DO ALGORITMO

O método tem como objetivo processar um *dataset* de trajetórias de modo a encontrar as subtrajetórias mais relevantes para classificação utilizando a implementação de AG para encontrar melhores subtrajetórias relevantes no decorrer das gerações.

Os indivíduos são formados por conjunto de genes, que correspondem a uma subtrajetória com n pontos pertencentes a uma trajetória t do conjunto de dados. Um indivíduo pode possuir subtrajetórias de diferentes classes, que depois vão ser avaliadas a fim de encontrar o *fitness* deste indivíduo.

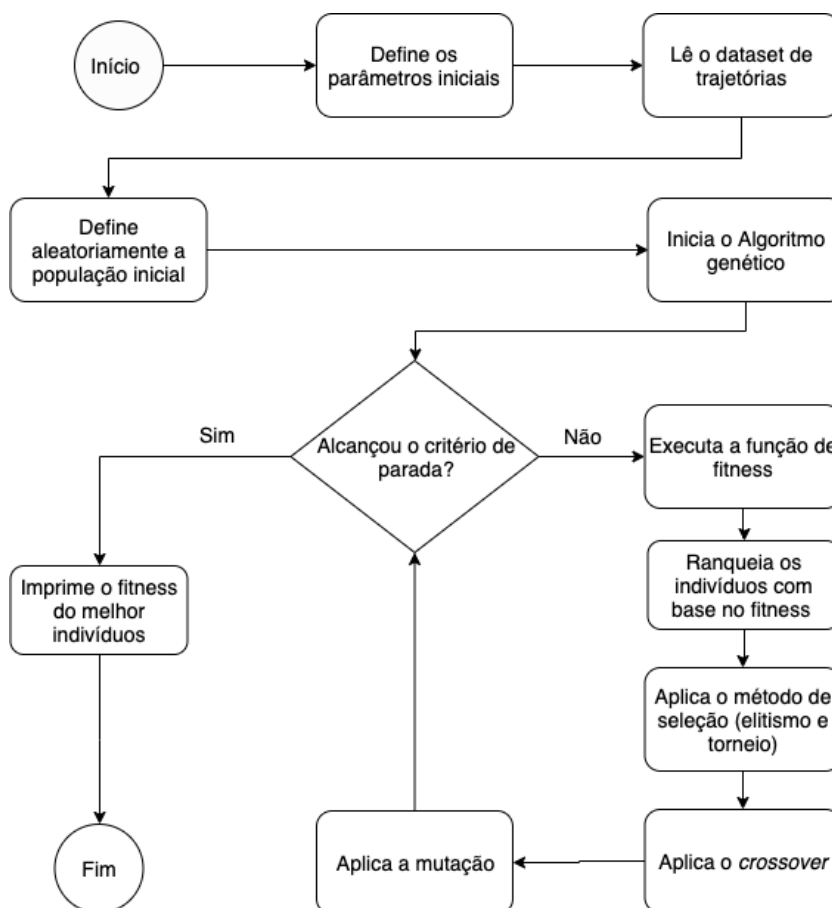
Os parâmetros de tamanho de população, de indivíduos e a quantidade de gerações utilizadas são variáveis configuradas na inicialização do método. A inicialização da população é feita de forma aleatória de modo a aumentar a diversidade da população inicial.

A Figura 11 representa a execução do AG, onde o método primeiro calcula o *fitness* de cada indivíduo utilizando um classificador e ranqueia os indivíduos com base na acurácia de classificação obtida. A seleção dos indivíduos usa uma função

de Elitismo para garantir os melhores na próxima geração e Torneio ou Roleta para a escolha dos indivíduos que participarão da reprodução. Os indivíduos selecionados são inseridos em uma função de *crossover*, que retorna um indivíduo com genes provindos dos pais selecionados, e ainda é aplicada uma função de mutação, onde ocorre a alteração dos genes de um indivíduo selecionado aleatoriamente.

No final da execução do método, é retornado o valor de acurácia de classificação do melhor indivíduo, que representa o valor final de acurácia para a execução em questão.

Figura 11 - Diagrama de execução do algoritmo proposto



Fonte: O autor (2019)

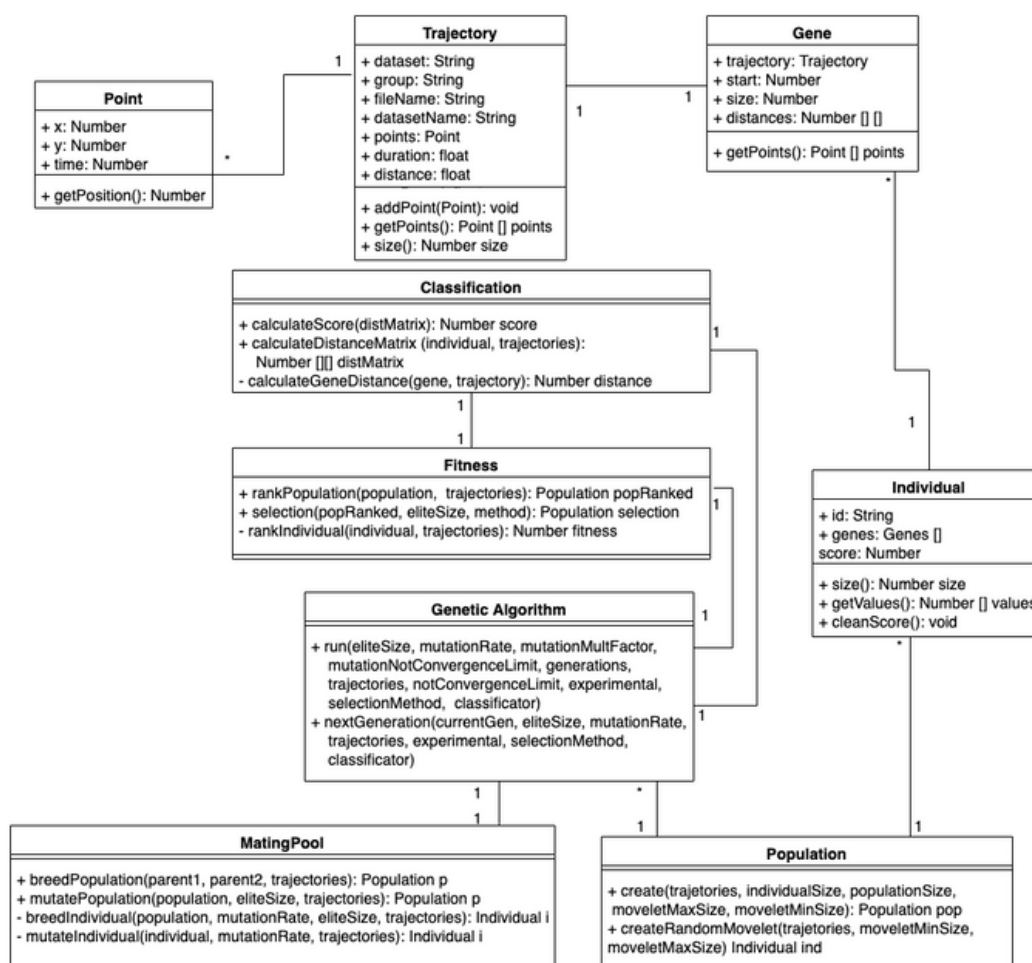
4.2 IMPLEMENTAÇÃO DO ALGORITMO

Para o desenvolvimento deste trabalho foi utilizada a linguagem Python, utilizando a biblioteca *sklearn*, que é uma biblioteca de aprendizado de máquina de código aberto. A implementação se baseou nos conceitos de análise de trajetória utilizadas pelo método Movelets, também foram utilizados os conceitos básicos de AG citados na fundamentação teórica.

4.2.1 Diagrama de classes

O algoritmo é estruturado utilizando a abordagem orientada a objeto, e a Figura 12 representa o seu diagrama de classes, descrevendo suas respectivas classes e relações.

Figura 12 - Diagrama de classes do algoritmo



Fonte: O autor (2019)

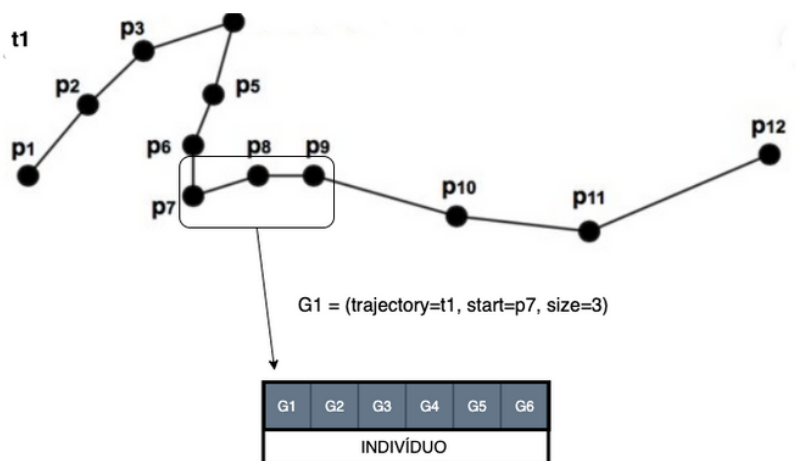
Point: representa um ponto de uma trajetória. Contém as coordenadas (x e y) do ponto e o momento que foi coletado (*time*).

Trajectory: representa uma trajetória. Contém um conjunto de pontos (*points*), o informativo de qual a classe dessa trajetória (*group*), e atributos globais como duração (*duration*), distância percorrida (*distance*) e velocidade média (*avgSpeed*).

Gene: equivalente à um candidato a subtrajetória representativa descrito no trabalho de Ferrero et al. (2018). Contém uma referência a qual trajetória pertence (*trajectory*), a quantidade de pontos que a subtrajetória possui (*size*) e o seu ponto de início na trajetória (*start*), como exemplificado na Figura 13. Para fins de

otimização da execução, é salva a matriz de distâncias entre o Gene e todas as trajetórias do *dataset* (*distances*), que são os valores utilizados de base para o cálculo de *fitness* de um indivíduo.

Figura 13 - Exemplo de gene



Fonte: O autor (2019)

Individual: é definido como um conjunto de genes, e também possui o atributo *score*, que representa o seu valor de *fitness*.

Population: equivale a um conjunto de indivíduos, possui o método de geração da população inicial em seu escopo.

GeneticAlgorithm: é a classe principal da implementação, nela que ocorre a evolução das gerações, e todas as outras classes são instanciadas dentro desta função de evolução. A função *nextGeneration* recebe a população, executa as funções da classe *Fitness* e *MatingPool*, e retorna uma nova população.

Fitness: classe utilizada para executar o método de ranqueamento e seleção dos indivíduos. Ela instancia a classe *Classification* para calcular o *fitness* dos indivíduos.

Classification: é a classe responsável por calcular a matriz de distâncias entre os genes de um indivíduo e as trajetórias do *dataset* e por executar a classificação dos indivíduos com base na matriz.

MatingPool: Esta classe implementa os operadores genéticos utilizados no algoritmo, dentro dela estão as funções de reprodução e mutação utilizadas na população.

4.2.2 Inicialização do AG

No início da execução, o algoritmo faz a leitura das trajetórias e cria a população inicial com base nos seguintes parâmetros setados:

- Tamanho do indivíduo
- Tamanho da população
- dataset que será analisado

Os indivíduos da população inicial são gerados de forma aleatória, de forma que os genes são criados por uma função randômica que seleciona uma trajetória de referência, um ponto de início e o tamanho da subtrajetória. Para os experimentos realizados neste trabalho, utilizamos subtrajetórias com tamanhos entre 2 e 3 pontos.

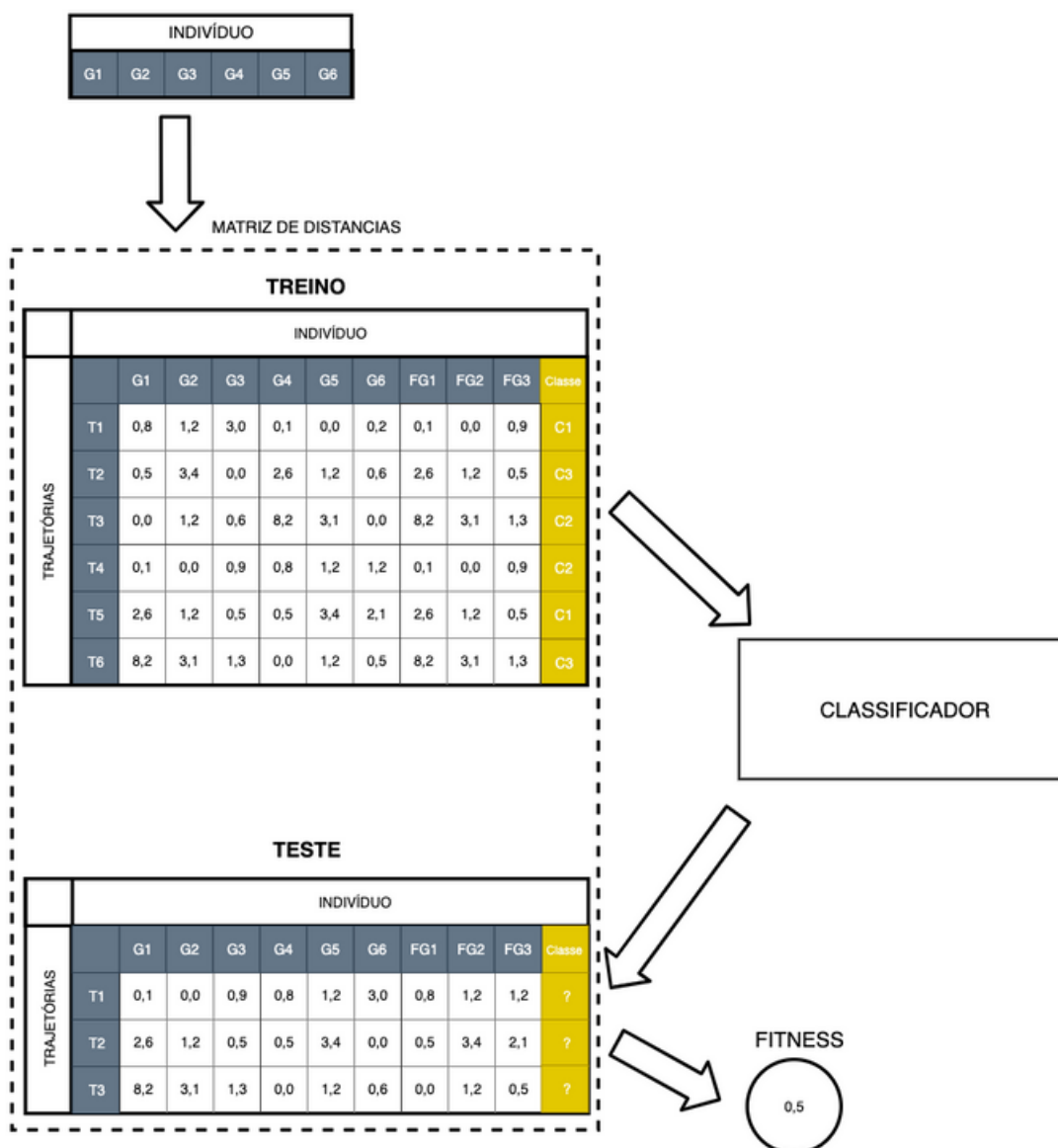
4.2.3 Fitness

A função de fitness consiste no resultado da acurácia obtida a partir da classificação do indivíduo com base nas trajetórias do conjunto de dados.

Uma matriz de distâncias é calculada, onde cada linha corresponde a uma trajetória e cada coluna corresponde a um gene de um determinado indivíduo. As células da matriz contém a distância entre o gene e a trajetória, cujo o valor corresponde a menor distância entre o indivíduo e todas as subtrajetórias de mesmo tamanho da trajetória comparada. Adicionalmente são também incluídos para cada trajetória três atributos globais à esta matriz: a distância total, o tempo de duração e velocidade média da trajetória (distância / tempo). E finalmente, como última coluna, temos a classe de cada trajetória.

O classificador utiliza a matriz fornecida para gerar um modelo, que é aplicado em uma matriz-teste, e o resultado da acurácia deste modelo resulta no valor de *fitness* do indivíduo, como representado na Figura 14.

Figura 14 - Exemplo de geração do fitness



Fonte: O autor (2019)

4.2.4 Elitismo

Para evitar a perda das melhores soluções já encontradas foi utilizada a técnica de elitismo para garantir que o melhor indivíduo continue na população. A cada nova geração é mantido um único indivíduo com o melhor valor de *fitness* da geração anterior para a nova geração.

4.2.5 Seleção

De modo a encontrar os indivíduos com os melhores genes para a reprodução, foram implementados os métodos de seleção de roleta e torneio, sendo possível setar qual método será utilizado pelo algoritmo por meio de um parâmetro de execução.

Todos os resultados deste trabalho foram coletados utilizando o método de torneio, pois ele se mostrou melhor no quesito de diversidade dos indivíduos na população.

4.2.6 Reprodução

Para a realização da reprodução, é utilizada uma função de cruzamento (*crossover*) de um ponto, onde é selecionado aleatoriamente um ponto de corte entre os genes, e o filho gerado recebe o subconjunto de genes do primeiro pai à esquerda do ponto de corte, e então os demais genes do novo indivíduo são completados com o subconjunto de genes a direita do ponto de corte do segundo pai.

4.2.7 Mutação

A mutação acontece com base no parâmetro de taxa de mutação, que

equivale a um percentual que representa a probabilidade do indivíduo sofrer alteração na sua carga genética. Uma função probabilística é executada para selecionar se o indivíduo vai sofrer mutação ou não. Caso ele tenha sido selecionado, uma segunda função é responsável por selecionar o gene que sofrerá a mutação, e uma terceira função é responsável por selecionar o tipo de mutação que irá acontecer. Neste processo, o gene pode ser totalmente recriado, ou somente sofrer alterações na sua trajetória de referência, ponto de início ou tamanho.

Na implementação realizada, os indivíduos selecionados pelo elitismo não sofrem mutações, evitando desta forma a perda de acurácia no decorrer das gerações.

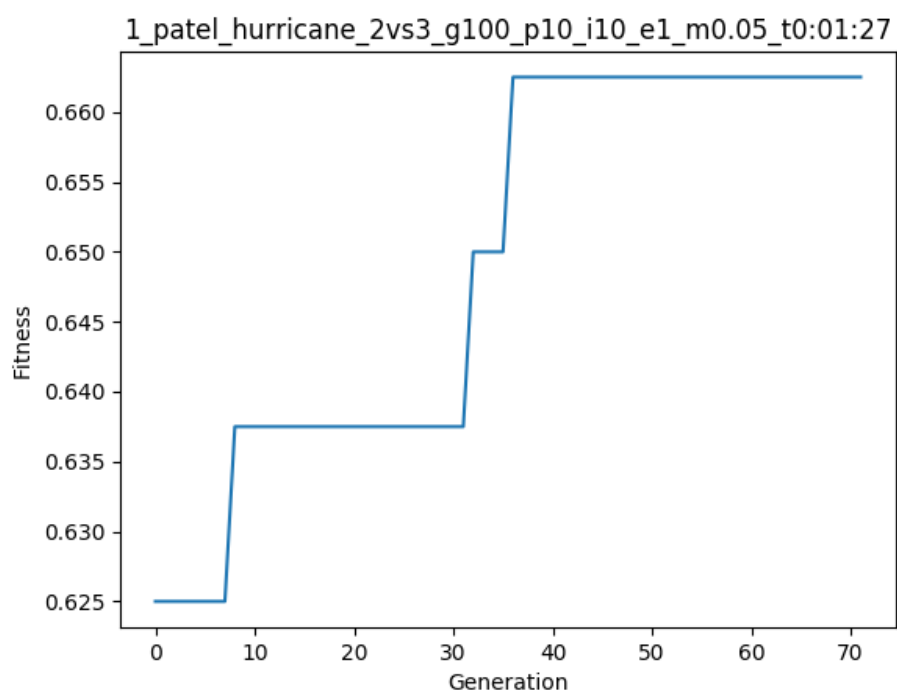
4.2.8 Critério de Parada

Como critério de parada, o método proposto utiliza dois parâmetros, quantidade de gerações, onde ele encerra a execução quando atinge um número de gerações pré-estabelecido, ou quantidade de gerações sem convergência, onde o método encerra a execução quando atinge uma quantidade de gerações em que não obteve um indivíduo com valor de *fitness* maior do que nas gerações anteriores.

4.2.9 Visualização da evolução do melhor indivíduo

Para visualizar a evolução do melhor indivíduo do AG no decorrer das gerações, é utilizado um gráfico que demonstra a variação do *fitness* do melhor indivíduo no decorrer das gerações, que é demonstrado na Figura 15. Este gráfico tem como intuito servir de parâmetro para avaliar a conversão do algoritmo genético no decorrer da sua execução.

Figura 15 - Exemplo de gráfico de fitness do melhor indivíduo x gerações



Fonte: O autor (2019)

5 EXPERIMENTOS

A fim de testar o algoritmo desenvolvido, inicialmente foram realizados testes para encontrar a melhor combinação de parâmetros de execução. Após isso, o método foi executado com os mesmos *datasets* descritos na seção 3.1 e utilizados no trabalho de Ferrero *et al de* (2018) e os resultados foram comparados com os resultados dos métodos citados no artigo que propõe o método Movelets. Um comparativo de tempo computacional entre o Movelets e o AG proposto também foi executado. Para a execução dos experimentos, foi utilizado um Macbook Pro com processador Intel Core I7 1.7GHz, 16GB de memória e sistema operacional macOS.

5.1 VARIAÇÃO DA ACURÁCIA E TEMPO DE EXECUÇÃO EM FUNÇÃO DOS PARÂMETROS

Para encontrar a melhor combinação de parâmetros de execução do AG utilizando *datasets* de trajetórias, foram executados testes levando em consideração o tempo e a acurácia de execução do método. Estes valores foram representados em gráficos, e no final, uma análise ponderada entre estes dois valores coletados foi feita, a fim de encontrar o parâmetro com melhor acurácia no menor tempo de execução, visto que a proposta deste trabalho é otimizar o tempo de execução do Movelets mantendo bons resultados de acurácia.

Como parâmetros iniciais para a execução dos testes, foram selecionados atributos que apresentaram bons resultados nas análises feitas durante a implementação do método, cujo os valores são:

Dataset: Hurricanes2,3

Tamanho da população: 10 indivíduos

Tamanho dos indivíduos: 8 genes

Gerações: 40

Taxa de mutação: 10%

Método de classificação: Naive Bayes

Método de seleção: torneio

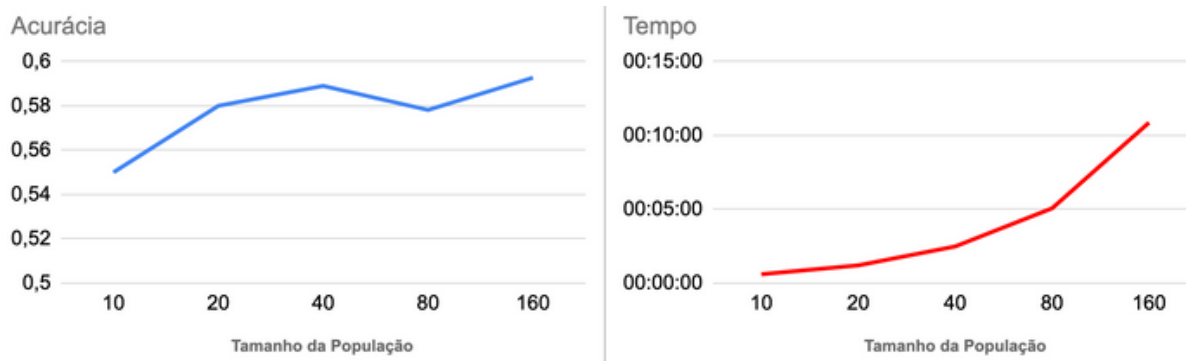
Levando em consideração o fato que os indivíduos são gerados inicialmente de forma aleatória e a característica não determinística dos algoritmos genéticos que pode fazer com que execuções realizadas com os mesmos parâmetros apresentem resultados diferentes, foi definido que a acurácia da classificação para a realização das análises é o valor médio do fitness dos melhores indivíduos resultantes de 5 execuções contíguas, isto é, a média da acurácia de 5 execuções do algoritmo.

Os gráficos a seguir representam os resultados obtidos variando os parâmetros a fim de encontrar o melhor custo-benefício, onde a linha azul representa a evolução da acurácia do método e a vermelha representa a evolução do tempo de execução, representado no formato hh:mm:ss. Todos os resultados foram coletados variando somente um parâmetro por vez.

5.1.1 Tamanho da população

A primeira avaliação foi executada variando o tamanho da população, começando com 10 indivíduos por geração e variando até 160 indivíduos. O Gráfico 1 apresenta os resultados obtidos.

Gráfico 1 - Acurácia e Tempo de execução variando o tamanho da população



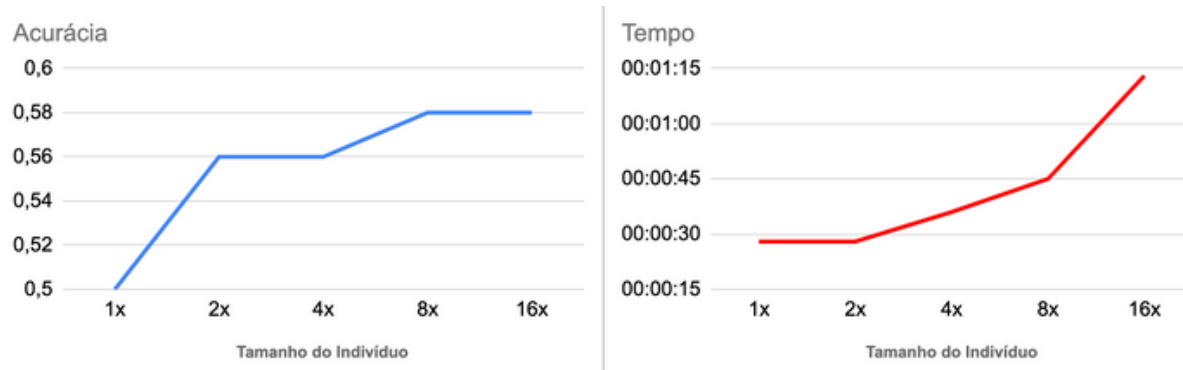
Fonte: O autor (2019)

Analisando os resultados, é possível observar que a variação do tamanho da população gerou uma variação de acurácia de 4 pontos percentuais entre a pior e a melhor execução (0,55 com população de tamanho 10 e 0,59 com população de tamanho 160), e a maior evolução da acurácia foi notada entre as 3 primeiras execuções do método. Já o tempo de execução tendeu a ter um aumento médio de 106,36% entre as execuções, o que representa que a variação deste parâmetro tem um grande impacto no tempo final de execução do método.

5.1.2 Tamanho dos indivíduos

Na segunda avaliação, alterou-se a quantidade de genes do indivíduo. Para a execução deste testes foi utilizado como valor de referência a quantidade de classes do *dataset* analisado, e o valor de tamanho do indivíduo deve ser múltiplo à referência (para um *dataset* com 4 classes, 2x representa que o tamanho do indivíduo é igual a 8, por exemplo). Foi escolhida esta forma de representação do indivíduo pois o número de classes tende a variar de um *dataset* para outro, e é esperado que os indivíduos tenham genes de referência para todas as classes existentes.

Gráfico 2 - Acurácia e Tempo de execução variando o tamanho do indivíduo



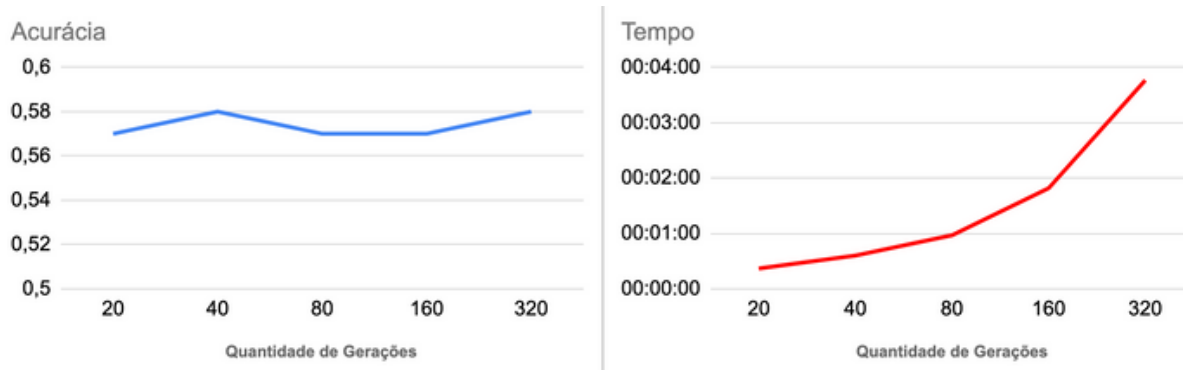
Fonte: O autor (2019)

Nesta avaliação demonstrada no Gráfico 2, foi observado um ganho de 8 pontos percentuais entre a pior e a melhor execução (0,50 com tamanho do indivíduo igual a 1x e 0,58 com tamanho do indivíduo igual a 16x). Também é possível observar que o ganho de acurácia aconteceu nos três primeiros testes, e depois estabilizou, já o tempo teve um crescimento médio de 28,95%, variando de 28 segundos na primeira execução até 1 minuto e 13 segundos na última execução.

5.1.3 Gerações

Na terceira avaliação, alterou-se a quantidade de gerações executadas pelo algoritmo genético, variando de 20 até 320 gerações. O Gráfico 3 mostra os resultados obtidos.

Gráfico 3 - Acurácia e Tempo de execução variando a quantidade de gerações



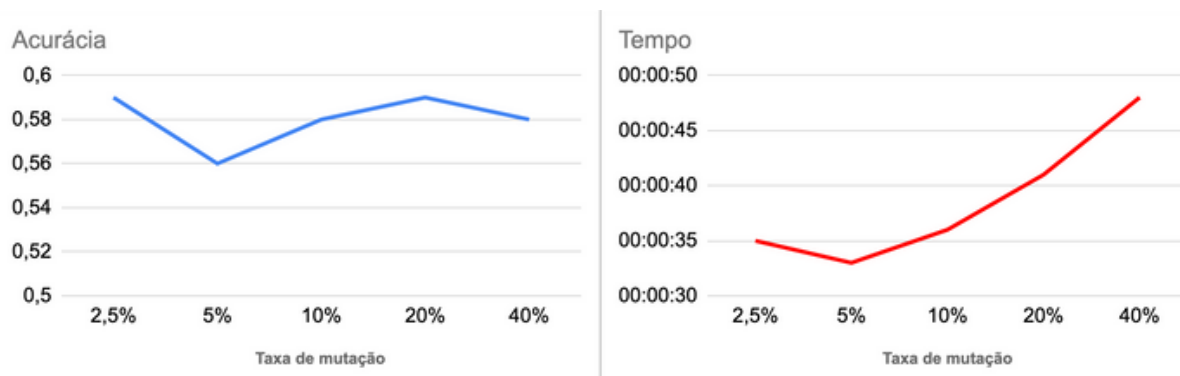
Fonte: O autor (2019)

Como é possível observar, a acurácia se manteve estável nos testes, variando 1 ponto percentual em média, já o tempo de execução tendeu a um aumento médio de 80% entre as execuções.

5.1.4 Taxa de mutação

Nesta avaliação, alterou-se a taxa de mutação, variando seu valor de 2,5% até 40%. O Gráfico 4 demonstra os resultados obtidos durante as execuções.

Gráfico 4 - Acurácia e Tempo de execução variando a taxa de mutação



Fonte: O autor (2019)

No Gráfico 4, é possível observar que a acurácia caiu e depois tendeu a subir novamente, oscilando 3 pontos percentuais entre as execuções, já o tempo tendeu a

um aumento de 8,58% entre as execuções, menor média obtida entre os parâmetros observados.

5.1.5 Definição dos parâmetros utilizados nas comparações

Com base nos resultados coletados nas avaliações, foram definidos os parâmetros para a execução dos testes comparativos, e os critérios utilizados para as escolhas foi uma ponderação entre o ganho de acurácia e de tempo de execução. Os parâmetros definidos foram:

(i) tamanho da população de 20 indivíduos: Neste caso, tivemos o maior ganho de acurácia entre população de tamanho 10 e tamanho 20, e como a variação deste parâmetro é muito custosa em tempo computacional para a execução (106,36%), a opção selecionada foi a que fornecia maior ganho com menor custo

(ii) tamanho de indivíduos igual a 8 vezes a quantidade de classes do dataset: Para esta hipótese, selecionamos o parâmetro que forneceu maior ganho nos testes, pois a variação deste parâmetro tem custo relativamente baixo (28,94%).

(iii) quantidade de gerações igual a 40 gerações sem convergência: Neste caso, inserimos um parâmetro de análise de gerações sem convergência, pois como a variação da acurácia nos resultados não foi significativa, foi concluído que o parâmetro para fim da execução seria de gerações que não resultam no aumento da acurácia entre uma geração e outra e pode se encerrar mais rapidamente caso não exista mais convergência.

(iv) taxa de mutação inicial de 2,5%: Para este parâmetro, decidimos utilizar a menor taxa de mutação testada como parâmetro inicial, porém a variação deste parâmetro é importante para contribuir com a diversidade das populações no decorrer da execução do método, e por isso, foi definido um fator de multiplicação que dobra a taxa de mutação a cada 10 gerações sem convergência, assim é possível aproveitar os indivíduos gerados aleatoriamente no início da execução, e depois de certo tempo sem convergência, o aumento na taxa de mutação busca

contribui para o aumento de diversidade.

5.2 COMPARATIVO COM OUTROS MÉTODOS

Com base nos parâmetros definidos na sessão anterior, foram executados testes comparativos entre os resultados do AG e os métodos de Ferrero et al. (2018), Zheng et al. (2010), Dodge, Weibel e Forootan (2009) e Xiao et al. (2017), que foram utilizados como base comparativa no artigo do Movelets. Para execução dos modelos e geração do *fitness* dos indivíduos, foram utilizados os algoritmos de classificação Naive Bayes, C4.5 e SVM com avaliação por abordagens de *holdout* e *cross-validation*. Os 5 *datasets* utilizados para a execução das análises são os mesmos utilizados nas comparações de Ferrero et al. (2018), e são relacionados a trajetórias de furacões (*Hurricane2,3*, *Hurricane1,4* e *Hurricane0,45*), animais (*Animals*) e veículos (*Vehicles*). Mais informações a respeito dos *datasets* podem ser obtidas consultando a Figura 8.

Para a realização destes testes com o intuito de análise comparativa com outros métodos de classificação de trajetórias, foi utilizado a mesma função de média da acurácia de 5 execuções do AG (AG médio), porém também apresentamos o valor de acurácia da melhor execução (AG melhor).

5.2.1 Holdout

Na realização dos experimentos com *holdout*, os *datasets* foram divididos em conjunto de treinamento (60%) e conjunto de teste (40%). Nesta modalidade, o AG utiliza o conjunto de dados de treino para a execução do AG, e o melhor indivíduo da última geração executada é utilizado como treino do classificador, que usa o conjunto de teste para gerar o resultado final da avaliação.

As Tabelas 1, 2 e 3 mostram os resultados da avaliação de *holdout* para a classificação utilizando Naive Bayes, C4.5 e SVM. Os melhores resultados da

classificação para cada conjunto de dados são destacados em negrito e os em segundo melhores estão sublinhados, para facilitar as comparações.

Tabela 1 - Avaliação de holdout utilizando Bayes

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.54	0.56	0.54	0.60	0.59	0.56
Hurricane1,4	0.72	0.68	0.68	0.80	0.77	0.74
Hurricane0,45	0.76	0.86	0.81	0.84	0.81	0.78
Animals	0.63	0.64	0.76	0.93	0.81	0.81
Vehicle	0.76	0.47	0.81	0.97	0.85	0.79

Fonte: O autor (2019)

Tabela 2 - Avaliação de holdout utilizando C4.5

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.39	0.62	0.51	0.62	0.67	0.60
Hurricane1,4	0.81	0.71	0.76	0.85	0.86	0.81
Hurricane0,45	0.84	0.85	0.82	0.83	0.83	0.80
Animals	0.67	0.76	0.74	0.93	0.93	0.87
Vehicle	0.73	0.90	0.94	0.96	0.98	0.97

Fonte: O autor (2019)

Tabela 3 - Avaliação de holdout utilizando SVM

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.56	0.59	0.53	0.60	0.65	0.61
Hurricane1,4	0.79	0.75	0.77	0.85	0.83	0.81
Hurricane0,45	0.87	0.87	0.86	0.88	0.87	0.84
Animals	0.67	0.68	0.81	0.90	0.89	0.89
Vehicle	0.89	0.78	0.98	0.99	0.92	0.89

Fonte: O autor (2019)

Analisando os resultados apresentados nestas tabelas, observa-se que o método proposto apresentou melhor acurácia para 33% dos casos e segundo melhor resultado para 47% das execuções. Em comparação com o Movelets, os resultados mostram que, para os classificadores Bayes e SVM, o método proposto apresentou resultados em média 3.3 pontos percentuais abaixo do Movelets. Já com o classificador C4.5, os resultados ficaram em média 1.6 pontos percentuais acima dos resultados do Movelets.

No que diz respeito ao tempo de execução entre o Movelets e o algoritmo proposto, a Tabela 4 demonstra o tempo de execução para cada *dataset*, e é possível observar que o tempo do Movelets foi em média 12.1 vezes mais veloz nos 4 primeiros *datasets*, e 5.7 vezes mais lento para o último *dataset*.

Tabela 4 - Tempo de execução da avaliação com holdout

Dataset	Movelets	AG/Bayes	AG/C4.5	AG/SVM
Hurricane2,3	00:00:08	00:02:01	00:02:11	00:00:50
Hurricane1,4	00:00:10	00:04:32	00:02:23	00:01:07
Hurricane0,45	00:00:14	00:03:34	00:03:36	00:01:46
Animals	00:00:38	00:04:59	00:02:50	00:05:46
Vehicle	04:08:47	01:17:54	00:27:43	00:24:24

Fonte: O autor (2019)

5.2.2 Cross-validation

Para a realização dos experimentos utilizando *cross-validation*, foi utilizado a divisão do *dataset* em 5 partes (*folds*). Como tanto para o Movelets quanto para o AG proposto neste trabalho, o método tem que ser reexecutado a cada vez que os dados de treino são alterados, o *cross-validation* foi executado “manualmente”, e os dados apresentados correspondem as médias dessas 5 execuções.

As Tabelas 5, 6 e 7 mostram os resultados da avaliação, e a Tabela 8 mostra o comparativo do tempo de execução do Movelets com o AG.

Tabela 5 - Avaliação de cross-validation utilizando Bayes

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.53	0.55	0.48	0.76	0.67	0.65
Hurricanes1,4	0.70	0.70	0.66	0.86	0.82	0.81
Hurricane0,45	0.80	0.82	0.78	0.87	0.84	0.83
Animals	0.51	0.70	0.77	0.91	0.89	0.88
Vehicle	0.71	0.60	0.74	0.99	0.99	0.98

Fonte: O autor (2019)

Tabela 6 - Avaliação de cross-validation utilizando C4.5

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.47	0.49	0.60	0.62	0.76	0.72
Hurricanes1,4	0.72	0.76	0.74	0.78	0.89	0.87
Hurricane0,45	0.83	0.83	0.81	0.85	0.91	0.90
Animals	0.74	0.83	0.81	0.96	0.96	0.95
Vehicle	0.85	0.94	0.92	0.98	0.99	0.99

Fonte: O autor (2019)

Tabela 7 - Avaliação de cross-validation utilizando SVM

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.50	0.50	0.52	0.75	0.78	0.74
Hurricanes1,4	0.72	0.77	0.78	0.86	0.89	0.88
Hurricane0,45	0.85	0.85	0.86	0.90	0.92	0.91
Animals	0.89	0.74	0.79	0.87	0.89	0.88
Vehicle	0.94	0.84	0.98	0.98	0.98	0.97

Fonte: O autor (2019)

Com base nos resultados demonstrados nas tabelas, é possível observar que o algoritmo proposto apresentou o melhor resultado de classificação para 73% das execuções e segundo melhor resultado para 27% das execuções. Comparando a diferença percentual entre o algoritmo proposto e o Movelets, os resultados do método ficaram em média 3.6 pontos percentuais abaixo para o classificador Bayes, e para os classificadores C4.5 e SVM, os resultados se mostraram em média 4.2 pontos acima. acurácia máximo de 18.42% no dataset Hurricanes2,3 para o classificador C4.5

No que diz respeito a tempo de execução, a Tabela 8 demonstra que o Movelets foi em média 29.65 vezes mais rápido do que o AG nos 4 primeiros *datasets*, e 4.97 vezes mais lento para o último *dataset*.

Tabela 8 - Tempo de execução da avaliação com cross-validation

Dataset	Movelets	AG/Bayes	AG/C4.5	AG/SVM
Hurricanes2,3	00:00:11	00:05:59	00:10:31	00:08:34
Hurricanes1,4	00:00:16	00:08:34	00:11:41	00:09:06
Hurricane0,45	00:00:27	00:10:30	00:16:18	00:12:12
Animals	00:01:24	00:10:42	00:11:10	00:09:26
Vehicle	09:18:25	01:54:18	01:38:58	02:03:57

Fonte: O autor (2019)

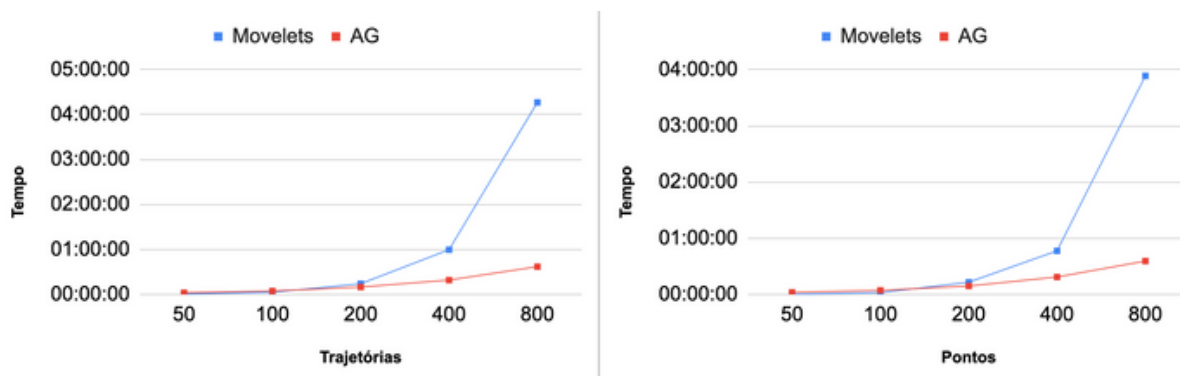
5.3 ANÁLISE DE ESCALABILIDADE

Com base nos resultados coletados, é possível observar que o método proposto apresenta tempo de execução superior ao do Movelets em 4 dos 5 datasets analisados, porém o AG consegue ser superior no conjunto de dados de Veículos, que é o maior dos conjuntos analisados.

A fim de realizar melhor um comparativo de escalabilidade e tempo de execução dos métodos foram utilizados *datasets* sintéticos gerados computacionalmente, variando sua quantidade de trajetórias e quantidade de pontos por trajetória, e os dois algoritmos foram comparados levando em consideração somente o tempo de execução para estes *datasets*.

Para a execução do AG foram utilizados os mesmos parâmetros definidos para a execução dos comparativos de acurácia citados na seção acima. O Gráfico 4 mostra o aumento do tempo de execução dos dois métodos com o incremento no número de trajetórias e número de pontos por trajetória. Para o Gráfico 4 (a) foi variado a quantidade de trajetórias, utilizando o tamanho fixo de 200 pontos por trajetória. No Gráfico 4 (b) foi variado a quantidade de pontos por trajetória, utilizando 200 trajetórias como valor fixo no 5 testes.

Figura 16 - Gráfico de comparação da escalabilidade do Movelets e do AG utilizando *datasets* sintéticos



Fonte: O autor (2019)

Analisando estes dados, é possível observar que a variação do tempo de execução do Movelets tende a ser exponencial, onde o tempo de execução cresce em escala gradativamente maior do que o aumento da quantidade de pontos ou trajetórias no *dataset*, já o AG proposto demonstrou uma variação muito mais linear, onde é possível observar que o aumento dos dados analisados não surte tanto

impacto no tempo de execução do método, pois os fatores de maior impacto na execução do AG são referentes ao tamanho de indivíduo e quantidade de gerações executadas, como foi possível observar nos experimentos realizados na Seção 5.1.

Desta forma, é possível afirmar que o Movelets tende a performar melhor em conjuntos de dados pequenos, porém quando o volume de dados é maior, o AG proposto tende a ser superior no quesito de tempo de execução.

6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho analisou o método Movelets com o intuito de otimizar seu tempo de execução utilizando Algoritmo Genético (AG). Essa revisão em conjunto com os trabalhos relacionados apresentados serviu de embasamento para o desenvolvimento de um AG capaz de receber um conjunto de trajetórias e encontrar as melhores subtrajetórias representativas (movelets).

Com base em estudos em outros métodos que utilizam AG para otimização e busca do resultado ótimo de execução, foi proposto um método para classificação de trajetórias parametrizável que utiliza os conceitos propostos no artigo de Ferrero et al. (2018) como base.

Os experimentos realizados mostraram que o método proposto apresenta a melhor acurácia em 18 das 30 análises realizadas, e uma redução de tempo de processamento para arquivos com grande volume de pontos e trajetórias, o que prova que o trabalho em questão cumpriu com o seu objetivo, que é de otimizar a execução do método Movelets utilizando a abordagem de Algoritmo Genético.

Mesmo com resultados satisfatórios apresentados, alguns pontos de melhoria foram observados, que podem ser utilizadas como embasamento para trabalhos futuros. Dentre eles, podemos citar a:

Implementação de uma função de geração da população inicial com certo grau de qualidade, pois a seleção da população inicial de forma aleatória pode nem sempre resultar em um bom conjunto de indivíduos, e como o AG trabalha dentro deste escopo de indivíduos iniciais, o resultado final da execução pode não ser bom, ou levar muito tempo para chegar num bom resultado utilizando a população inicial gerada aleatoriamente;

Implementação de um método para manter a diversidade da população, permitindo assim maior variedade dentre os indivíduos. De Jong (1975) propôs um método conhecido com *crowding*, que realiza a substituição de indivíduos na população para fazer com que novos indivíduos substituam

outros similares com menor valor de fitness dentro da população. Este método não foi utilizado na implementação apresentada.

Alteração da implementação para comparação de resultados com o trabalho de Ferrero et al. (2019) chamado MasterMovelets, que é uma evolução do método Movelets que permite a classificação de trajetórias multi-aspecto (múltiplas dimensões).

Implementação de uma função de reparo dos indivíduos resultantes da reprodução, pois após análises da implementação, foi possível observar que o resultado dessa função pode violar o tamanho fixo do indivíduo.

Executar testes utilizando outros conjuntos de parâmetros e operadores genéticos.

REFERÊNCIAS

- BOGORNY, Vania. **Introdução a trajetórias de objetos móveis: conceitos, armazenamento e análise de dados**. Univile, 2012.
- DE JONG, K.A.. **An Analysis of the Behavior of a Class of Genetic Adaptive Systems**.. University of Michigan, 1975.
- DODGE, Somayeh; WEIBEL, Robert; FOROOTAN, Ehsan. **Revealing the physics of movement**: Comparing the similarity of movement characteristics of different types of moving objects. *Computers, Environment and Urban Systems* 33, 2009.
- FERNANDES, A.M.R.. **Inteligência Artificial: noções gerais**. 2. ed. Florianópolis: VisualBooks, 2005.
- FERRERO, Carlos Andres *et al.* **Master Movelets**: Discovering Heterogeneous Movelets for Multiple Aspect Trajectory Classification. *Data Mining and Knowledge Discovery*, 2019.
- FERRERO, Carlos Andres *et al.* **MOVELETS**: Exploring Relevant Subtrajectories for Robust Trajectory Classification. Pau - France: SAC, 2018.
- GIANNOTTI, Fosca; PEDRESCHI, Dino. **Mobility, Data Mining and Privacy: Geographic Knowledge Discovery**. Nova York: Springer, 2008.
- GRABOCKA, J. *et al.* Learning time-series shapelets. *In: 20TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*. ACM, 2014.
- HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data Mining: Concepts and Techniques**. 3. ed. Waltham: Elsevier, 2012.
- LINDEN, Ricardo. **Algoritmos genéticos**. 3. ed. Rio de Janeiro: Ciência Moderna, 2012.
- LINES, J. *et al.* A shapelet transform for time series classification. *In: 18TH ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING*. ACM, 2012.
- MITCHELL, Melaine. **An Introduction to genetic algorithms**. 5. ed. Massachusetts: Bradford Book, 1999.
- POZO, Aurora *et al.* **Computação Evolutiva**. Universidade Federal do Paraná, 2005.
- RAKTHANMANON, T.; KEOGH, E.. Fast shapelets: A scalable algorithm for discovering time series shapelets. *In: SIAM INTERNATIONAL CONFERENCE ON DATA MINING*. SIAM, 2016.
- RIDLEY, Mark. **Evolução**. 3. ed. São Paulo: Artmed Editora SA, 2006.

SILVA, João Paulo Domingos. **Algoritmos de Classificação baseados em Análise Formal de Conceitos**. Belo Horizonte, 2007.

TAN, Pang-ning; STEINBACH, Michael; KUMAR, Vipin. **Introduction to Data Mining**. Boston: Pearson, 2006.

VANDEWIELE, Gilles; ONGENAE, Femke; DE TURCK, Filip. **GENDIS: GENetic Discovery of Shapelets**. Massachusetts Institute of Technology, 2019.

XIAO, Zhibin *et al.* **Identifying Dierent Trans- portation Modes from Trajectory Data Using Tree-Based Ensemble Classiers**. ISPRS International Journal of Geo-Information, 2017.

YE, Lexiang; KEOGH, Eamonn. **Time Series Shapelets: A New Primitive for Data Mining**. ACM, 2009.

ZHENG, Yu *et al.* **Understanding transportation modes based on GPS data for web applications**. ACM Transactions on the Web (TWEB), 2010.

APÊNDICE A — CÓDIGO-FONTE

O código do algoritmo desenvolvido e apresentado neste trabalho está disponível na internet através do repositório GitHub no link: <<https://github.com/JackDaniells/moveletes-genetic-algorithm>>.

A instalação e execução do algoritmo está descrita no no arquivo README.md

APÊNDICE B — ARTIGO

Uso de Algoritmo Genético para otimização de seleção de subtrajetórias relevantes para classificação

Daniel H. Kock¹, Luis O. C. Alvares², Vania Bogorny²,
Camila L. da Silva³, Rafael de Santiago³

¹Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Campus Universitário – Trindade – Caixa Postal 476
CEP 88010-970 – Florianópolis – Santa Catarina – SC – Brazil

danielk.kock@gmail.com, {vania.bogorny, luis.alvares}@inf.ufsc.br

Abstract. *With the current advancement of trajectory data collection technologies such as GPS and smartphones, we have a larger amount of trajectory data each day, and due to the increasing use of information in this context, it is important to analyze this data set. -temporal in order to add value to this data. In this work, we studied a method of selection of relevant sub trajectories called Movelets, which seeks to determine frequent sub trajectories strictly in a specific class, but its execution is very computationally costly. The ultimate goal was to propose and implement a Movelets-based genetic algorithm solution that finds relevant sub-trajectories in a shorter processing time while maintaining search accuracy. The results obtained show that the proposed implementation was able to present good accuracy and runtime results, surpassing Movelets in accuracy for 18 out of 30 tests performed and runtime for large datasets.*

Resumo. *Com o atual avanço de tecnologias para a coleta de dados de trajetória, tais como GPS e smartphones, temos cada dia uma maior quantidade de dados de trajetórias, e devido ao crescente uso de informações neste contexto, é importante a análise deste conjunto de dados espaço-temporais a fim de agregar valor a estes dados. Neste trabalho, estudou-se um método de seleção de subtrajetórias relevantes chamado Movelets, que procura determinar subtrajetórias frequentes estritamente em uma classe específica, porém sua execução é muito custosa computacionalmente. O objetivo final foi propor e implementar uma solução baseada algoritmo genético baseada no Movelets, que encontre as subtrajetórias relevantes em um menor tempo de processamento mantendo a acurácia nas buscas. Os resultados obtidos mostram que a implementação proposta conseguiu apresentar bons resultados de acurácia, e de tempo de execução, superando o Movelets em acurácia para 18 dos 30 testes executados e em tempo de execução para grandes datasets.*

1. Palavras-chave

Trajetórias; Classificação; Movelets; Inteligência Artificial; Algoritmo Genético

2. Introdução

Trajetórias são formadas por sequências de pontos registrados para cada indivíduo correspondentes a sua localização em determinado momento do tempo [Bogorny 2012]. Os

pontos representam informações a respeito do objeto móvel, e normalmente são representados por coordenadas geográficas, formando o que é conhecido como trajetória bruta.

Os dados de movimento de objetos móveis só serão realmente úteis se forem analisados e essa análise resultar em conhecimento. Para a análise de trajetórias, uma das formas mais utilizadas é a classificação, que corresponde a identificar a classe de trajetórias. A classe pode ser o meio de transporte utilizado na trajetória, como por exemplo, ônibus, carro ou táxi, a identificação do usuário de uma trajetória de redes sociais, a qual tipo de animal corresponde uma trajetória, etc [Bogorny 2012].

Um dos métodos de classificação de trajetórias existente na literatura é o método Movelets [Ferrero et al. 2018], que faz uso da comparação exaustiva de subtrajetórias para obter as partes da trajetória que melhor identificam a classe em comparação com as outras classes. Esse método se mostrou muito eficiente para a classificação de dados de trajetória, superando outros métodos utilizados para a mesma finalidade. Porém sua implementação exige um processamento computacional muito grande, tornando praticamente inviável a sua execução em grandes conjuntos de dados.

Para resolver este problema, uma possível solução seria a utilização de meta-heurísticas para a otimização da busca destas subtrajetórias relevantes. O objetivo deste trabalho é a análise e implementação de uma destas técnicas de modo a reduzir o consumo de recursos computacionais em relação ao método Movelets, e analisar os resultados obtidos em comparação a implementação atual do método.

A abordagem escolhida para a resolução do problema é a técnica de Algoritmo Genético, um método que busca encontrar a melhor solução para os problemas, utilizando um processo de busca iterativa da melhor solução para o problema em questão, que parte de uma população inicial e tende a manter os melhores resultados para as populações posteriores [Fernandes 2005].

O trabalho de [Vanderwiele et al. 2019] por exemplo, utiliza algoritmos genéticos para encontrar shapelets em séries temporais, que é um problema similar ao de encontrar as melhores subtrajetórias para a classificação de trajetórias. Os resultados mostraram que a solução proposta apresenta uma complexidade computacional inferior quando comparada com métodos de busca que utilizam força bruta, e com resultados muito próximos do melhor método de busca comparado, o que demonstra a viabilidade do uso desta abordagem para a solução do problema em questão.

3. Trabalhos Relacionados

Os trabalhos existentes sobre classificação de trajetória consistem em extrair características das trajetórias. O método proposto por [Zheng et al. 2010] utiliza somente atributos globais para a classificação das trajetórias, que demonstraram bons resultados para a classificação de trajetórias, porém esta abordagem possui limitações, visto que um objeto móvel pode variar sua velocidade durante o percurso, e essa variação não é abordada por este método. O método de [Dodge et al. 2009], que corresponde a uma proposta para extrair características locais transformando uma trajetória em uma série temporal com atributos para cada ponto. Estas séries temporais são discretizadas, a fim de diminuir a complexidade dos dados, porém a limitação desta abordagem é a perda de informações neste processo de discretização. A proposta de [Xiao et al. 2017], que corresponde a

um método que utiliza características globais gerados por um método estatístico e características locais extraídos por segmentação das trajetórias analisadas, que são combinadas para obter melhores resultados de classificação. [Ferrero et al. 2018] propõem em seu artigo um método de descoberta de subtrajetórias relevantes baseado no conceito de Shapelets, que encontra subtrajetórias relevantes, fazendo a comparação por força bruta entre todas as subtrajetórias possíveis, que depois são avaliadas, e as melhores são selecionadas para a classificação. Este se mostrou muito eficaz para a classificação de trajetórias, apresentado ótimos valores de acurácia de classificação, porém a comparação por força bruta torna sua execução extremamente custosa em questão de tempo computacional. O artigo de [Vanderwiele et al. 2019] por outro lado, propõe a criação de um modelo para a descoberta de shapelets baseado em algoritmo genético, trazendo como vantagens a facilidade de encontrar candidatos adequados, redução de complexidade computacional de execução e definição de parâmetros em tempo de execução.

4. Método proposto

A proposta deste artigo é a implementação de um método que tem como objetivo processar um dataset de trajetórias de modo a encontrar as subtrajetórias mais relevantes para classificação utilizando a implementação de AG para encontrar melhores subtrajetórias relevantes no decorrer das gerações.

Os indivíduos são formados por conjunto de genes, que correspondem a uma subtrajetória com n pontos pertencentes a uma trajetória t do conjunto de dados. Um indivíduo pode possuir subtrajetórias de diferentes classes, que depois vão ser avaliadas a fim de encontrar o fitness deste indivíduo.

Os parâmetros de tamanho de população, de indivíduos e a quantidade de gerações utilizadas são variáveis configuradas na inicialização do método. A inicialização da população é feita de forma aleatória de modo a aumentar a diversidade da população inicial.

A Figura 1 representa a execução do AG, onde o método primeiro calcula o fitness de cada indivíduo utilizando um classificador e ranqueia os indivíduos com base na acurácia de classificação obtida. A seleção dos indivíduos usa uma função de Elitismo para garantir os melhores na próxima geração e Torneio ou Roleta para a escolha dos indivíduos que participarão da reprodução. Os indivíduos selecionados são inseridos em uma função de crossover, que retorna um indivíduo com genes provindos dos pais selecionados, e ainda é aplicada uma função de mutação, onde ocorre a alteração dos genes de um indivíduo selecionado aleatoriamente.

No final da execução do método, é retornado o valor de acurácia de classificação do melhor indivíduo, que representa o valor final de acurácia para a execução em questão.

4.1. Definições básicas

Como definições básicas deste artigo, temos:

Ponto: representa um ponto de uma trajetória. Contém as coordenadas (x e y) do ponto e o momento que foi coletado (time).

Trajetória: representa uma trajetória. Contém um conjunto de pontos (points), o informativo de qual a classe dessa trajetória (group), e atributos globais como duração

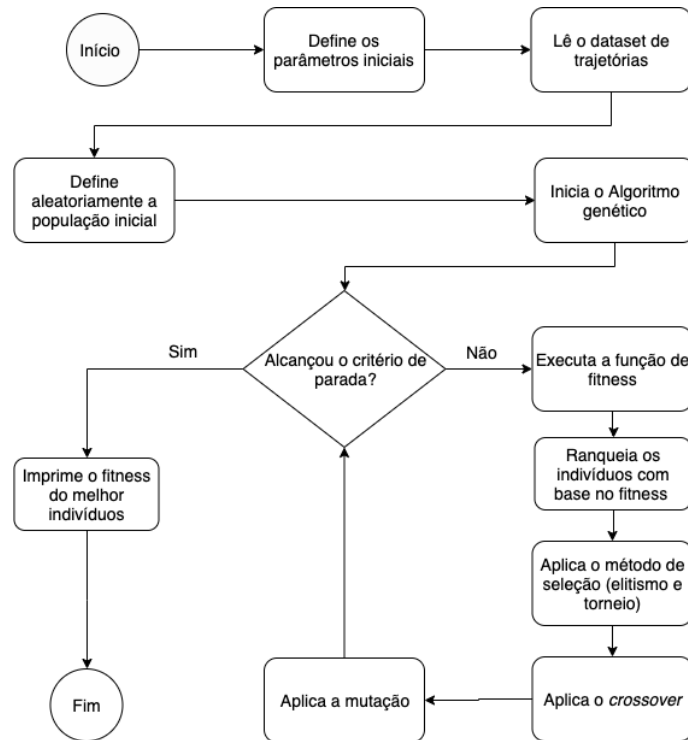


Figura 1. Diagrama de execução do algoritmo proposto

(duration), distância percorrida (distance) e velocidade média (avgSpeed).

Gene: equivalente à um candidato a subtrajetória representativa descrito no trabalho de [Ferrero et al. 2018]. Contém uma referência a qual trajetória pertence (trajectory), a quantidade de pontos que a subtrajetória possui (size) e o seu ponto de início na trajetória (start), como exemplificado na Figura 2. Para fins de otimização da execução, é salva a matriz de distâncias entre o Gene e todas as trajetórias do dataset (distances), que são os valores utilizados de base para o cálculo de fitness de um indivíduo.

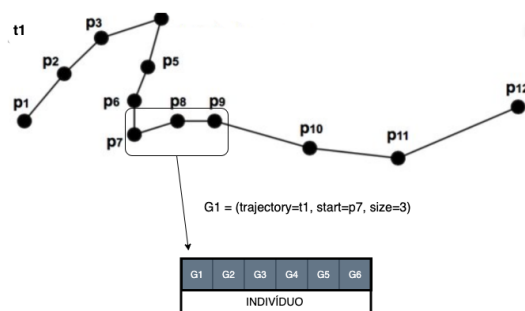


Figura 2. Exemplo de gene

Indivíduo: é definido como um conjunto de genes, e também possui o atributo score, que representa o seu valor de fitness.

População: Equivale a um conjunto de indivíduos, possui o método de geração da população inicial em seu escopo.

Fitness: A função de fitness consiste no resultado da acurácia obtida a partir da classificação do indivíduo com base nas trajetórias do conjunto de dados.

Uma matriz de distâncias é calculada, onde cada linha corresponde a uma trajetória e cada coluna corresponde a um gene de um determinado indivíduo. As células da matriz contém a distância entre o gene e a trajetória, cujo o valor corresponde a menor distância entre o indivíduo e todas as subtrajetórias de mesmo tamanho da trajetória comparada. Adicionalmente são também incluídos para cada trajetória três atributos globais à esta matriz: a distância total, o tempo de duração e velocidade média da trajetória (distância / tempo). E finalmente, como última coluna, temos a classe de cada trajetória.

O classificador utiliza a matriz fornecida para gerar um modelo, que é aplicado em uma matriz-teste, e o resultado da acurácia deste modelo resulta no valor de fitness do indivíduo, como representado na Figura 3.

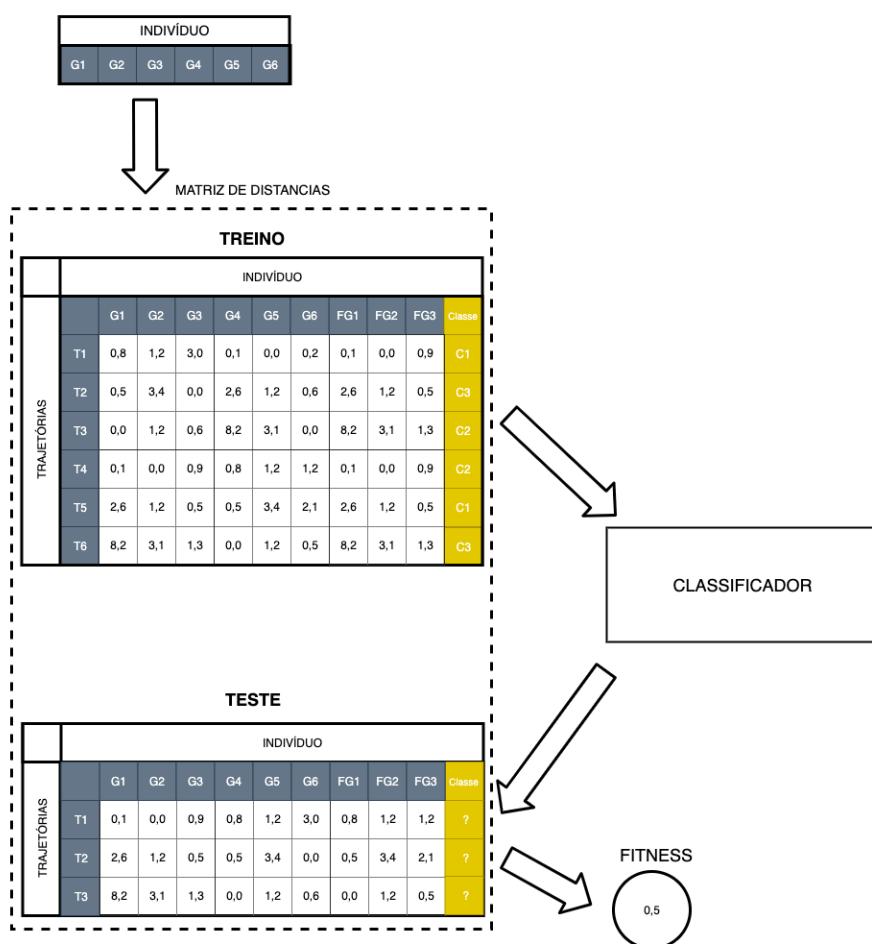


Figura 3. Exemplo de cálculo do valor de fitness de um indivíduo

Elitismo: Para evitar a perda das melhores soluções já encontradas foi utilizada a técnica de elitismo para garantir que o melhor indivíduo continue na população. A cada nova geração é mantido um único indivíduo com o melhor valor de fitness da geração anterior para a nova geração.

Seleção: De modo a encontrar os indivíduos com os melhores genes para a

reprodução, é utilizado o método de seleção por torneio, que seleciona de forma aleatória um conjunto de indivíduos que irão competir entre si com as suas avaliações.

Mutação: A mutação acontece com base no parâmetro de taxa de mutação, que equivale a um percentual que representa a probabilidade do indivíduo sofrer alteração na sua carga genética. Uma função probabilística é executada para selecionar se o indivíduo vai sofrer mutação ou não. Caso ele tenha sido selecionado, uma segunda função é responsável por selecionar o gene que sofrerá a mutação, e uma terceira função é responsável por selecionar o tipo de mutação que irá acontecer. Neste processo, o gene pode ser totalmente recriado, ou somente sofrer alterações na sua trajetória de referência, ponto de início ou tamanho.

Critério de parada: Como critério de parada, o método proposto utiliza dois parâmetros, quantidade de gerações, onde ele encerra a execução quando atinge um número de gerações pré-estabelecido, ou quantidade de gerações sem convergência, onde o método encerra a execução quando atinge uma quantidade de gerações em que não obteve um indivíduo com valor de fitness maior do que nas gerações anteriores.

5. Experimentos

A fim de testar o algoritmo desenvolvido, inicialmente foram realizados testes para encontrar a melhor combinação de parâmetros de execução. Após isso, o método foi executado com os mesmos datasets descritos na seção 3.1 e utilizados no trabalho de [Ferrero et al. 2018] e os resultados foram comparados com os resultados dos métodos citados no artigo que propõe o método Movelets. Um comparativo de tempo computacional entre o Movelets e o AG proposto também foi executado.

Para a execução dos experimentos, foi utilizado um Macbook Pro com processador Intel Core I7 1.7GHz, 16GB de memória e sistema operacional macOS.

Para encontrar a melhor combinação de parâmetros de execução do AG utilizando datasets de trajetórias, foram executados testes levando em consideração o tempo e a acurácia de execução do método. Estes valores foram representados em gráficos, e no final, uma análise ponderada entre estes dois valores coletados foi feita, a fim de encontrar o parâmetro com melhor acurácia no menor tempo de execução, visto que a proposta deste trabalho é otimizar o tempo de execução do Movelets mantendo bons resultados de acurácia.

Os gráficos a seguir representam os resultados obtidos variando os parâmetros a fim de encontrar o melhor custo-benefício, onde a linha azul representa a evolução da acurácia do método e a vermelha representa a evolução do tempo de execução, representado no formato hh:mm:ss. Todos os resultados foram coletados variando somente um parâmetro por vez.

5.1. Tamanho da população

A primeira avaliação foi executada variando o tamanho da população, começando com 10 indivíduos por geração e variando até 160 indivíduos. A Figura 4 apresenta os resultados obtidos.

Analisando os resultados, é possível observar que a variação do tamanho da população gerou uma variação de acurácia de 4 pontos percentuais entre a pior e a melhor

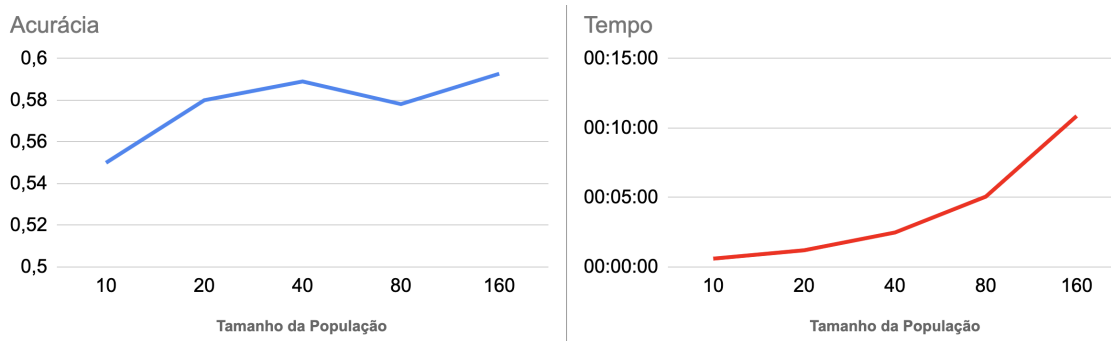


Figura 4. Acurácia e Tempo de execução variando o tamanho da população

execução (0,55 com população de tamanho 10 e 0,59 com população de tamanho 160), e a maior evolução da acurácia foi notada entre as 3 primeiras execuções do método. Já o tempo de execução tendeu a ter um aumento médio de 106,36% entre as execuções, o que representa que a variação deste parâmetro tem um grande impacto no tempo final de execução do método.

5.2. Tamanho dos indivíduos

Na segunda avaliação, alterou-se a quantidade de genes do indivíduo. Para a execução deste testes foi utilizado como valor de referência a quantidade de classes do dataset analisado, e o valor de tamanho do indivíduo deve ser múltiplo à referência (para um dataset com 4 classes, 2x representa que o tamanho do indivíduo é igual a 8, por exemplo). Foi escolhida esta forma de representação do indivíduo pois o número de classes tende a variar de um dataset para outro, e é esperado que os indivíduos tenham genes de referência para todas as classes existentes.

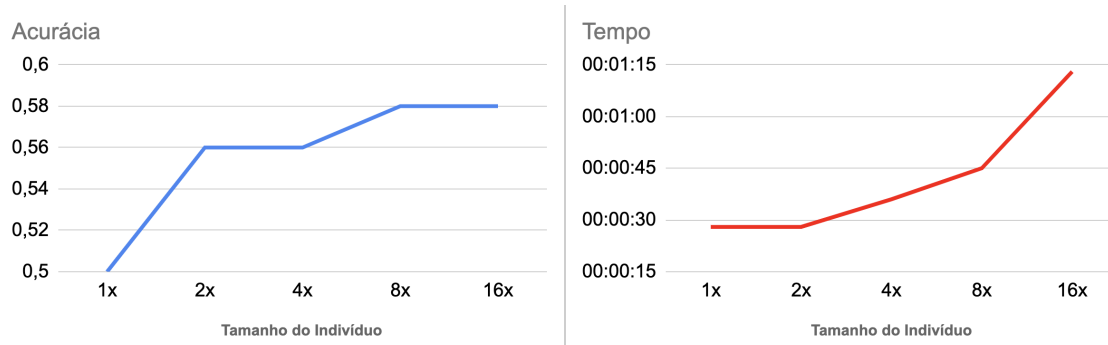


Figura 5. Acurácia e Tempo de execução variando o tamanho da indivíduo

Nesta avaliação demonstrada na Figura 5, foi observado um ganho de 8 pontos percentuais entre a pior e a melhor execução (0,50 com tamanho do indivíduo igual a 1x e 0,58 com tamanho do indivíduo igual a 16x). Também é possível observar que o ganho de acurácia aconteceu nos três primeiros testes, e depois estabilizou, já o tempo teve um crescimento médio de 28,95%, variando de 28 segundos na primeira execução até 1 minuto e 13 segundos na última execução.

5.3. Quantidade de gerações

Na terceira avaliação, alterou-se a quantidade de gerações executadas pelo algoritmo genético, variando de 20 até 320 gerações. A Figura 6 mostra os resultados obtidos.

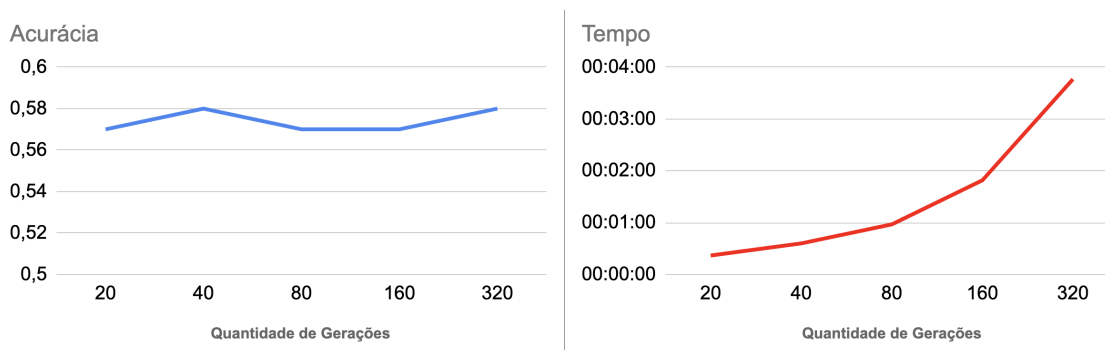


Figura 6. Acurácia e Tempo de execução variando a quantidade de gerações

Como é possível observar, a acurácia se manteve estável nos testes, variando 1 ponto percentual em média, já o tempo de execução tendeu a um aumento médio de 80% entre as execuções.

5.4. Taxa de mutação

Nesta avaliação, alterou-se a taxa de mutação, variando seu valor de 2,5% até 40%. A Figura 7 demonstra os resultados obtidos durante as execuções.

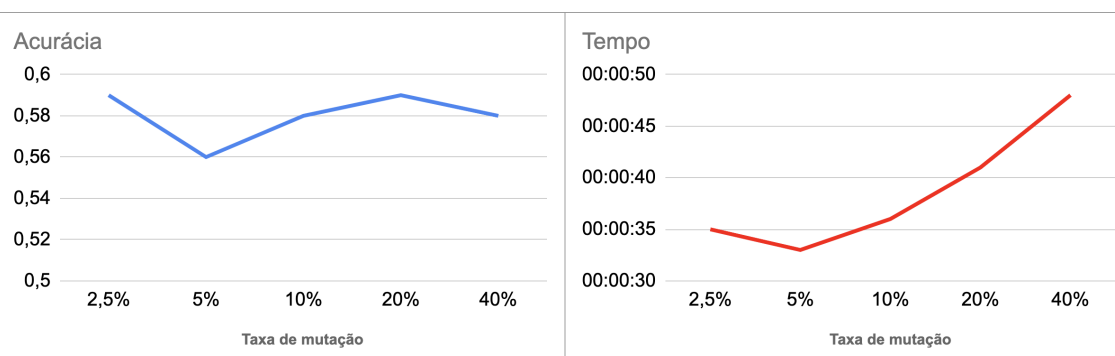


Figura 7. Acurácia e Tempo de execução variando a taxa de mutação

Na Figura 7, é possível observar que a acurácia caiu e depois tendeu a subir novamente, oscilando 3 pontos percentuais entre as execuções, já o tempo tendeu a um aumento de 8,58% entre as execuções, menor média obtida entre os parâmetros observados.

5.5. Definição dos parâmetros utilizados nas comparações

Com base nos resultados coletados nas avaliações, foram definidos os parâmetros para a execução dos testes comparativos, e os critérios utilizados para escolhas foi uma ponderação entre o ganho de acurácia e de tempo de execução. Os parâmetros definidos foram:

(i) tamanho da população de 20 indivíduos: Neste caso, tivemos o maior ganho de acurácia entre população de tamanho 10 e tamanho 20, e como a variação deste parâmetro é muito custosa em tempo computacional para a execução (106,36%), a opção selecionada foi a que fornecia maior ganho com menor custo

(ii) tamanho de indivíduos igual a 8 vezes a quantidade de classes do dataset: Para esta hipótese, selecionamos o parâmetro que forneceu maior ganho nos testes, pois a variação deste parâmetro tem custo relativamente baixo (28,94%).

(iii) quantidade de gerações igual a 40 gerações sem convergência: Neste caso, inserimos um parâmetro de análise de gerações sem convergência, pois como a variação da acurácia nos resultados não foi significativa, foi concluído que o parâmetro para fim da execução seria de gerações que não resultam no aumento da acurácia entre uma geração e outra e pode se encerrar mais rapidamente caso não exista mais convergência.

(iv) taxa de mutação inicial de 2,5%: Para este parâmetro, decidimos utilizar a menor taxa de mutação testada como parâmetro inicial, porém a variação deste parâmetro é importante para contribuir com a diversidade das populações no decorrer da execução do método, e por isso, foi definido um fator de multiplicação que dobra a taxa de mutação a cada 10 gerações sem convergência, assim é possível aproveitar os indivíduos gerados aleatoriamente no início da execução, e depois de certo tempo sem convergência, o aumento na taxa de mutação busca contribuir para o aumento de diversidade.

6. Comparativos com outros métodos

Com base nos parâmetros definidos na sessão anterior, foram executados testes comparativos entre os resultados do AG e os métodos de [Ferrero et al. 2018], [Zheng et al. 2010], [Dodge et al. 2009] e [Xiao et al. 2017], que foram utilizados como base comparativa no artigo do Movelets. Para execução dos modelos e geração do fitness dos indivíduos, foram utilizados os algoritmos de classificação Naive Bayes, C4.5 e SVM com avaliação por abordagens de holdout e cross-validation.

Os 5 datasets utilizados para a execução das análises são os mesmos utilizados nas comparações de [Ferrero et al. 2018], e são relacionados a trajetórias de furacões (Hurricane2,3, Hurricane1,4 e Hurricane0,45), animais (Animals) e veículos (Vehicles).

Para a realização destes testes com o intuito de análise comparativa com outros métodos de classificação de trajetórias, foi utilizado a mesma função de média da acurácia de 5 execuções do AG (AG médio), porém também apresentamos o valor de acurácia da melhor execução (AG melhor).

6.1. Holdout

Na realização dos experimentos com holdout, os datasets foram divididos em conjunto de treinamento (60%) e conjunto de teste (40%). Nesta modalidade, o AG utiliza o conjunto de dados de treino para a execução do AG, e o melhor indivíduo da última geração executada é utilizado como treino do classificador, que usa o conjunto de teste para gerar o resultado final da avaliação.

As Figuras 8, 9 e 10 mostram os resultados da avaliação de holdout para a classificação utilizando Naive Bayes, C4.5 e SVM. Os melhores resultados da

classificação para cada conjunto de dados são destacados em negrito e os em segundo melhores estão sublinhados, para facilitar as comparações.

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.54	0.56	0.54	0.60	<u>0.59</u>	0.56
Hurricane1,4	0.72	0.68	0.68	0.80	<u>0.77</u>	0.74
Hurricane0,45	0.76	0.86	0.81	<u>0.84</u>	0.81	0.78
Animals	0.63	0.64	0.76	0.93	<u>0.81</u>	0.81
Vehicle	0.76	0.47	0.81	0.97	<u>0.85</u>	0.79

Figura 8. Avaliação de holdout utilizando Bayes

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.39	<u>0.62</u>	0.51	<u>0.62</u>	0.67	0.60
Hurricane1,4	0.81	0.71	0.76	<u>0.85</u>	0.86	0.81
Hurricane0,45	<u>0.84</u>	0.85	0.82	0.83	0.83	0.80
Animals	0.67	<u>0.76</u>	0.74	0.93	0.93	0.87
Vehicle	0.73	0.90	0.94	<u>0.96</u>	0.98	0.97

Figura 9. Avaliação de holdout utilizando C4.5

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricane2,3	0.56	0.59	0.53	<u>0.60</u>	0.65	0.61
Hurricane1,4	0.79	0.75	0.77	0.85	<u>0.83</u>	0.81
Hurricane0,45	<u>0.87</u>	<u>0.87</u>	0.86	0.88	<u>0.87</u>	0.84
Animals	0.67	0.68	0.81	0.90	<u>0.89</u>	0.89
Vehicle	0.89	0.78	<u>0.98</u>	0.99	0.92	0.89

Figura 10. Avaliação de holdout utilizando SVM

Analisando os resultados apresentados nestas tabelas, observa-se que o método proposto apresentou melhor acurácia para 33% dos casos e segundo melhor resultado para 47% das execuções. Em comparação com o Movelets, os resultados mostram que, para os classificadores Bayes e SVM, o método proposto apresentou resultados em média 3.3 pontos percentuais abaixo do Movelets. Já com o classificador C4.5, os resultados ficaram em média 1.6 pontos percentuais acima dos resultados do Movelets.

No que diz respeito ao tempo de execução entre o Movelets e o algoritmo proposto, a Figura 11 demonstra o tempo de execução para cada dataset, e é possível observar que o tempo do Movelets foi em média 12.1 vezes mais veloz nos 4 primeiros datasets, e 5.7 vezes mais lento para o último dataset.

Dataset	Movelets	AG/Bayes	AG/C4.5	AG/SVM
Hurricane2,3	00:00:08	00:02:01	00:02:11	00:00:50
Hurricane1,4	00:00:10	00:04:32	00:02:23	00:01:07
Hurricane0,45	00:00:14	00:03:34	00:03:36	00:01:46
Animals	00:00:38	00:04:59	00:02:50	00:05:46
Vehicle	04:08:47	01:17:54	00:27:43	00:24:24

Figura 11. Tempo de execução da avaliação com holdout

6.2. Cross-validation

Para a realização dos experimentos utilizando cross-validation, foi utilizado a divisão do dataset em 5 partes (folds). Como tanto para o Movelets quanto para o AG proposto neste trabalho, o método tem que ser reexecutado a cada vez que os dados de treino são alterados, o cross-validation foi executado “manualmente”, executando-se 5 vezes os métodos, e os dados apresentados correspondem à média da acurácia e do tempo de execução dessas 5 execuções.

As Figuras 12, 13 e 14 mostram os resultados da avaliação, e a Figura 15 mostra o comparativo do tempo de execução do Movelets com o AG.

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.53	0.55	0.48	0.76	<u>0.67</u>	0.65
Hurricanes1,4	0.70	0.70	0.66	0.86	<u>0.82</u>	0.81
Hurricane0,45	0.80	0.82	0.78	0.87	<u>0.84</u>	0.83
Animals	0.51	0.70	0.77	0.91	<u>0.89</u>	0.88
Vehicle	0.71	0.60	<u>0.74</u>	0.99	0.99	0.98

Figura 12. Avaliação de cross-validation utilizando Bayes

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.47	0.49	0.60	<u>0.62</u>	0.76	<u>0.72</u>
Hurricanes1,4	0.72	0.76	0.74	<u>0.78</u>	0.89	0.87
Hurricane0,45	0.83	0.83	0.81	<u>0.85</u>	0.91	0.90
Animals	0.74	<u>0.83</u>	0.81	0.96	0.96	0.95
Vehicle	0.85	0.94	0.92	<u>0.98</u>	0.99	0.99

Figura 13. Avaliação de cross-validation utilizando C4.5

Com base nos resultados demonstrados nas tabelas, é possível observar que o algoritmo proposto apresentou o melhor resultado de classificação para 73% das execuções

Dataset	Dodge	Zheng	Xiao	Movelets	AG(melhor)	AG(média)
Hurricanes2,3	0.50	0.50	0.52	<u>0.75</u>	0.78	0.74
Hurricanes1,4	0.72	0.77	0.78	<u>0.86</u>	0.89	0.88
Hurricane0,45	0.85	0.85	0.86	<u>0.90</u>	0.92	0.91
Animals	0.89	0.74	0.79	<u>0.87</u>	0.89	0.88
Vehicle	0.94	0.84	0.98	0.98	0.98	0.97

Figura 14. Avaliação de cross-validation utilizando SVM

e segundo melhor resultado para 27% das execuções. Comparando a diferença percentual entre o algoritmo proposto e o Movelets, os resultados do método ficaram em média 3.6 pontos percentuais abaixo para o classificador Bayes, e para os classificadores C4.5 e SVM, os resultados se mostraram em média 4.2 pontos acima. acurácia máximo de 18.42% no dataset Hurricanes2,3 para o classificador C4.5

No que diz respeito a tempo de execução, a Figura 15 demonstra que o Movelets foi em média 29.65 vezes mais rápido do que o AG nos 4 primeiros datasets, e 4.97 vezes mais lento para o último dataset.

Dataset	Movelets	AG/Bayes	AG/C4.5	AG/SVM
Hurricanes2,3	00:00:11	00:05:59	00:10:31	00:08:34
Hurricanes1,4	00:00:16	00:08:34	00:11:41	00:09:06
Hurricane0,45	00:00:27	00:10:30	00:16:18	00:12:12
Animals	00:01:24	00:10:42	00:11:10	00:09:26
Vehicle	09:18:25	01:54:18	01:38:58	02:03:57

Figura 15. Tempo de execução da avaliação com cross-validation

7. Análise de escalabilidade

Com base nos resultados coletados, é possível observar que o método proposto apresenta tempo de execução superior ao do Movelets em 4 dos 5 datasets analisados, porém o AG consegue ser superior no conjunto de dados de Veículos, que é o maior dos conjuntos analisados.

A fim de realizar melhor um comparativo de escalabilidade e tempo de execução dos métodos foram utilizados datasets sintéticos gerados computacionalmente, variando sua quantidade de trajetórias e quantidade de pontos por trajetória, e os dois algoritmos foram comparados levando em consideração somente o tempo de execução para estes datasets.

Para a execução do AG foram utilizados os mesmos parâmetros definidos para a execução dos comparativos de acurácia citados na seção acima. A Figura 16 mostra o aumento do tempo de execução dos dois métodos com o incremento no número de

trajetórias e número de pontos por trajetória. Para a Figura 16 (a) foi variado a quantidade de trajetórias, utilizando o tamanho fixo de 200 pontos por trajetória. Na Figura 16 (b) foi variado a quantidade de pontos por trajetória, utilizando 200 trajetórias como valor fixo no 5 testes.

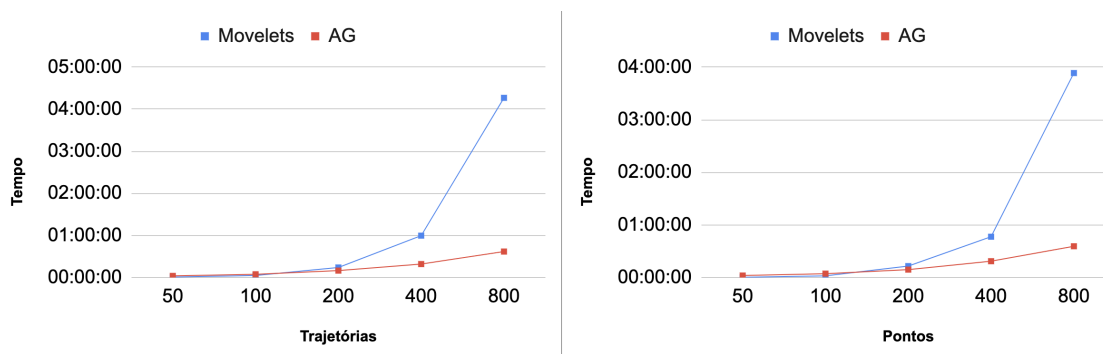


Figura 16. Gr fico de compara o da escalabilidade do Movelets e do AG utilizando datasets sint ticos

Analisando estes dados,   poss vel observar que a varia o do tempo de execu o do Movelets tende a ser exponencial, onde o tempo de execu o cresce em escala gradativamente maior do que o aumento da quantidade de pontos ou trajet rias no dataset, j  o AG proposto demonstrou uma varia o muito mais linear, onde   poss vel observar que o aumento dos dados analisados n o surte tanto impacto no tempo de execu o do m todo, pois os fatores de maior impacto na execu o do AG s o referentes ao tamanho de indiv duo e quantidade de gera es executadas, como foi poss vel observar nos experimentos realizados na Se o 5.1.

Desta forma,   poss vel afirmar que o Movelets tende a performar melhor em conjuntos de dados pequenos, por m quando o volume de dados   maior, o AG proposto tende a ser superior no quesito de tempo de execu o.

8. Conclus es e trabalhos futuros

O presente trabalho analisou o m todo Movelets com o intuito de otimizar seu tempo de execu o utilizando Algoritmo Gen tico (AG). Essa revis o em conjunto com os trabalhos relacionados apresentados serviu de embasamento para o desenvolvimento de um AG capaz de receber um conjunto de trajet rias e encontrar as melhores subtrajet rias representativas (movelets).

Com base em estudos em outros m todos que utilizam AG para otimiza o e busca do resultado  timo de execu o, foi proposto um m todo para classifica o de trajet rias parametriz vel que utiliza os conceitos propostos no artigo de [Ferrero et al. 2018] como base.

Os experimentos realizados mostraram que o m todo proposto apresenta a melhor acur cia em 18 das 30 an lises realizadas, e uma redu o de tempo de processamento para arquivos com grande volume de pontos e trajet rias, o que prova que o trabalho em quest o cumpriu com o seu objetivo, que   de otimizar a execu o do m todo Movelets utilizando a abordagem de Algoritmo Gen tico.

Mesmo com resultados satisfatórios apresentados, alguns pontos de melhoria foram observados, que podem ser utilizadas como embasamento para trabalhos futuros. Dentre eles, podemos citar a:

- Implementação de uma função de geração da população inicial com certo grau de qualidade, pois a seleção da população inicial de forma aleatória pode nem sempre resultar em um bom conjunto de indivíduos, e como o AG trabalha dentro deste escopo de indivíduos iniciais, o resultado final da execução pode não ser bom, ou levar muito tempo para chegar num bom resultado utilizando a população inicial gerada aleatoriamente;
- Implementação de um método para manter a diversidade da população, permitindo assim maior variedade dentre os indivíduos. [Jong 1975] propôs um método conhecido com crowding, que realiza a substituição de indivíduos na população para fazer com que novos indivíduos substituam outros similares com menor valor de fitness dentro da população. Este método não foi utilizado na implementação apresentada.
- Alteração da implementação para comparação de resultados com o trabalho de [Ferrero et al. 2019] chamado MasterMovelets, que é uma evolução do método Movelets que permite a classificação de trajetórias multi-aspecto (múltiplas dimensões).
- Implementação de uma função de reparo dos indivíduos resultantes da reprodução, pois após análises da implementação, foi possível observar que o resultado dessa função pode violar o tamanho fixo do indivíduo.
- Executar testes utilizando outros conjuntos de parâmetros e operadores genéticos.

Referências

- Bogorny, V. (2012). *Introdução a trajetórias de objetos móveis: conceitos, armazenamento e análise de dados*. Univile.
- Dodge, S., Weilbel, R., and Forootan, E. (2009). Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects.
- Fernandes, A. M. R. (2005). *Inteligência Artificial: noções gerais*. VisualBooks.
- Ferrero, C. A., Bogorny, V., Alvares, L. O., and Zalewski, W. (2018). Movelets: Exploring relevant subtrajectories for robust trajectory classification.
- Ferrero, C. A., Bogorny, V., Alvares, L. O., and Zalewski, W. (2019). Master movelets: Discovering heterogeneous movelets for multiple aspect trajectory classification.
- Jong, K. D. (1975). An analysis of the behavior of a class of genetic adaptive systems.
- Vanderwiele, G., Ongenaes, F., and Turck, F. D. (2019). Gendis: Genetic discovery of shapelets.
- Xiao, Z., Wang, Y., Fu, K., and Wu, F. (2017). Identifying different transportation modes from trajectory data using tree-based ensemble classifiers.
- Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y. (2010). Understanding transportation modes based on gps data for web applications.