

Modelo Logístico para *Map-Matching Online* de Trajetória de Veículos

Juan Camilo Fonseca-Galindo¹, José Maia Neto²,
Cristiano Leite de Castro³, André Paim Lemos⁴, and Antônio Pádua Braga⁵

Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil

¹ juankmilofg@ufmg.br,² jmnt@ufmg.br,³ crislcastro@ufmg.br,
⁴ andreipaim@ufmg.br,³ apbraga@ufmg.br⁵

Resumo Este trabalho apresenta uma proposta de método *map-matching online* para pré-processamento de trajetórias de veículos. O método utiliza um modelo logístico para determinar a probabilidade de localização de uma trajetória pertencer a um segmento que representa uma rua ou estrada. A avaliação do modelo foi feita em duas etapas. A primeira avalia a curva ROC, AUC, acurácia, sensibilidade e especificidade do modelo em diferentes bases de dados. Na segunda etapa foi feita uma comparação com métodos propostos na literatura. A performance dos métodos foi avaliada com relação à média da distância de erro, à mediana da distância do erro e ao *Dynamic Time Warping*. Os resultados para diferentes bases de dados demonstram que a abordagem é promissora ainda que a medição do GPS seja realizada em situações de diferentes valores de precisão.

1 Introdução

A mineração de dados de trajetórias de veículos é uma área em crescente desenvolvimento. Os avanços tecnológicos em telemetria facilitam a obtenção da localização de veículos e favorecem o surgimento de aplicações que utilizam estes dados para oferecer serviços, em tempo real, aos seus usuários. A Figura 1a apresenta uma arquitetura geral de um serviço baseado em localização. Inicialmente, tais sistemas recebem como entrada dados referentes à posição de veículos através do GPS (*Global Positioning System*). Estes dados são armazenados e processados em um servidor de aplicação com o objetivo de se extrair conhecimento (padrões, anomalias ou tendências) e finalmente, o sistema fornece como saída algum tipo de serviço para os usuários [13].

Embora os avanços em sensores sejam um grande motivador da implementação deste tipo de sistema, ainda existe um grande desafio com relação à qualidade dos dados de entrada [13]. O sinal de GPS torna-se pouco confiável quando está operando perto de obstáculos, devido às reflexões multi-caminho. Outros problemas consistem na exposição à interferências, especialmente quando o GPS estiver funcionando em frequências civis [2]. Portanto, faz-se necessária uma etapa de pré-processamento dos dados, localizada entre a coleta e aplicação. Neste cenário, uma importante técnica de pré-processamento é o *map-matching* [6].

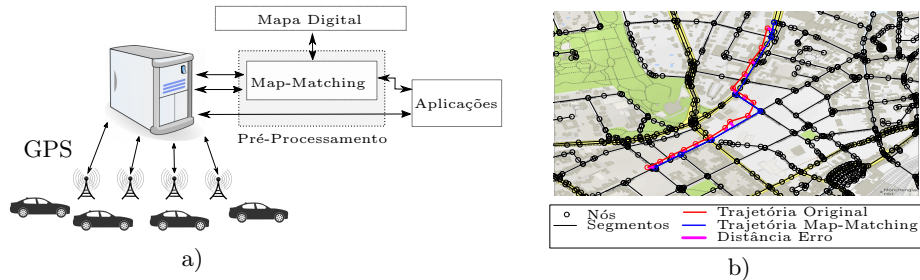


Figura 1: a) Arquitetura geral de um sistema baseado em localização. b) Mapa digital de uma região.

Os algoritmos *map-matching* utilizam como entrada, dados de um sistema de posicionamento, como GPS, e dados de um mapa digital de ruas. O mapa digital representa a rede rodoviária de uma região na forma de um grafo, no qual os segmentos representam ruas e os nós interseções entre ruas, conforme ilustrado na Figura 1b. Considerando a imprecisão inerente à localização fornecida pelo GPS (linha vermelha da Figura 1b), o propósito geral de um algoritmo de *map-matching* é identificar o segmento correto (linha azul da Figura 1b) no qual o veículo está viajando e determinar a sua localização neste segmento. Este procedimento não só permite identificar a localização física do veículo, mas também melhora a precisão do posicionamento. Os algoritmos *map-matching* podem ser classificados em métodos *online* e *offline* [10]. No primeiro caso, os dados de GPS são recebidos na forma de fluxos de dados contínuos, o que torna necessário o processamento *online* e em muitos casos em tempo real. Os métodos *offline*, por sua vez, processam a trajetória completa, i.e., executam o casamento dos pontos após toda a trajetória ter sido coletada e estar disponível e por isso, são mais propensos a encontrar uma solução de ótimo global [10].

Os estudos apresentados por [4,7,10] apresentam uma revisão da literatura sobre métodos de *map-matching online*. Os trabalhos apontam que os principais métodos empregam técnicas de Inteligência Computacional e apresentam, em geral, acurácia satisfatória. Porém ressaltam que as diferenças entre os cenários de teste utilizados e o uso de bases de dados pouco representativas impedem uma medida de performance conclusiva entre os algoritmos da literatura.

No estudo realizado em [7] foi utilizada uma metodologia de comparação onde os métodos foram submetidos a condições homogêneas de teste e comparados de acordo com diferentes métricas. Os resultados mostraram que o modelo linear proposto por [11] foi um dos algoritmos que apresentaram melhores resultados, no entanto, o trabalho não apresentou uma metodologia de estimação dos parâmetros para o modelo, estes foram obtidos "testando diferentes alternativas", como sugerido em [11].

A principal motivação deste trabalho é propor um algoritmo no qual seus parâmetros são calculados utilizando técnicas de regressão, diferentemente da abordagem empírica sugerida em [11]. Para tanto foi adotado um modelo logístico

de classificação, dado que o problema de *map matching* pode ser modelado como a classificação de segmentos, situados dentro de uma região de interesse, em duas classes, os segmentos falsos e o segmento verdadeiro, resultando em um problema de classificação binária. Dessa forma, um novo método para *map-matching online* é proposto. O método consiste em um modelo logístico que utiliza como entrada, características extraídas da distância e da diferença dos ângulos *heading* entre os pontos de GPS e os segmentos candidatos. O modelo logístico fornece como saída um valor de probabilidade de cada segmento candidato ser o segmento correto, onde se encontra o veículo. Foram realizados testes com dados reais em diferentes cenários. Os resultados mostraram que a abordagem proposta foi superior aos algoritmos da literatura, sendo promissora para aplicações reais.

O artigo está organizado da seguinte forma: na segunda seção é detalhada a metodologia utilizada para a implementação do método proposto. Na terceira seção descreve-se os experimentos realizados para avaliar a proposta. Na quarta seção os resultados obtidos são apresentados e discutidos. Por último são apresentadas as conclusões e trabalhos futuros.

2 Metodologia

O objetivo do algoritmo *map-matching* é encontrar o segmento ou estrada ao qual pertencem os pontos capturados pelo GPS em uma trajetória realizada por um veículo. A proposta deste artigo utiliza um modelo não linear para encontrar a probabilidade de cada segmento do mapa digital ser a verdadeira posição do veículo. A busca é realizada dentro de uma região de interesse com centro no ponto do GPS com raio de 10 metros, este parâmetro foi escolhido com base aos resultados obtidos em experimentos realizados e baseado em trabalho da literatura [5]. Desta forma, a projeção do ponto de GPS no segmento com maior probabilidade dentro da região de interesse é indicada como a posição correta do veículo.

Esta seção está dividida em 3 subseções. Inicialmente são descritas cada uma das variáveis utilizadas no modelo, em seguida é detalhado o modelo implementado, e finalmente é explicado o método utilizado para estimar os parâmetros do modelo.

2.1 Variáveis utilizadas no modelo

Duas variáveis muito utilizados nos algoritmos de *map-matching* são a distância e a direção [11,5,9]. A distância é a projeção do ponto do GPS até uma linha ou segmento, e a direção é a medida *heading* do ponto de GPS [11].

A Figura 2 ilustra a distância (d) entre um ponto de posicionamento dado pelo GPS (P) e um segmento de linha que representa uma rua (ℓ). Dois casos podem existir ao se calcular a distância. O primeiro caso é quando o ponto (C) da projeção de P no segmento (ℓ) está incluso no intervalo dos extremos do segmento $[A, B]$, neste caso a distância é definida pela projeção perpendicular do ponto ao segmento, representado por $d(P_2, C_1)$ e $d(P_3, C_2)$ na Figura 2. Caso contrário, a distância é definida pela distância mínima entre $d(P, A)$ e $d(P, B)$,

representado por $d(P_1, A)$, $d(P_4, B)$ e $d(P_5, B)$ na Figura 2. A Equação 1 descreve o cálculo da distância.

$$d(P, \ell) = \begin{cases} d(P, C) & \text{Se } C \in [A, B] \\ \text{Min}(d(A, P), d(B, P)) & \text{caso contrário} \end{cases}, \quad (1)$$

A direção (O) é a diferença (Δ) entre o ângulo *heading* do ponto de localização dado pelo GPS e a direção do segmento de linha ℓ . O Δ pode ter valores entre 180° e -180° . A direção é baseada na resolução dos ângulos dos segmentos do mapa de rotas, e dado que a maioria das interseções são maiores a 45° [11], este parâmetro foi utilizado como tamanho do intervalo de discretização. Após isso, o valor do Δ é discretizado como pode-se observar na Figura 3. A discretização ocorre da seguinte forma. Se o Δ está entre o intervalo de -25.5° a 25.5° seu valor será de 0, se o Δ está entre o intervalo de 67.5° a 125.5° ou entre -125.5° e -67.5° seu valor será de 2, assim sucessivamente.

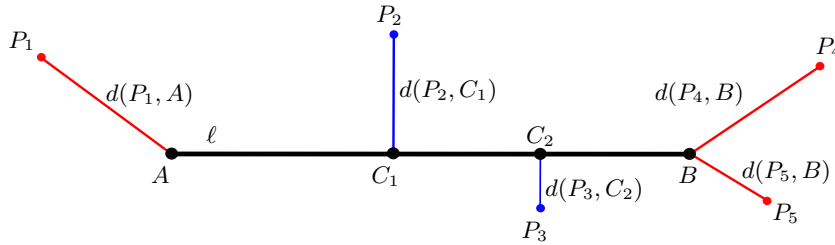


Figura 2: Distância entre os pontos P_1, P_2, P_3, P_4 e P_5 com o segmento ℓ .

2.2 Modelo

A distância e a direção são dois fatores medidos para calcular o *matching* entre o ponto de localização e o segmento de linha. Neste artigo foi proposto um modelo logístico para determinar a probabilidade ($M_{i,j}$) do ponto P_i pertencer ao segmento de linha ℓ_j , i são os pontos de localização que compõem a trajetória e j são os segmentos candidatos dentro da região de interesse do ponto i . No modelo, a distância é uma variável numérica, contínua e positiva, representada por $D_{i,j}$ (Equação 1). É utilizada a direção no ponto atual e em um ponto futuro, o intervalo de tempo entre os pontos depende da frequência de amostragem do GPS. Para a direção são utilizadas variáveis *dummy* para os valores de Δ discretizado (Figura 3), representada pela $O_{i,j,k,h}$. Onde $k = 2$ para o ponto atual e $k = 3$ para o tempo futuro, e h são os valores de Δ (0, 1, 2, 3 e 4).

O modelo implementado foi:

$$M_{i,j} = \frac{e^{\theta_{i,j}}}{1 + e^{\theta_{i,j}}} \quad (2)$$

$$\theta_{i,j} = B_0 + B_1 * D_{i,j} + \sum_{k=2}^3 \sum_{h=0}^4 B_{k,h} O_{i,j,k,h} \quad (3)$$

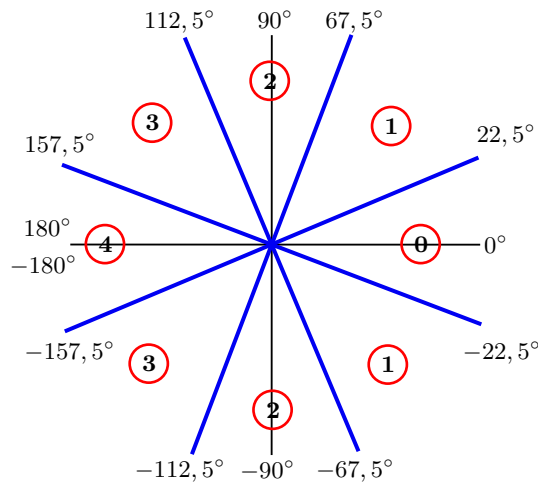


Figura 3: Ângulo Δ discretizado. $0 = [-22.5, 22.5]$, $1 = \{[-67.5, -22.5], [22.5, 67.5]\}$, $2 = \{[-112.5, -67.5], [67.5, 112.5]\}$, $3 = \{[-157.5, -112.5], [112.5, 157.5]\}$, $4 = \{157.5, -157.5\}$.

Uma região de interesse pode obter mais de um segmento candidato com probabilidade alta, isto pode ser, por exemplo, devido à ruas paralelas ou que estejam muito próximas. Desta forma, depois de ter calculada as probabilidades $M_{i,j}$ para cada segmento dentro de uma região de interesse, o segmento escolhido é aquele que apresenta maior probabilidade. Portanto, se garante que dentro de uma região de interesse somente um segmento seja classificado como verdadeiro. Caso, dentro da região de interesse não exista nenhum segmento, ponto do GPS será a posição correta da localização, indicada como fora de rota (*off-road*).

2.3 Estimação dos parâmetros do modelo

Para estimar os parâmetros do modelo (Equação 2) foi usada a base de dados "Totais" disponível no repositório¹ [7]. A escolha desta base de dados foi devido à sua disponibilidade e ao fato de que possui todas as variáveis requeridas. Os dados desta base foram obtidos de diferentes usuários e em ambientes não controlados, o que cria um cenário real e com uma grande diversidade de trajetórias. O mapa digital foi obtido pelo OpenStreetMap². As ruas ou segmentos dentro da região de interesse de cada ponto de GPS em cada trajetória foram rotuladas como 1 se o segmento é a localização verdadeira e 0 caso contrário. Na rotulação foram utilizadas como trajetórias de referência os resultados gerados pela API Map Matching Graphhopper³. Esta API implementa um algoritmo de

¹ <https://github.com/juankmilofg/Map-Maching>

² <http://www.openstreetmap.org>

³ <https://graphhopper.com/api/1/docs/map-matching>

map-matching offline [8] o qual apresenta resultados muito próximos do ótimo global. A base de dados "Totais" tem um conjunto de 96 trajetórias, das quais foram selecionados 48750 *matching* de pontos de GPS com segmentos de linha, sendo que 22871 foram rotuladas com 1 e 25879 com 0. Finalmente, a estimação dos parâmetros do modelo foi realizada com o método de ajuste de mínimos quadrados ponderados utilizando o *software R Project*. Os valores dos parâmetros obtidos foram:

$$\begin{aligned} B_0 &= 5.072594 & B_1 &= -0.892882 \\ B_{2,1} &= -1.079052 & B_{2,2} &= -3.203425 & B_{2,3} &= -1.791771 & B_{2,4} &= -0.621893 \\ B_{3,1} &= -0.614030 & B_{3,2} &= -0.688642 & B_{3,3} &= -0.956318 & B_{3,4} &= -0.378171 \end{aligned} \quad (4)$$

Os valores de $B_{2,0}$ e $B_{3,0}$ estão inclusos em B_0 .

3 Experimentos

Com a finalidade de avaliar o desempenho do algoritmo proposto, primeiramente é analisada a resposta do modelo logístico. Para isso foi utilizada a curva ROC (*Receiver Operating Characteristic*), que representa o compromisso entre a sensibilidade e a especificidade considerando todos os possíveis limiares de classificação (τ), e foi calculada a AUC (*Area Under ROC Curve*) na base de dados de treinamento. Além disso foi calculada a acurácia, sensibilidade e especificidade em diferentes bases de dados. No modelo logístico ajustado, define-se como Positivo o segmento que contém o ponto real da trajetória e Negativo, caso contrário. Deste modo, define-se como Verdadeiro Positivo (VP) o segmento classificado como Positivo pelo modelo logístico quando é positivo realmente, Falso Positivo (FP) o segmento classificado como Positivo pelo modelo logístico quando é Negativo, Verdadeiro Negativo (VN) o segmento classificado como Negativo pelo modelo logístico quando é Negativo de efetivamente e Falso Negativo (FN) o segmento classificado como Negativo pelo modelo logístico quando é Positivo.

A acurácia do modelo é a porcentagem de acertos Positivos e Negativos do modelo de classificação, isto é representado pela Equação 5. A sensibilidade reflete o quanto o modelo é eficaz para classificar segmentos Positivos corretamente, Equação 6, e a sensibilidade mede o quanto o método é capaz de identificar segmentos Negativos corretamente, Equação 7.

$$\text{Acurácia} = \frac{(\text{VP})+(\text{VN})}{(\text{VP})+(\text{FP})+(\text{VN})+(\text{FN})} \quad (5)$$

$$\text{Sensibilidade} = \frac{(\text{VP})}{(\text{VP})+(\text{FN})} \quad (6)$$

$$\text{Especificidade} = \frac{(\text{VN})}{(\text{FP})+(\text{VN})} \quad (7)$$

As bases de dados utilizadas foram "Totais", "Grupo₁", "Grupo₂" e "Grupo₃", disponíveis no repositório⁴ [7]. As bases de dados foram divididas com base no

⁴ <https://github.com/juankmilofg/Map-Matching>

índice de precisão PDOP⁵ (*Position Dilution of Precision*). O "Grupo₁" é formado por 38 trajetórias com boa precisão (PDOP < 2). O "Grupo₂" corresponde a 49 trajetórias com precisão média (2 ≤ PDOP < 3). O "Grupo₃" é formado por 38 trajetórias com baixa precisão (3 ≤ PDOP) ⁶.

Em seguida, foi realizada uma comparação com algoritmos *map – matching online* propostos na literatura, Quddus et al. (2006) [9], Jagadeesh et al. (2004) [5] e as variações linear e não linear propostas por Ren and Karimi (2009) [11]. Estes serão chamados de Algoritmo Quddus, Jagadeesh, Ren. Linear e Ren. N.Linear respectivamente.

O método Quddus possui três etapas. Na primeira (*map-matching* inicial), o algoritmo seleciona o primeiro segmento correto do veículo baseado na leitura dos primeiros pontos de GPS da trajetória. Na segunda etapa (*map-matching* no mesmo segmento), o algoritmo verifica se o veículo está no mesmo segmento que o ponto de GPS anterior. Na terceira (*map-matching* em uma junção), o algoritmo verifica e identifica o segmento correto quando o veículo se encontra próximo a uma interseção de segmentos. Em cada uma das três etapas, o algoritmo utiliza um conjunto de regras *fuzzy* específicas para ponderar os segmentos candidatos e selecionar aquele de maior peso.

O método Jagadeesh seleciona, para cada novo ponto de GPS recebido, um conjunto de segmentos candidatos, que se encontram dentro de uma região de interesse. Em seguida, um conjunto de regras *fuzzy* é utilizado para selecionar o segmento correto. Neste conjunto de regras, o algoritmo verifica também, a probabilidade do segmento selecionado ser o correto, com base em informações dos segmentos selecionados anteriormente pelo algoritmo.

O método Ren seleciona os segmentos candidatos com base na região de interesse calculada para cada novo ponto de GPS. Em seguida, com base na discretização da medida do *heading* do ponto atual e dos seguintes dois pontos, e a distância com respeito aos segmentos candidatos, faz-se a ponderação dos segmentos candidatos e seleciona-se aquele com maior peso. Para tanto, os autores propõem duas formas de ponderação, Ren. Linear e Ren. N.Linear. Ren. Linear consiste em um modelo linear, que considera uma soma ponderada das diferenças entre os ângulos do veículo e dos segmentos candidatos. Ren. N.Linear utiliza uma rede RBF para ponderar os segmentos candidatos. Ao final, tanto para o modelo linear como para o não linear, o segmento com maior peso é selecionado.

O algoritmo proposto foi implementado utilizando o *software R Project*. Os algoritmos Quddus, Jagadeesh, Ren Linear e Ren N.Linear utilizados foram implementados em [7] e estão disponíveis no repositório⁴. Todos os algoritmos foram implementados utilizando os parâmetros recomendados pelos autores.

Para a análise comparativa dos algoritmos, foram utilizadas três métricas trajetória-trajetória. A primeira métrica é a distância média, usada para avaliar o erro médio dos algoritmos ao longo das trajetórias. A distância média entre os pontos da trajetória calculada pelo algoritmo *map-matching* ($ta = [q_{a,1}, \dots, q_{a,n}]$)

⁵ PDOP representa a precisão instantânea da posição fornecida pelo GPS. Menor valor indica maior precisão.

⁶ http://wiki.openstreetmap.org/wiki/Accuracy_of_GPS_data

de tamanho n e a trajetória de referência ($tr = [q_{r,1}, \dots, q_{r,m}]$) de tamanho m , é definida pela Equação 8, onde $d(p, q)$ é a distância superficial entre dois pontos (Equação 9).

$$d_{média}(ta, tr) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m d(q_{a,i}, q_{r,j}) \quad (8)$$

$$d(p_1, p_2) = \cos^{-1}(\sin(lat_1) \times \sin(lat_2) + \cos(lat_1) \times \cos(lat_2) \times \cos(lon_1 - lon_2)) \quad (9)$$

A segunda métrica é a mediana da distância entre trajetórias calculadas e trajetórias de referência, com o objetivo de avaliar o desempenho médio dos algoritmos desconsiderando possíveis *outliers*. Para definir a distância mediana, é calculado um vetor de distâncias mínimas D_{min} entre os pontos da trajetória calculada pelo algoritmo e a trajetória de referência (Equação 10). Desta forma, a distância mediana é o valor que encontra-se no centro do vetor $D_{min}(ta, tr)$.

$$D_{min}(ta, tr) = [d_{min}(q_{a,1}, tr), \dots, d_{min}(q_{a,n}, tr)] \quad (10)$$

Onde a distância mínima (em metros) entre o ponto p_i capturado pelo GPS, e a trajetória de referência tr_j , definida na Equação 11.

$$d_{min}(p_x, tr_j) = \min\{d(p_x, p_i), \dots, d(p_x, p_m)\} \quad (11)$$

A terceira métrica é chamada *Dynamic Time Warping* (DTW), utilizada para medir a similaridade entre trajetórias espaciais [14]. A DTW é geralmente mais adequada do que a distância média quando se deseja avaliar a similaridade baseada na forma global da trajetória, pois é menos suscetível a deslocamentos locais, provocados por *outliers*.

Para verificar significância estatística entre os algoritmos foi utilizado o teste de Friedman [3]. Este teste pertence à categoria de métodos não-paramétricos para múltipla comparação de amostras pareadas. Caso o teste de Friedman retorne evidências de que existe diferença significativa entre os métodos, é feita uma análise *post-hoc* todos contra todos. A análise *post-hoc* utiliza o teste estatístico Wilcoxon [12]. Todos os testes consideram um grau de confiança de 95%.

4 Resultados

A curva ROC e a variação da sensibilidade e a especificidade no modelo sobre a base de dados são apresentadas na Figura 4, onde verifica-se que modelo obteve um valor AUC do 93,08%. Os resultados da acurácia, sensibilidade e especificidade do modelo logístico são apresentados na Tabela 1. O modelo foi testado em três bases de dados de teste, *Grupo₁*, *Grupo₂* e *Grupo₃*, e os resultados para base de dados *Totais* usada no treinamento também são apresentados. Os resultados mostraram que modelo conseguiu realizar uma boa classificação dos segmentos. A curva ROC apresentada na Figura 4 sugere que diferentes limiares podem ser

escolhidos para cada aplicação dependendo de requerimentos específicos. Na implementação do modelo não foi escolhido um limiar de classificação pois dentro de uma região de interesse somente um segmento pode ser considerado Positivo sendo assim, apenas o segmento com maior probabilidade foi escolhido. A tabela 1 mostra que, selecionando o segmento com maior probabilidade, se pode obter um bom compromisso entre a sensibilidade e a especificidade, uma vez que os valores ficaram muito próximos ao valor de corte da Figura 4b. Os resultados demonstram que apesar do desbalanceamento presente nas bases de dados, isto é, o fato de que dentre todos os segmentos presentes em uma região de interesse somente um deles é verdadeiro, o desempenho do algoritmo foi robusto nos testes realizados.

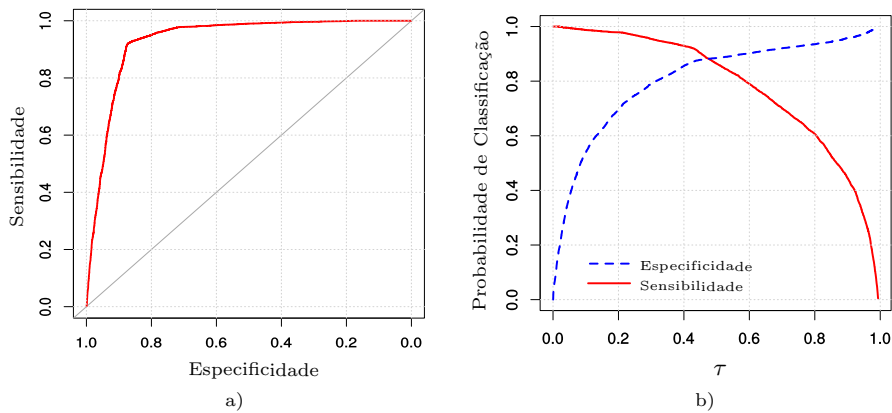


Figura 4: a) Curva ROC. b) Sensibilidade e especificidade para diferentes valores de τ .

Tabela 1: Análise do modelo logístico

	Acurácia	Sensibilidade	Especificidade
<i>Totais</i>	0.869575	0.928157	0.826631
<i>Grupo₁</i>	0.908889	0.948322	0.871277
<i>Grupo₂</i>	0.859952	0.872513	0.851738
<i>Grupo₃</i>	0.817362	0.836726	0.807240

Os resultados dos testes para as três métricas utilizadas na avaliação do algoritmo proposto, junto com os algoritmos da literatura são mostrados na Figura 5. A figura representa o alinhamento do gráfico de dispersão [1] do desempenho dos algoritmos em cada métrica. Foram realizados os testes com relação ao erro da distância média, o erro da distância mediana e ao DTW de cada algoritmo para as trajetórias do *Grupo₁*, *Grupo₂*, *Grupo₃* e *Totais*. Os resultados dos testes estatísticos sugerem que existe uma diferença significativa entre os métodos em todos os grupos. Além disso, os resultados mostraram evidências de que o método proposto foi significativamente superior aos demais algoritmos em todos os grupos e todas as métricas, exceto no *grupo₃* no qual o Ren. Linear e

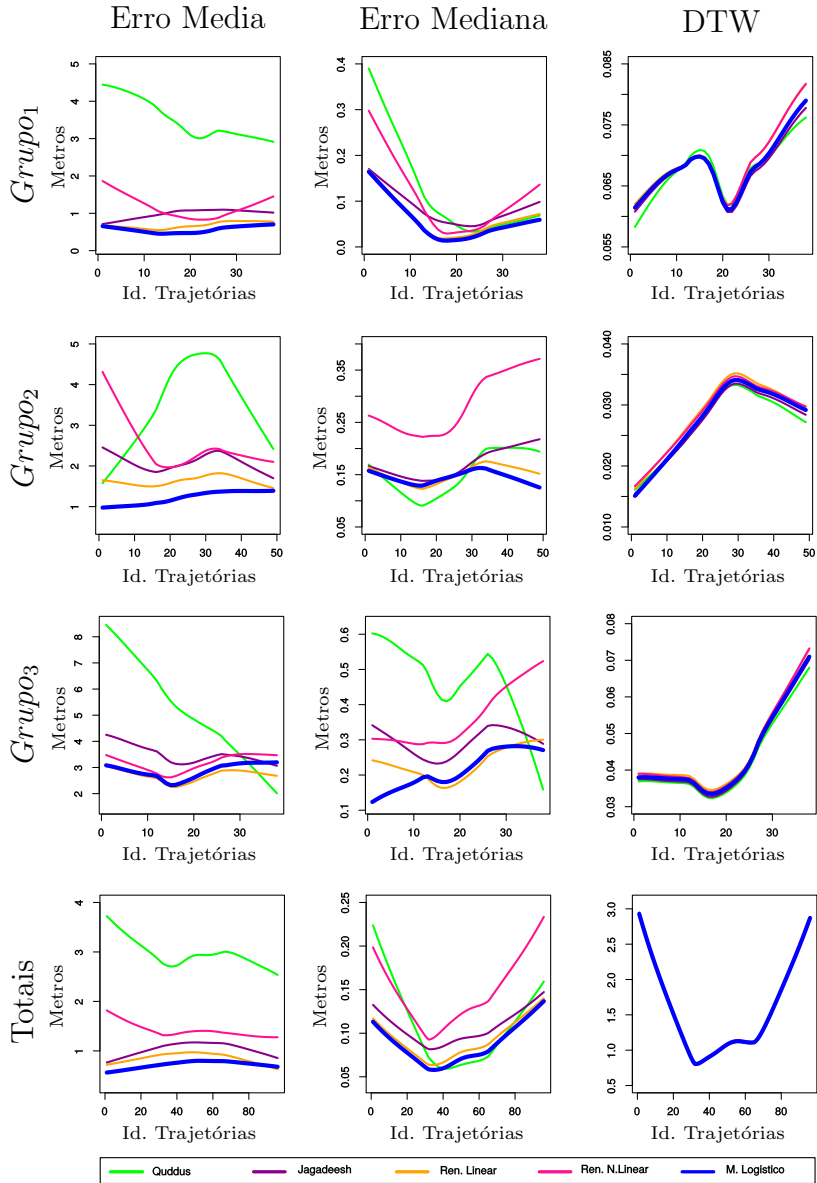


Figura 5: Resultados da implementação das métricas de avaliação: distância média, distância mediana e DTW, nas bases de dados do *Grupo1*, *Grupo2*, *Grupo3* e *Totais*.

o algoritmo proposto não obtiveram uma diferença significativa na métrica do erro médio e mediana. Entretanto, pode ser verificado na Figura 5 que os algoritmos obtiveram resultados muito próximos neste cenário. Observou-se que, de modo geral, o algoritmo Quddus não conseguiu desempenho estável durante os experimentos. Recomenda-se assim, fazer uma comparação mais justa com este algoritmo, utilizar trajetórias maiores e realizar algum procedimento de ajuste de seus parâmetros com base em dados.

Os resultados sugerem que o método proposto apresentou resultados mais robustos e estáveis ao longo das trajetórias nas diferentes bases de dados, sendo significativamente melhor que os demais algoritmos. Sendo assim, recomenda-se seu uso em aplicações reais, onde não se tem informação sobre a qualidade do sinal de GPS. Dessa forma, o algoritmo poderia ser utilizado, em aplicações como navegação em tempo real, em que pontos com grande erro são indesejáveis. Ademais, o desempenho apresentado na métrica DTW, sugere efetividade para aplicações como mineração de dados, onde a forma global da trajetória possui maior impacto. Verificou-se também que os resultados obtidos pelo método proposto em trajetórias com PDOP menores a três (*Grupo₁* e *Grupo₂*) e em trajetórias reais (*Totais*) mostraram um erro da distância média menor a um metro. Já no *Grupo₃* (PDOP maior que três) o erro médio ficou abaixo de quatro metros, o que demonstra o bom desempenho do algoritmo proposto.

5 Conclusões

Este trabalho apresentou um algoritmo *map-matching online* para pré-processamento de trajetórias de veículos baseado em um modelo logístico. O modelo logístico considerou como entrada variáveis relacionadas à distância e à direção de cada ponto de GPS para os segmentos candidatos, gerando na saída a probabilidade de cada ponto pertencer a cada um dos segmentos. Os experimentos foram realizados com um conjunto abrangente de trajetórias e de modo geral, os resultados mostraram que para todas as métricas de desempenho avaliadas, o método proposto apresenta desempenho satisfatório sendo superior aos demais algoritmos da literatura em grande parte dos casos e equivalente nos outros. Desta maneira, é possível afirmar que a metodologia proposta é promissora para aplicação em problemas reais de *map matching online*. Os resultados da avaliação do algoritmo no problema de classificação de segmentos candidatos, evidenciaram que a natureza desbalanceada dos dados, inerente a esta aplicação, degradou um pouco o desempenho do algoritmo em algumas situações, porém ainda assim, o modelo mostrou-se robusto com valores satisfatórios de especificidade e sensibilidade. Assim, com o objetivo de melhorar o desempenho do método proposto, sugere-se como trabalho futuro, um estudo relacionado à análise e seleção de outras características como entradas para o modelo logístico, e o uso de técnicas de regressão específicas para bases de dados desbalanceadas. Além disso, sugere-se fazer um estudo comparativo do custo computacional do modelo proposto com técnicas já implementadas na literatura.

Agradecimentos

O presente trabalho foi realizado com o apoio financeiro da FAPEMIG, CNPq e CAPES - Brasil.

Referências

1. Cleveland, W.S.: Lowess: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician* 35(1), 54–54 (1981)
2. Fonseca-Galindo, J.C., Lemos, A.P.: Proposta de um método de extração de características aplicado ao problema de estimação da posição de um vant em navegação autônoma. *Anais do XXI Congresso Brasileiro de Automática* pp. 3139–3144 (2016)
3. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics* 11(1), 86–92 (1940)
4. Hashemi, M., Karimi, H.A.: A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems* 48, 153 – 165 (2014)
5. Jagadeesh, G.R., Srikanthan, T., Zhang, X.D.: A map matching method for gps based real-time vehicle location. *Journal of Navigation* 57(3), 429–440 (2004)
6. Krumm, J.: *Trajectory Analysis for Driving*, pp. 213–241. Springer New York, New York, NY (2011)
7. Maia Neto, J., Fonseca-Galindo, J.C., Castro, C.L., Lemos, A.P.: Algoritmos de map-matching online para processamento de trajetórias de veículos: um estudo comparativo. *Simpósio Brasileiro de Automação Inteligente (Aceito para publicação)* (2017)
8. Newson, P., Krumm, J.: Hidden markov map matching through noise and sparseness. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. pp. 336–343. GIS '09, ACM, New York, USA (2009)
9. Quddus, M.A., Noland, R.B., Ochieng, W.Y.: A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems* 10(3), 103–115 (2006)
10. Quddus, M.A., Ochieng, W.Y., Noland, R.B.: Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies* 15(5), 312 – 328 (2007)
11. Ren, M., Karimi, H.A.: A chain-code-based map matching algorithm for wheelchair navigation. *T. GIS* 13(2), 197–214 (2009)
12. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics bulletin* 1(6), 80–83 (1945)
13. Zheng, Y.: *Trajectory data mining: An overview*. *ACM Transaction on Intelligent Systems and Technology* (September 2015)
14. Zheng, Y., Zhou, X.: *Computing with spatial trajectories*. Springer Science & Business Media (2011)