Pontifícia Universidade Católica de Minas Gerais
Instituto de Ciêcias Exatas e Informática

Estudo da Mobilidade Social em Redes Veiculares*

Túlio Nunes Polido Lopes[1]
Felipe Domingos da Cunha[2]

*Artigo apresentado ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais como pré-requisito para obtenção do título de Bacharel em Ciência da Computação.

[1]Aluno do Programa de Graduação em Ciência da Computação, Brasil– tulio.polido@sga.pucminas.br.
[2]Professor do Programa de Graduação em Ciência da Computação, Brasil– felipe@pucminas.br.

1 INTRODUÇÃO

Desde a popularização de veículos automotores no século XX, diversas cidades, em especial as grandes metrópoles, vêm sofrendo com a crescente demanda por controle de trânsito. No Brasil, a cidade de São Paulo é o ápice desse problema, e, mesmo utilizando rodízios frequentes de veículos, não há mudança significativa. Um possível passo para buscarmos uma solução prática, viável e com bons resultados é analisar como se comporta o trânsito das cidades ao redor do mundo, buscando padrões de movimentação entre os veículos. O Seguinte trabalho busca fazer uma análise inicial da forma como o trânsito se comporta na cidade de Roma, utilizando dados de GPS de táxis, e estabelecer um algoritmo para separar os dados brutos de cada veículo em viagens individuais. Na seção 2, os trabalhos relacionados são descritos. Em seguida, na seção 3, é feita uma análise utilizando o trace da cidade de Roma. Na seção 4, são tiradas conclusões com base nos dados obtidos e, por fim, na seção 5 possíveis trabalhos futuros são descritos.

2 TRABALHOS RELACIONADOS

Em (BASTA et al., 2016) vemos a proposta de um modelo de mobilidade social para ser utilizados em simulações de redes veiculares. (CELES et al., 2019) estuda a mobilidade de ônibus intramunicipais e seu impacto na construção de redes veiculares em uma cidade. Em (GAINARU et al., 2009) também temos a proposta de um modelo de mobilidade social, porém nele temos a preocupação com um simulação microscópica, que permite testar diferentes personalidades para os motoristas. (GONG et al., 2015) estabelece parâmetros e métodos para seccionar dados de GPS de veículos que não estejam divididos em viagens. (KONG et al., 2018) propõe um método de simulação de dados de veículos particulares a fim de facilitar a obtenção de *datasets* para estudos futuros. Por fim, (NING et al., 2017) dá uma breve introdução a o que são as *Vehicular Social Networks*(VSN) comparando com outros tipos de redes já conhecidas.

3 ANÁLISE DO TRACE DE ROMA

3.1 Disposição dos dados

O trace de Roma está contido em um arquivo *.csv*. Neste arquivo, cada linha representa um ponto de GPS gravado, possuindo, necessariamente nesta ordem: Um inteiro para representar o ID do veículo; Um String contendo a data e hora em que o ponto foi gravado; Um float para a longitude; Um float para a latitude; Um booleano que define se o ponto sofreu calibração ou não.

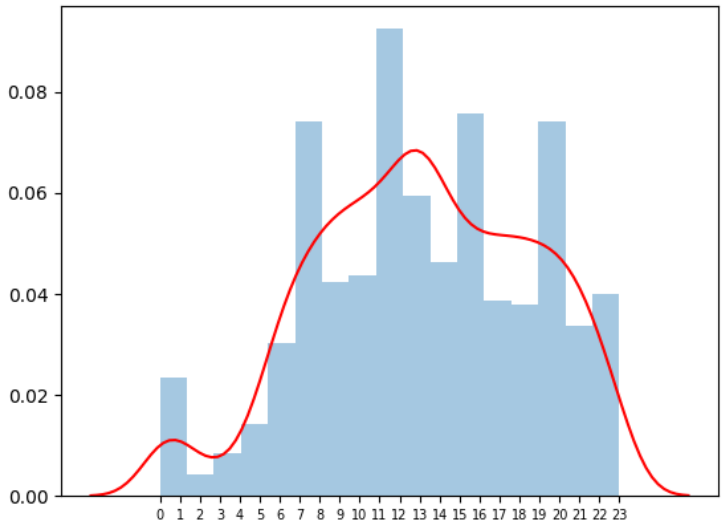
Quadro 1 - Exemplo dos Dados de GPS de Roma

ID	time	long-x	lat-y	is-calibrated
101	2014-02-04 05:00:01	12.48884	41.90304	1
101	2014-02-04 05:00:03	12.48875	41.90301	0

3.2 Fluxo de veículos

É de conhecimento geral que durante um dia o fluxo de veículos é heterogêneo. O Gráfico 1 demonstra claramente uma concentração bem maior de veículos durante alguns horários específicos do dia. Entre os horários com maior densidade de veículos, destacam-se três picos: O de 07:00 às 08:00, o de 11:00 às 12:00 e o de 19:00 às 20:00.

Gráfico 1 - Gráfico da Densidade de Veículos X Hora do Dia

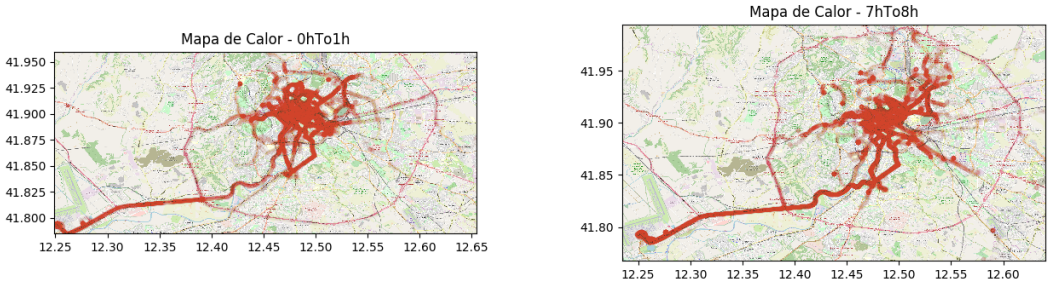


Fonte: Do Autor

Esses picos coincidem com os horários de uma jornada de trabalho padrão, onde o primeiro pico se daria no momento em que as pessoas da cidade se deslocam ao trabalho ou escola, o segundo ocorreria no momento de intervalo, onde há também uma tendência maior de jovens saindo e chegando nas escolas, e o terceiro seria o movimento de retorno.

Isso é reforçado pelos mapas de calor da cidade, onde pode ser visualizado a diferença de movimento de veículos na cidade.

Mapa de Calor 1:



Fonte: Do Autor

No Mapa de Calor - 7hTo8h, comparado ao Mapa de Calor - 0hTo1h, é visível o maior fluxo de carros em direção ao aeroporto da cidade, localizado na região sudoeste do mapa. Há também um fluxo maior nas rodovias que contornam Roma, indicando ser um horário com maior entrada e saída de veículos da cidade. A região central se mantém concentrada nos dois mapas, porém entre 00:00 e 01:00 as vias ao redor da região central são menos utilizadas.

Os gráficos mostrados evidenciam o que já é conhecido no ramo da Geografia como Migração Pendular, isto é, um movimento diário de pessoas que se deslocam a regiões da cidade onde há maior concentração de comércio, indústrias e instituições de ensino durante o dia e, durante o início do período noturno, há o retorno dessas pessoas para suas residências, em geral mais afastadas dos centros das cidades. A causa desse fluxo é a horizontalização das cidades, em que a população com seu próprio crescimento tende a estabelecer novas moradias ao redor da cidade, cada vez mais afastadas do centro. Levando a um aumento cada vez maior no fluxo de veículos durante os horários de pico.

3.3 Reorganização da base

3.3.1 Necessidade da reorganização

Para uma análise mais bem fundamentada das informações, o ideal é a criação de uma matriz de origem/destino(OD), onde as linhas representarão as posições iniciais das viagens, enquanto as colunas, as posições finais. Como explicado na seção 3.1, os dados das viagens não possuem um parâmetro que indica o início e o fim de uma corrida. Assim, se faz necessário a utilização de metodologias, como as descritas em (GONG et al., 2015), para a separação desses dados brutos em cada corrida dos respectivos táxis.

3.3.2 Algoritmos para reorganização

Nesse trabalho a linguagem utilizada no processamento dos dados foi Python devido a suas bibliotecas científicas otimizadas para processamento de grande volume de dados como Pandas e NumPy. Todos os dados estão ordenados por duas chaves, a primeira é o ID do veículo, já a segunda, o horário em que o ponto foi gravado. Os dados passam por um loop onde a distância e o tempo de gravação entre dois pontos consecutivos são gravados em duas listas. O algoritmo pode ser visto a seguir.

Algoritmo 1 - Geração das listas de intervalos

Algorithm 1:

Entrada: Pontos ordenados por horário, onde cada ponto possui uma coordenada e um horário de gravação.
Saída: Uma lista contendo a diferença de tempo entre cada ponto e outra lista contendo a diferença de distância.
for $key, value$ in $Entrada$ **do**
    timeGaps[key] = **nova lista**
    coordGaps[key] = **nova lista**
    for i=0 to $value.length$ -1 **do**
        point1 = $value[i]$
        point2 = $value[i+1]$
        pointCoord1 = point1.$coordinate$
        pointCoord2 = point2.$coordinate$
        distance = $haversine$(pointCoord1,pointCoord2)
        coordGaps[key].$append$(distance)
        pointTime1 = point1.$hour$
        pointTime2 = point2.$hour$
        timeGaps[key].$append$(pointTime2 - pointTime1)
    end for
end for
Retorna timeGaps,coordGaps; FIM

Fonte: Do autor.

O *for* externo cria uma nova lista para cada ID de veículos, contido no atributo *key*, enquanto o atributo *value* contém a lista de pontos específica daquele veículo. O ciclo interno percorre a lista de pontos do ID atual, onde grava a diferença entre os dois horários, em segundos, e a diferença entre as duas coordenadas, em metros, respectivamente nas listas *timeGaps* e *coordGaps*. O algoritmo de Haversine utilizado acima é um método desenvolvido para calcular a distância em metros entre duas coordenadas geográficas.

Em seguida o algoritmo para separação de viagens é chamado. Ele recebe as duas listas de intervalos e retorna uma nova lista contendo os índices dos pontos onde devem ocorrer o início de novas *trips*.

Algoritmo 2 - Criação das viagens

Algorithm 2:

Entrada: Uma lista contendo a diferença de tempo entre cada ponto, outra lista contendo a diferença de distância e um vetor dos IDs.
Saída: Uma lista contendo os índices onde devem ocorrer as separações das viagens.
for key in $Keys$ **do**
    flag = $FALSE$
    tempo = 0
    separator[key] = **nova lista**
    for i=0 to $timeGaps.length$ -1 **do**
        if coordGaps[key][i] / timeGaps[key][i] < 0.1 **then**
            tempo += timeGaps[key][i]
        else
            if flag == $TRUE$ **then**
                separator[key].$append(i)$
                flag = $FALSE$
                tempo = 0
            else
                tempo = 0
                flag = $FALSE$
            end if
        end if
        if tempo > 30.0 **then**
            flag = $TRUE$
        end if
    end for
end for
Retorna separator; FIM

Fonte: Do autor.

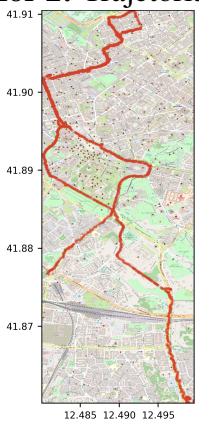
O ciclo externo percorre um vetor contendo as chaves, ou IDs, dos veículos. Para cada ID presente é criada uma lista onde serão gravadas os índices de fim de viagens. Também são criadas duas variáveis de controle, tempo, que grava quantos segundos o carro ficou parado, e *flag*, que define se o tempo parado é suficiente para definir uma nova viagem.

O ciclo interno caminha entre os dois vetores de entrada simultaneamente. A velocidade de cada posição é analisada, e, caso seja menor que 0.1 metros por segundo, é considerado que naquele momento o carro estava possivelmente parado, logo soma-se à variável tempo o valor em segundos daquele intervalo. Caso a velocidade seja maior que 0.1 metros por segundo, é verificado se a *flag* é verdadeira, caso seja, aquele ponto é considerado um ponto de ruptura e é salvo na saída, enquanto as variáveis de controle são zeradas. Caso a *flag* seja falsa as variáveis de controla são zeradas e a possível parada, desconsiderada. No fim do ciclo uma condição *if* confere se o tempo acumulado em velocidade baixa é maior que 30 segundos, e, caso verdadeiro, a *flag* assume valor *TRUE*, significando que assim que houver um sinal de aceleração do veículo um ponto de ruptura deve ser criado.

3.4 Resultados

Esse método de reorganização se mostrou pouco eficiente para a base utilizada. Numa análise feita acompanhando os pontos de ruptura selecionados durante a trajetória dos veículos, os resultados não foram tão satisfatórios. Muitos pontos parecem aleatórios seguindo uma lógica simples que uma viagem de táxi deve ir do ponto A ao ponto B da cidade na menor distância possível. Isso é evidente no Mapa de Calor 2.

Mapa de Calor 2: Trajetória ID 367 12h-13h



Fonte: Do Autor

É perceptível que essa trajetória selecionada pelo algoritmo não é viável, tendo em vista que o veículo dá uma volta claramente muito maior que necessária pra uma viagem. Assim, é provável que o algoritmo não tenha reconhecido outros pontos começo e fim de viagem nesse trecho.

4 CONCLUSÃO

Tendo como base a análise feita, é possível concluir que o algoritmo proposto não é eficiente para todas as viagens selecionadas. O provável erro é a falta de variáveis para definir o início e fim de viagens, além de não ser ideal estabelecer esses valores manual e aleatoriamente como foi proposto. Outras métricas deveriam ser adicionadas a fim de otimizar o código e eventualmente melhorar a capacidade do próprio de selecionar os pontos mais prováveis de ruptura de viagens.

5 TRABALHOS FUTUROS

Num próximo trabalho deve ser feito uma melhoria na técnica de separação das viagens na base de dados, podendo ser utilizados algoritmos de *Machine Learning* para atingir os melhores valores possíveis na geração das novas *trips*.

Outro possível passo seria o de enriquecer a base de dados de Roma tendo como base o trabalho feito em (KONG et al., 2018). Com esse enriquecimento seria possível comparar a base original com a recém gerada a fim de verificar a eficácia e precisão dos novos dados gerados.

REFERENCIAS

BASTA, Nardine et al. Generic geo-social mobility model for vanet. In: IEEE. **2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)**. [S.l.], 2016. p. 1–5.

CELES, Clayson; BOUKERCHE, Azzedine; LOUREIRO, Antonio AF. Towards understanding of bus mobility for intelligent vehicular networks using real-world data. In: IEEE. **2019 IEEE Global Communications Conference (GLOBECOM)**. [S.l.], 2019. p. 1–6.

GAINARU, Ana; DOBRE, Ciprian; CRISTEA, Valentin. A realistic mobility model based on social networks for the simulation of vanets. In: IEEE. **VTC Spring 2009-IEEE 69th Vehicular Technology Conference**. [S.l.], 2009. p. 1–5.

GONG, Lei et al. Identification of activity stop locations in gps trajectories by density-based clustering method combined with support vector machines. **Journal of Modern Transportation**, Springer, v. 23, n. 3, p. 202–213, 2015.

KONG, Xiangjie et al. Mobility dataset generation for vehicular social networks based on floating car data. **IEEE Transactions on Vehicular Technology**, IEEE, v. 67, n. 5, p. 3874–3886, 2018.

NING, Zhaolong et al. Vehicular social networks: Enabling smart mobility. **IEEE Communications Magazine**, IEEE, v. 55, n. 5, p. 16–55, 2017.