**Nanodegree Engenheiro de Machine Learning**

**Proposta de projeto final**

Fernando Roberto Schwartzer  
2 de novembro de 2018

**Proposta**

**Histórico do assunto**

Identificar os modos de transporte através de observações dos usuários, ou observação do ambiente, é um tópico crescente de pesquisa, com muitas aplicações no planejamento da mobilidade urbana. A detecção do modo de transporte fornece informações para o diagnóstico do uso da malha viária, da ocupação do solo, do deslocamento de cargas e, principalmente, dos deslocamentos das pessoas nas cidades.

O reconhecimento do modo de transporte do usuário pode ser considerado como uma tarefa de HAR (Human Activity Recognition). Seu objetivo é identificar que tipo de transporte - caminhar, dirigir etc. - uma pessoa está usando.

Historicamente, os dados dos sensores para reconhecimento de atividades eram difíceis e caros de coletar, exigindo hardware personalizado. Agora, telefones inteligentes e outros dispositivos de rastreamento pessoal usados ​​para monitoramento de saúde e fitness são baratos e onipresentes. Como tal, os dados de sensores destes dispositivos são mais baratos de coletar, mais comuns e, portanto, são uma versão mais comumente estudada do problema geral de reconhecimento de atividades.

**Descrição do problema**

O problema consiste na previsão da atividade dada uma captura instantânea de dados, geralmente de um ou de um pequeno número de tipos de sensores. Geralmente, esse problema é enquadrado como uma tarefa de classificação de série temporal univariada ou multivariada.

É um problema desafiador, pois não há maneiras óbvias ou diretas de relacionar os dados do sensor registrado à atividades humanas específicas e cada sujeito pode realizar uma atividade com variação significativa, resultando em variações nos dados do sensor gravado.

A intenção é registrar os dados do sensor e as atividades correspondentes para assuntos específicos, ajustar um modelo a partir desses dados e generalizar o modelo para classificar a atividade de novos assuntos não vistos a partir de seus dados de sensor.

**Conjuntos de dados e entradas**

O conjunto de dados utilizados no projeto (http://cs.unibo.it/projects/us-tm2017/download.html) foi desenvolvido na Universidade de Bolonha com o esforço de diferentes pessoas:

Marco Di Felice • Professor Associado • email: marco.difelice3@unibo.it

Luciano Bononi • Professor Associado • email: luciano.bononi@unibo.it

Luca Bedogni • Professor Assistente • email: luca.bedogni4@unibo.it

Vincenzo Lomonaco • Estudante de doutorado • email: vincenzo.lomonaco@unibo.it

Colaboradores anteriores

Claudia Carpineti • Mestranda • e-mail: claudia.carpineti@studio.unibo.it

Matteo Cappella • Aluno de mestrado • email: matteo.cappella@studio.unibo.it

Simone Passaretti • Aluno de mestrado • email: [simone.passaretti@studio.unibo.it](mailto:simone.passaretti@studio.unibo.it)

A coleta de dados foi controlada por um aplicativo Android em execução no telefone dos usuários enquanto eles realizavam atividades. Esse aplicativo, por meio de uma interface gráfica simples, permitiu que os voluntários gravassem seu nome, iniciassem e interrompessem a coleta de dados e rotulassem a atividade que estava sendo executada. Foi pedido aos usuários para usar o aplicativo durante atividades específicas, como caminhar, estar em um carro, em um trem, em um ônibus ou ficar parado. As atividades com estas abreviações:

T M = {bus, car, train, still, walking}

O aplicativo registra cada evento do sensor com uma frequência máxima de 20 Hz. Os eventos ocorrem toda vez que um sensor detecta uma alteração nos parâmetros que está medindo, fornecendo quatro informações:

* o nome do sensor que acionou o evento;
* o timestamp do evento;
* a acurácia do evento;
* os dados brutos do sensor que acionaram o evento.

Id:

Time:

activityrecognition#0:

activityrecognition#1:

android.sensor.accelerometer#mean:

android.sensor.accelerometer#min:

android.sensor.accelerometer#max:

android.sensor.accelerometer#std:

android.sensor.game\_rotation\_vector#mean:

android.sensor.game\_rotation\_vector#min:

android.sensor.game\_rotation\_vector#max:

android.sensor.game\_rotation\_vector#std:

android.sensor.gravity#mean:

android.sensor.gravity#min:

android.sensor.gravity#max:

android.sensor.gravity#std:

android.sensor.gyroscope#mean:

android.sensor.gyroscope#min:

android.sensor.gyroscope#max:

android.sensor.gyroscope#std:

android.sensor.gyroscope\_uncalibrated#mean:

android.sensor.gyroscope\_uncalibrated#min:

android.sensor.gyroscope\_uncalibrated#max:

android.sensor.gyroscope\_uncalibrated#std:

android.sensor.light#mean:

android.sensor.light#min:

android.sensor.light#max:

android.sensor.light#std:

android.sensor.linear\_acceleration#mean:

android.sensor.linear\_acceleration#min:

android.sensor.linear\_acceleration#max:

android.sensor.linear\_acceleration#std:

android.sensor.magnetic\_field#mean:

android.sensor.magnetic\_field#min:

android.sensor.magnetic\_field#max:

android.sensor.magnetic\_field#std:

android.sensor.magnetic\_field\_uncalibrated#mean:

android.sensor.magnetic\_field\_uncalibrated#min:

android.sensor.magnetic\_field\_uncalibrated#max:

android.sensor.magnetic\_field\_uncalibrated#std:

android.sensor.orientation#mean:

android.sensor.orientation#min:

android.sensor.orientation#max:

android.sensor.orientation#std:

android.sensor.pressure#mean:

android.sensor.pressure#min:

android.sensor.pressure#max:

android.sensor.pressure#std:

android.sensor.proximity#mean:

android.sensor.proximity#min:

android.sensor.proximity#max:

android.sensor.proximity#std:

android.sensor.rotation\_vector#mean:

android.sensor.rotation\_vector#min:

android.sensor.rotation\_vector#max:

android.sensor.rotation\_vector#std:

android.sensor.step\_counter#mean:

android.sensor.step\_counter#min:

android.sensor.step\_counter#max:

android.sensor.step\_counter#std:

sound#mean:

sound#min:

sound#max:

sound#std:

speed#mean:

speed#min:

speed#max:

speed#std:

target:

user:

**Descrição da solução**

Modelos de redes neurais profundas estão alcançando resultados de ponta para o reconhecimento da atividade humana. Eles são capazes de realizar o aprendizado de recursos automáticos a partir dos dados brutos do sensor e os modelos de desempenho superior se ajustam a recursos específicos do domínio criados manualmente.

*“[…], Os procedimentos de extração de características e construção de modelos são freqüentemente executados simultaneamente nos modelos de aprendizagem profunda. Os recursos podem ser aprendidos automaticamente através da rede, em vez de serem projetados manualmente. Além disso, a rede neural profunda também pode extrair uma representação de alto nível na camada profunda, o que a torna mais adequada para tarefas complexas de reconhecimento de atividades.”*

*— Deep Learning for Sensor-based Activity Recognition: A Survey, 2018.*

Existem duas abordagens principais para as redes neurais que são apropriadas para a classificação de séries temporais e que demonstraram ter um bom desempenho no reconhecimento de atividades usando dados de sensores de telefones inteligentes e dispositivos de rastreamento de condicionamento físico.

Eles são modelos de redes neurais por convolução e modelos de redes neurais recorrentes.

*“Recomenda-se que a RNN e a LSTM reconheçam atividades curtas que tenham ordem natural, enquanto a CNN é melhor em inferir atividades repetitivas a longo prazo. A razão é que a RNN poderia fazer uso da relação de ordem do tempo entre as leituras do sensor, e a CNN é mais capaz de aprender recursos profundos contidos em padrões recursivos.”*

*— Deep Learning for Sensor-based Activity Recognition: A Survey, 2018.*

**Modelo de referência (benchmark)**

O modelo de referência será o trabalho de detecção de modo de transporte realizado por equipe da Universidade de Bolonha, Itália:

Carpineti C., Lomonaco V., Bedogni L., Di Felice M., Bononi L., "Custom Dual Transportation Mode Detection by Smartphone Devices Exploiting Sensor Diversity", in Proceedings of the 14th Workshop on Context and Activity Modeling and Recognition (IEEE COMOREA 2018), Athens, Greece, March 19-23, 2018

Pré-impressão disponível: <https://arxiv.org/abs/1810.05596>

Em seu trabalho, utilizaram 3 conjuntos de dados, aplicando 4 algoritmos.

Para cada conjunto, foram construídos quatro modelos com quatro algoritmos de classificação diferentes:

* Decision Trees (DT)
* Random Forest (RF)
* Support Vector Machines(SVM)
* Neural Network (NN)

Os sensores incluídos no primeiro conjunto (parâmetro 1) foram acelerômetro, som e giroscópio. Esses três sensores possuem os maiores valores de precisão obtidos individualmente.

O primeiro conjunto de dados é formado por doze recursos, quatro para cada sensor. Foi realizada a classificação com os quatro algoritmos de classificação mencionados anteriormente. A precisão geral dos algoritmos está entre 82% e 88%. Mesmo que a floresta aleatória produza os maiores valores de precisão (88%), todos os algoritmos têm um desempenho substancialmente bom.

Ao expandir o conjunto de dados adicionando todos os outros sensores relevantes, exceto a velocidade, para fins de economia de bateria, foram alcançados melhores resultados em termos de precisão. Com o segundo conjunto de dados, formado por oito sensores e trinta e dois recursos, a precisão aumenta até valores entre 86% e 93%.

Por fim, foi treinado um modelo no terceiro conjunto de dados formado por todos os nove sensores relevantes e trinta e seis recursos, diferindo do anterior apenas para recursos derivados de velocidade. O resultado mostra como se considera a velocidade, aumentando ainda mais a capacidade do modelo de inferir qual modo de transporte o usuário está usando atualmente. Neste último caso, a precisão atingiu um nível de alcance entre 91% e 96%.

**Métricas de avaliação**

*(aprox. 1-2 parágrafos)*

Nesta seção, proponha ao menos uma métrica de avaliação que pode ser usada para quantificar o desempenho tanto do modelo de benchmark como do modelo de solução apresentados. A(s) métrica(s) de avaliação proposta(s) deve(m) ser adequada(s), considerando o contexto dos dados, da definição do problema e da solução pretendida. Descreva como a(s) métrica(s) de avaliação pode(m) ser obtida(s) e forneça um exemplo de representação matemática para ela(s) (se aplicável). Métricas de avaliação complexas devem ser claramente definidas e quantificáveis (podem ser expressas em termos matemáticos ou lógicos)

**Design do projeto**

*(aprox. 1 página)*

Nesta seção final, sintetize um fluxo de trabalho teórico para obtenção de uma solução para o problema em questão. Discuta detalhadamente quais estratégias você considera utilizar, quais análises de dados podem ser necessárias de antemão e quais algoritmos serão considerados na sua implementação. O fluxo de trabalho e discussão propostos devem estar alinhados com as seções anteriores. Adicionalmente, você poderá incluir pequenas visualizações, pseudocódigo ou diagramas para auxiliar na descrição do design do projeto, mas não é obrigatório. A discussão deve seguir claramente o fluxo de trabalho proposto para o projeto de conclusão.

**Antes de enviar sua proposta, pergunte-se. . .**

* A proposta que você escreveu segue uma estrutura bem organizada, similar ao modelo de projeto?
* Todas as seções (em especial, **Descrição da solução** e **Design do projeto**) estão escritas de uma forma clara, concisa e específica? Existe algum termo ou frase ambígua que precise de esclarecimento?
* O público-alvo de seu projeto será capaz de entender sua proposta?
* Você revisou sua proposta de projeto adequadamente, de forma a minimizar a quantidade de erros gramaticais e ortográficos?
* Todos os recursos usados neste projeto foram corretamente citados e referenciados?