Черновик курсовой
Методы сравнения траекторий

Кудрявцев Георгий Алексеевич

23 апреля 2015 г.

Содержание

1 Введение

С развитием Интернета, географическая информационная система (ГИС) и Location-based service(LBS) играют важную роль в различных приложениях. Все больше и больше устройств способны собирать, обрабатывать и хранить информацию о местоположении подвижных объектов. Таким образом, огромное количество информации о местоположении объектов представленно в форме траекторий.

Анализ сходства траекторий применяется в таких областях как экология, биология, телематика, геоинформатики.

В биологии и экологии сходства траекторий используются для наблюдения за миграцией животных [6]. Анализ траекторий и маршрутов диких животных является важнейшим элементом их изучения, а так же хороший показатель экологической обстановки в природе. Важной подзадачей в ислледовании миграций является нахождей главных маршрутов. Маршруты диких животных довольно хаотичны, поэтому для биологов важно определять главные пути, по которым перемещаются звери [5].

Также анализ сходства траекторий применяется в распознавании рукописных текстов. [1]

В телематике анализ траекторий играют важнейшую роль. При помощи него выполняются такие задачи как оптимизации маршрутов, возможность обнаружения угнанного транспорта, а также навигация водителей в незнакой местности.

Последние исследования в этой области посвящены нахождения различных патернов в стиле вождения водителей. [8, 9] В частности телематика представляется собой хороший способ оценки риска страхового случая водителя. При помощи анализа траекторий можно непосредственно оценивать поведение водителя на дороге. На Kaggle прошел Driver Telematics Analysis, который как раз посвещен этой теме.

Далее будет обзор способов измерения сходства траекторий. Мое решение конкурса Driver Telematics Analysis, а также решение других участников.

2 Существующие метрики

2.1 Расстояние Хаусдорфа(HF)

Пусть у нас есть два набора точек(траекторий) A и B. d – расстояние между двумя точками. Тогда расстояние Хаусдорфа между траекториями H.

$$\begin{aligned} H(A,B) &= \max(h(A,B), h(B,A)) \\ h(A,B) &= \max_{a_i \in A}(\min_{b_j \in B} \ d(a_i, b_j)) \\ h(B,A) &= \max_{b_i \in B}(\min_{a_j \in B} \ d(b_i, a_j)) \end{aligned} \tag{1}$$

2.2 Модифицированное расстояние Хаусдорфа(MHF)

Расстояние Хаусдорфа чувствительно к выбросам. Всего одна точка может сильно изменить ответ. Чтобы метрика стала менее чувствительнак выбросам, ее можно немного подкорректировать.

$$h(A,B) = \frac{1}{m_a} \sum_{a_i \in A}(\min_{b_j \in B} \ d(a_i, b_j)) \tag{2}$$

2.3 Интерполяция на основе модифицированного расстояния Хаусдорфа(IMHF)

Чтобы уменьшить чувствительность расстояния между траекториями от точек измерения можно проводить интерполяцию по точкам траекторий. В данном случае берется среднее арифсетическое. Этот метод является одним из самых лучших основанных на метрике Хаусдорфа.

$$h(A,B) = \frac{1}{m_a} \sum_{a_i \in A} (\min_{\overline{b_j, b_{j-1}} \in B} d(a_i, \overline{b_j, b_{j-1}})) \tag{3}$$

2.4 OWD (One Way Distance)

Простой и эффективный способо вычисления расстояния между траетриями Сначала задается расстояние между точкой p и всей траекторией Tr.

$$D_{point}(p, Tr) = \min_{q \in Tr} d(p, q) \tag{4}$$

Далее находим расстояние между траекториями Tr_1 и Tr_2 следующим образом.

$$\begin{aligned} D_{owd}(Tr_1, Tr_2) &= \frac{1}{|Tr_1|} \sum_{p \in Tr_1} (D_{point}(p, Tr_2) \\ D(Tr_1, Tr_2) &= \frac{1}{2}(D_{owd}(Tr_1, Tr_2) + D_{owd}(Tr_2, Tr_1) \end{aligned} \tag{5}$$

2.5 Dynamic Time Warping

Впервые этот методы был успешно применен в разпознавании речи [7]. В настоящее время активно используется в распознавании жестов, рукописных текстов, наблюдением за дикими животными. Рассмотри две траектории

$$\begin{aligned} Q &= q_1, q_2, q_3 ... q_n \\ C &= c_1, c_2, c_3 ... c_m \end{aligned} \tag{6}$$

Сначала строится матрица расстояний D порядка $n \times m$, где $D_{ij} = d(i, j)$ Затем строится матрица трансформаций K, где $K_{ij} = D_{ij} + min(D_{i-1j}, D_{i-1j}, D_{i-1j-1})$ После заполнения матрицы деформации строится путь трансформации. Это последоватьность элементов матрицы трансформации, которая минимизирует расстояние между траекториями.

Пусть p путь трансформации $p = (p_1, p_2, ...p_k)$, где $p_l = (q_i, c_j)$. Тогда она должна удовлетворять следующим требованиям.

Граничные условия: $p_1 = d(q_1, c_1)$ и $p_k = d(q_n, c_m)$. Это означает, что пусть трансформации начинается на начальные точек траекторий и кончается на конечных точек траекторий.

Непрерывность: Для $p_l = d(q_i, c_j)$ и $p_{l-1} = d(q_{i'}, c_{j'})$ выполняется $i - i' \leq 1$, $j - j' \leq 1$. Это значит, что путь трансформации состоит из смежных ячеек матрицы трансформаци

Монотонность: Для $p_l = d(q_i, c_j)$ и $p_{l-1} = d(q_{i'}, c_{j'})$ выполняется $q_i - q_{i'} \geq 0$ и $c_j - c_{j'} \geq 0$. Это гарантирует, что путь трансформациии не будет проходит через одну точку несколько раз.

Среди всех возможных путей трансформации, которые удовлетворяют условиям, выбирается минимальный. DTW расстояние между двумя последовательностиями через оптимальный путь трансформации выражается следующим образом.

$$D_{DTW}(Q, C) = \frac{p_k}{k} \tag{7}$$

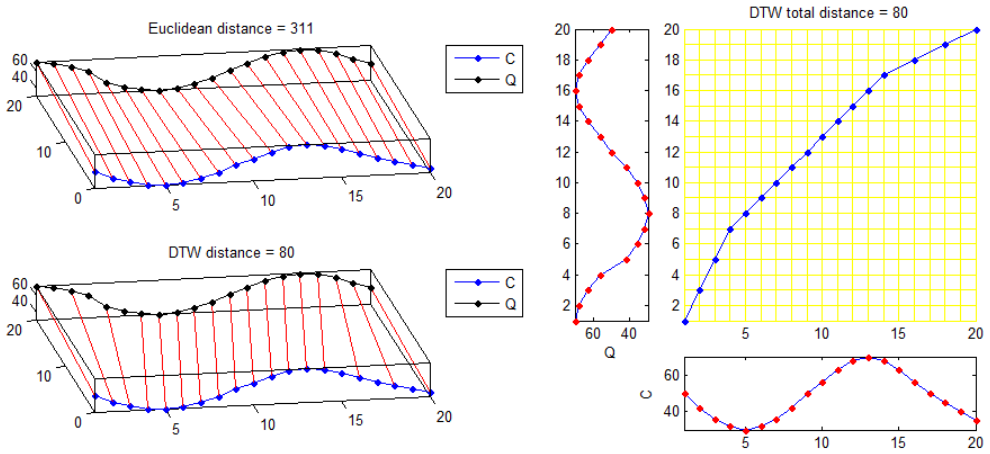

Рис. 1: Dynamic Time Warping

3 Конкурс Driver Telematics Analysis

На платформе Kaggle проводился конкурс Driver Telematics Analysis. Организаторы конкурса предложили следующую задачу. Дан набор данных из 2736 водителей. Каждому водителей соответствует набор из 200 траекторий. Известно что из 200 траекторий большинство являются траекториями данного водителями, остальные описывают движение других водителей.

Также организаторы преобразовывали траектории для деанонимизации траекторий. Траектории были повернуты на случайные углы. Начало и конец траектории были вырезаны. После этого траектория была смещена так, чтобы начальная точка была (0, 0).

Требуется определить какие траектории принадлежат данному водителю, а какие нет.

3.1 Авторское решение

3.1.1 Способ обучения

Данная задача является задачей без учителя. Но она была сведенена к задаче к учителем. Всем двумстам трактириям данного водителя ставилась метка первого класса(1). Затем в эту выборку добавлялось еще двести траекторий, которые принадлежат другим водителям, и им ставилась метка второго класса 0. Затем запускался алгоритм машиного обучения на этой выборке. Чтобы скомпенсировать то, что прибавлялись случайные трактории при создании тренировочной выборки, процесс обучение и получения ответы проводился несколько раз. Этот метод должен был работать главным образом из-за того, что в условии задачи говорилось, что большинство траекторий принадлежали данному водителю.

3.1.2 Способ кросс-валидации

К 200 траекториям добавлялось еще 50 случайных траекторий. И всем им ставилась метка первого класса(1). Потом добалялось еще 250 случайных траекторий и ставилась метка второго класса(0). Далее проводилось обучение на этой выборке. А ответ проверялся на первых 250 траекторий, причем для первых 200 траекторий за правильный ответ ставилась метка 1, а последним 50 метка 0.

3.1.3 Извлечение статистических признаков

Статистические признаки были следующие: среднее значение, дисперсию, максимальное значение скорости и ускорения, время простоя и максимальное время разгона. Отноше-

ние среднего значения и дисперсии к максимальному значению к максимальной скорости. Аналогичные признаки для изменения угла направления относительно предыдущей точки и относительно центра масс.

Были проведены попытки использования гистограммами скоростей и ускорений, но они почему-то ухудшали результат. Обучение проводилось при помощи Random Forest, GBM, Logistc Regression.

Алгоритм	Результат на лидерборде
Random Forest	0.86116
GBM	0.84279
Logistic Regression	0.74810

Таблица 1: Точность разных алгоритмов на лидерборде

Лучший результат показал Random Forest. Все остальные алгоритмы показывали результат хуже.

3.1.4 Инвариантные к пововоротам траектории

Было решено использовать DTW. Но эта метрика не является инвариантной к поворотам и сдвигам. Пакетов с инвариантном к поворотам DTW не было найдено. Поэтому было решено реализовать несколько подходов с прочитанных статей.

3.1.5 Реализация статьи Rotation Invariant Distance Measures for Trajectories

Этот метод был создан для сравнения рукописных чисел [1]. Основная идея – перевод трактории к инвариантной к поворотам системе координат. Пусть $P = [P_1, ..P_n]$ – траектория.

$V_t = P_t - P_{t-1}$ – модуль скорости

V_{ref} – направление скорости относительно центра масс

$\alpha_t = sign(V_t, V_{ref}) \cdot arccos(\frac{<V_t V_{ref}>}{\|V_t\|\|V_{ref}\|})$, где функция sign определена следующим образом

$$sign(V_t, V_{ref}) = \begin{cases} 1 & \text{если } [V_t \times V_{ref}] \cdot [0\ 0\ 1]^T > 0 \\ -1 & \text{если } [V_t \times V_{ref}] \cdot [0\ 0\ 1]^T \leq 0 \end{cases}$$

Полученное α_t есть изначальная траектория в новой системе координат. Далее в новой системе координат выполнить DTW.

Основным минусом это метрики – инвариантность к масштабу. Поэтому вытянутые траектории слабо различимы с малым количеством поворотов.

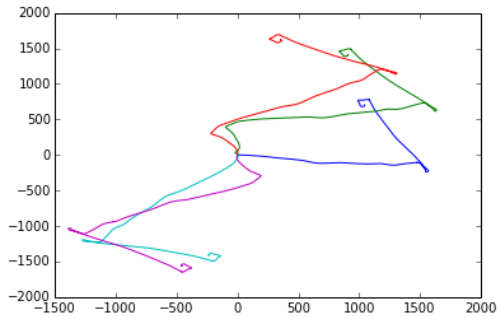

Рис. 2: Хорошая работа алгоритма

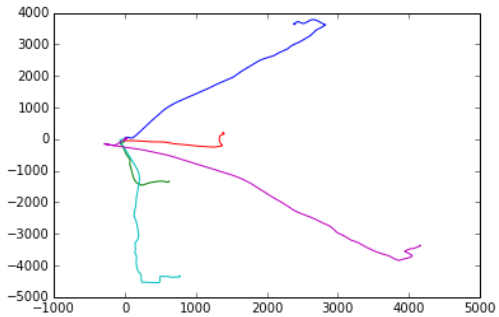

Рис. 3: Плохая работа алгоритма

В этом методе качество кластеризации заметно уменьшается с увеличением количества кластеров. Но также результат зависит от тех траекторий, которые были созданы.

3.1.6 Реализация статьи Affine Invariant Dynamic Time Warping and its Application to Online Rotated Handwriting Recognition

Этот метод был также создан для сравнения рукописных чисел [2]. Основная идея – использовать EM алгоритм для потората и сдвига траекторий для их наилучшего приближения. Пусть даны две последовательности одинаковой длины. $R = [r_1, r_2, ..., r_n]$
$T = [t_1, t_2, ..., t_n]$ = Надо минимизировать следующий функционал

$$\sum_{1}^{k} \|t_i - r_i A\|, \text{где A - матрица поворота.} \tag{8}$$

Т.е. надо найти такую матрицу повора, которая лучше всех сближала две траектории. Эту задачу можно решить простым способом. Пусть $D_r = [r_{w_1}, r_{w_2}, ...r_{w_k}]^T$
$D_t = [t_{w_1}, t_{w_2}, ...t_{w_k}]^T$

Тогда матрицу поворота можно найти при помощи равенства нулю производной. $A = (D_r^T D_r)^{-1} D_r^T D_t$ Но при больших размера траектории эта задача становится трудоемкой. Поиск более быстрого решения является ортогональной проблемой Прокруста. Эта задача хорошо решается при помощи алгоритма Кабша [4].

Но мы имеем траектории разной длины в общем случае. Эту проблему авторы статьи решили следующим образом. Они применили EM-алгоритм, в котором на E-шаге минимизируется функционал по пути транформации, а на M - шаге получаю новый путь трансформации.
Инициализация:
минимальный путь трансформации $w^{(0)} = DTW\_PATH(T, R)$
k = 1
Пока не сошлось выполнять:

$$A^{(k)} = \underset{A}{\operatorname{argmin}} \sum_{1}^{k} \|t_{w_i^{(k-1)}} - r_{w_i^{(k-1)}} A\|$$
$$w^{(k)} = DTW\_PATH(T, R)$$
$$k = k + 1$$

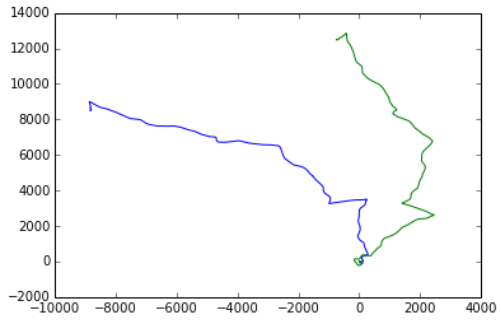

Рис. 4: Изначальное положение траекторий

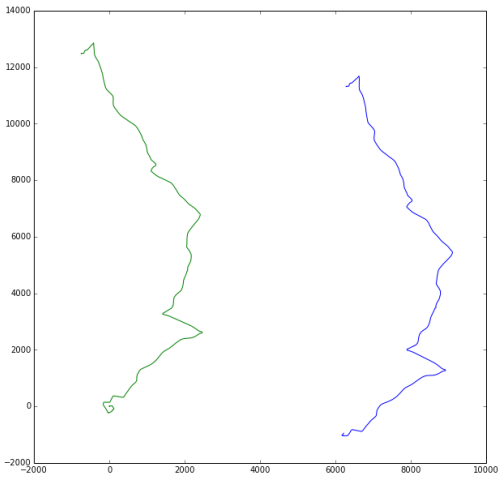

Рис. 5: Первый проход EM алгоритма

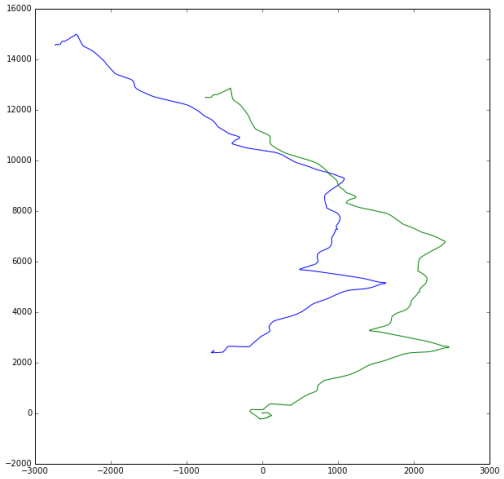

Рис. 6: Четвертый проход EM алгоритма

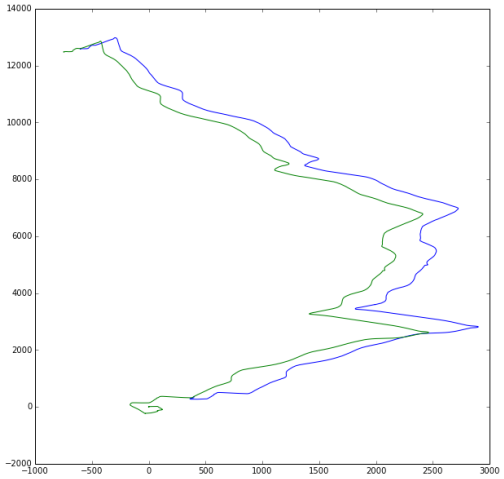

Рис. 7: Десятый проход EM алгоритма

3.1.7 Генерация выборки и сравнение алгоритмов

Проверять качество метрик непосредственно на платформе Kaggle было неправильно, т.к. в конкурсе надо было находить траектории принадлежавшие одному водителю, а траектории могли принадлежать одному водителю и одновременно абсолютно непохожими друг на друга.

Сравнение алгоритмов были произведенено на искуственной выборке. Одна половина выборки состояла из шума, т.е траекторий разных водителей. Другая состояла из искусственных кластеров. Кластеры были созднаны также, как делали организаторы конкурса. Бралась траетория, поворачивалась на случайный угол, а также с начала и конца удалась случайная по длине(но не больше, чем 5% от длины траектории) часть траектории. Для каждой траектории этот процесс повторялся несколько раз для создания кластера.

Далее из траекторий удалялись лишние точки. Это очень важно для быстрой скорости работы. Строилась матрица расстояний. И по этой матрице производилась кластеризация при помощи метода DBscan. Качество кластеризации проверялось при помощи метрики Rand index.

В среднем второй метод работает лучше, чем первый. Это связано с тем, что второй метод не является инвариантным к масштабированию.

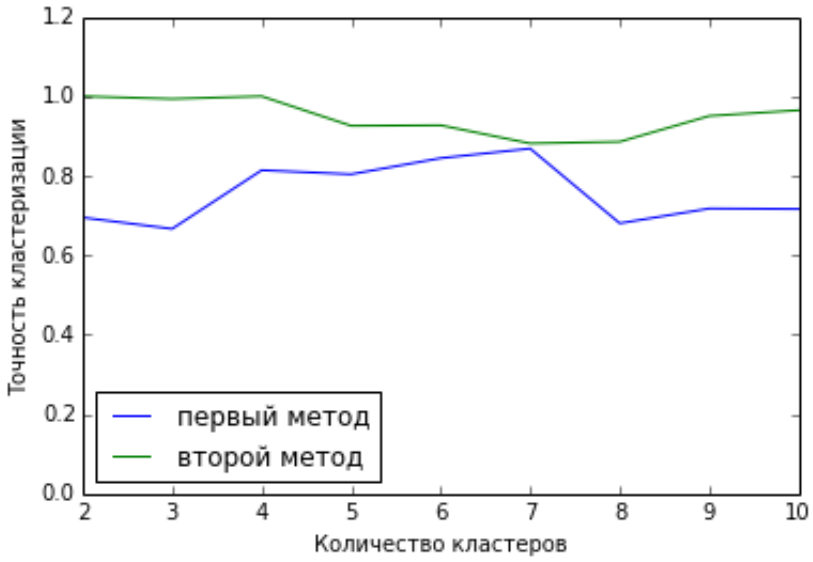

Рис. 8: Сравнение первого и второго методов

3.2 Метрики других участников

Для проблемы Прокруста необходимы траектории одинаковой длины. Другие участники обошли эту проблеме иным образом. Они сравнивали части траекторий одинаковой длины между собой. В этой задаче эта метрика подходит лучше, так как было много траекторий, которые были частью другой. Мои представленные метрики это не отлавливали.

Другое решение было такое. Строилась гистограмма углов поворотов водителей на каждой траектории. Углы брались по модулю, т.е. не имело значение в какую сторону производился поворот. Затем данные гистограммы сравнивались следующим образом. Пусть $A = [a_1, a_2, ..., a_n]$ и $B = [b_1, b_2, ..., b_n]$

$$D(A, B) = \frac{\sum_{i=1}^{n} |a_i - b_i|}{\sum_{i=1}^{n} a_i + b_i} \tag{9}$$

Затем в зависимости от количества близких траекторий увеличивался ответ статистической модели для данной траектории.

Список литературы

[1] Affine Invariant Dynamic Time Warping and its Application to Online Rotated Handwriting Recognition.

[2] Rotation Invariant Distance Measures for Trajectories

[3] A Dynamic Time Warping based Algorithm for Trajectory Matching in LBS

[4] Kabsch, Wolfgang, "A solution for the best rotation to relate two sets of vectors

[5] Mapping migratory flyways in Asia using dynamic Brownian bridge movement models

[6] On a Wildlife Tracking and Telemetry System: A Wireless Network Approach

[7] Fundamentals of speech recognition

[8] Driving Behavior Improvement and Driver Recognition Based on Real-Time Driving Information

[9] Driving Style Recognition for Co-operative Driving: A Survey