

Определение местоположения по сигналам акселерометра

Макаров М. В.

Мы рассматриваем задачу определения местоположения человека по данным акселерометра телефона и другим вспомогательным данным. Уже существуют некоторые методы решения данной задачи [1]. В данной статье акцент делается на использовании дополнительной информации, например сигналов гироскопа или магнетометра, для повышения точности.

1 Введение

В данной работе рассматривается задача определения местоположения человека по данным акселерометра его телефона. Данная задача актуальна как часть более общей проблемы определения местоположения. Поскольку акселерометры энергоэффективны и не требуют для работы наличие внешних устройств, таких как спутник или радиоточка, точные методы решения этой задачи востребованы.

В силу того, что акселерометры, используемые в мобильных устройствах, неточны, и наивное решение поставленной задачи путём двойного интегрирования даёт путь, значительно отклоняющийся от истинной траектории. В [1] эта проблема решается использованием информации о том, где находится телефон во время перемещения человека.

Местоположение также можно определить, используя данные других датчиков, таких как магнитометр [2] и гироскоп [3].

В данной работе рассматривается метод, опирающийся как на априорные знания о положении телефона, так и на данные с магнитометра и гироскопа. Это позволяет использовать фильтр Маджвика для предобработки сигналов. После чего для вычисления скорректированных значений вектора ускорения используется метод PLS.

2 Постановка задачи

Данные с датчиков представляются виде временного ряда $s = \{(\mathbf{c}(t), \mathbf{r}(t), \mathbf{b}(t)) | t \in T\} \in \mathbb{R}^{9T} = \mathbb{X}$, составленного из показаний акселерометра, гироскопа и магнитометра по 3 координатам, где $T = \{t_1, \dots, t_m\}$ — множество моментов в которые проводились измерения. Аналогично, ряд истинных положений объекта имеет вид $y \in \mathbb{R}^{3T} = \mathbb{Y}$. Обозначим через $\mathbf{X} = \{s_i | i \in \mathcal{I}\}$ матрицу всех рядов выборки, а через $\mathbf{y} = \{y_i | i \in \mathcal{I}\}$ — матрицу положений объекта в соответствующие моменты времени.

Предполагается, что

- В момент времени t_1 базис системы отсчёта объекта совпадает с системой отсчёта, относительно которой происходят измерения перемещения.
- Погрешности измерений $\mathbf{c}_i(t)$, $\mathbf{r}_i(t)$, $\mathbf{b}_i(t)$ независимы и имеют нулевое матожидание.
- Каждый элемент выборки был получен при фиксированном расположении смартфона на теле человека — в сумке, в руке, на теле или на ноге — соответственно классы $P = \{0, 1, 2, 3\}$.

Требуется найти такую модель

$$f : \mathbb{X} \rightarrow \mathbb{Y}, f(s) = \hat{y}$$

что среднеквадратичная функция ошибки

$$S(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{t \in T} \|y_i(t) - \hat{y}_i(t)\|^2$$

минимальна.

В данной работе рассматриваются модели следующего вида: с помощью фильтра Маджвика M получается ряд сглаженных значения вектора ускорения, $M(s) = \hat{c}$, после чего с помощью вспомогательной модели $g : \mathbb{X} \rightarrow P, g(\hat{c}) = p$ определяется, к какому классу расположения относится данная траектория. После чего используется модель $f_p(\hat{c}) = \tilde{c}$ дальнейшего уточнения вектора ускорения. Наконец, ускорения \tilde{c} дважды интегрируются для получения оценки \hat{y} . Обозначим это преобразование за I . Для моделей f_p, g используется метод опорных векторов.

Таким образом, определяя $\tilde{f}(s) = I(f_{g(M(s))}(M(s)))$ формальная постановка задачи такова:

$$\mathbf{f}^* = \arg \min_{g, f_1, \dots, f_3} S(\tilde{f}(\mathbf{X}), \mathbf{y})$$

3 Базовый алгоритм

В качестве базового алгоритма рассматривается численное интегрирование сигнала акселерометра без фильтрации сигнала в предположении, что ориентация смартфона не меняется во времени, то есть

$$f(s)(t_i) = \sum_{j=2}^i \left(\sum_{k=2}^{j-1} \mathbf{c}_k(t_k - t_{k-1}) + \frac{1}{2} \mathbf{c}_j(t_j - t_{j-1}) \right) (t_j - t_{j-1}), f(s)(t_1) = 0.$$

Как известно, интегрирование даёт неточный результат на шумном входе.

Более точный алгоритм предложен в [1].

4 Базовый эксперимент

Для получения матрицы признаков X предварительно из данных удаляются высокочастотные шумы с помощью сглаживания Гаусса. Полученные линейные и угловые скорости уже преобразуются в вектор признаков.

В качестве модели рассматривается каскадная регрессия, состоящая из классификатора положения датчиков и семейства регрессоров на каждый из классов, как в [1].

По матрице признаков X классификатор определяет, какому расположению датчиков (в руке, на ноге, на поясе, в сумке) соответствует данное описание, т.е. решает задачу многоклассовой классификации с 4 классами.

После этого для каждого класса на тренировочной выборке обучается свой регрессор, который выдает скорости движения пешехода для каждого временного блока. Полученные скорости содержат ошибку, связанную с неточностями инерционных датчиков, поэтому далее для этих скоростей находится смещение (по предположению низко-частотное) из следующей задачи минимизации:

$$\min_{\{x_I^1, x^5 1_I, \dots\}} V_{bias} = \min_{\{x_I^1, x^5 1_I, \dots\}} \sum_{f \in F_2} \|v_C^F - v_R^f\| + \lambda \sum_{f \in F_1} \|x_I^f\|^2,$$

$$v_C^f = R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'} (a_I^{f'} + x_I^{f'}),$$

где f - единица блока выборки, F - блок выборки, v_C^F - скорректированное значение скорости, v_R^f - предсказанное значение скорости, I - система координат устройства, W - глобальная система координат, S - IMU-стабилизированная система координат, R_{AB} - матрица перехода из системы координат B в систему координат A .

Также для каждого класса создается регрессор, предсказывающий угловые скорости пешехода в каждом временном блоке.

На контрольной выборке для классификатора и каждого регрессора подбираются оптимальные значения гиперпараметров.

По полученным значениям скоростей восстанавливается траектория пешехода.

Для эксперимента используется часть данных, собранных в статье, описывающей алгоритм RIDI [1]. Эти данные были получены с помощью инерционных датчиков, расположенных в смартфонах в нескольких случаях: когда смартфон располагался на поясе, в руке, на ноге или в сумке. Выборки содержат различные траектории длиной в 100 минут и частотой сигнала 200 Гц. В качестве объекта рассматривается положение в определенный момент времени i . Признаками объекта являются угловые скорости и линейные ускорения в стабилизированной системе координат датчиков в моменты времени $i - window_size, \dots, i$, где $window_size$ - размер окна (равен 200). Целевыми переменными являются метки классов, характеризующие то, в каком положении находился смартфон при получении определенных данных, а также скорости в данный момент времени i , которые вычисляются через координаты пешехода и прошедшее время. По полученным данным после уточнения скоростей с помощью оптимизации функции V_{bias} строится предсказанная траектория пешехода.

Цель эксперимента: подобрать такие модели и их параметры, что предсказанная траектория пешехода будет наиболее близкой к истинной.

Формально алгоритм описывается следующим образом:

Вход: $X, Y_{class}, Y, X_{test}$

- 1: *initialize classifier_options*
- 2: *classifier = Classifier(classifier_options);*
- 3: *classifier.fit(X, Y_{class})*
- 4: **для** *cls in classes*:
- 5: *initialize regressor_cls_options*
- 6: *regressor_cls = Regressor(regressor_cls_options)*
- 7: *regressor_cls.fit(X[X[ind] ∈ cls], Y[Y[ind] ∈ cls])*
- 8: *Y_{test-class} = classifier.predict(X_{test})*
- 9: **для** *cls in classes*:
- 10: *Velocity_cls = regressor_cls.predict(X_{test}[Velocity_class[ind] == cls])*
- 11: $x_I^1, x_I^5, \dots = \arg \min_{\{x_I^1, x_I^5, \dots\}} V_{bias_cls}$
- 12: $Velocity_cls = R_{SW}^f \sum_{f'=1}^f R_{WI}^{f'}(a_I^{f'} + x_I^{f'})$
- 13: *Trajectory_cls recovery depending on Velocity_cls*
- 14: **return** *Full_trajectory*

5 Рассмотрение моделей

Вся выборка была разбита на обучающую и тестовую, причём данные в тестовой выборке были сняты с других людей. Для моделей производился подбор гиперпараметров с помощью кросс-валидации по тренировочной выборке, а также измерялось их качество на тестовой выборке.

В качестве первого класса моделей рассматривался метод ближайших соседей. Подбирались следующие параметры: число соседей, метрика и функция весов.

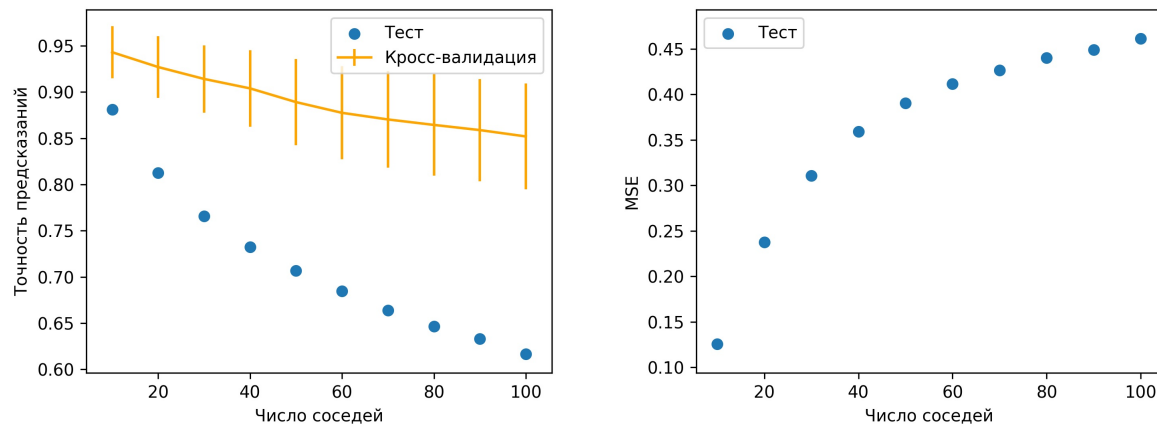


Рис. 1 Точность предсказания класса и ошибка на тесте, kNN

В итоге $l1$ -метрика оказалась наилучшей, а наилучшей функцией весов оказалось расстояние.

Литература

- [1] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. *CoRR*, abs/1712.09004, 2017.
- [2] W. Kang and Y. Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 15(5):2906–2916, May 2015.
- [3] Jian Kuang, Xiaoji Niu, and Xingeng Chen. Robust pedestrian dead reckoning based on mems-imu for smartphones. *Sensors*, 18(5), 2018.