Fingerprinting Localisation z wykorzystaniem algorytmu XGBoost[1]

Bartosz Topolski , Magdalena Mazurek

8 grudnia 2016

[1]The research is supported by the National Centre for Research and Development, grant No PBS2/B3/24/2014, application No 208921.

Zagadnienie lokalizacji

Główny cel: ustalenie położenia użytkownika w przestrzeni.

Trzy podstawowe sposoby:

- triangulacja
- trilateracja
- **fingerprinting**

Fingerprinting

Technika składająca się z dwóch faz: w fazie treningowej (offline) zbierane są pomiary siły sygnałów tworząc siatkę punktów referencyjnych. Następnie na ich podstawie buduje się algorytm który w fazie śledzenia (online) wykorzystywany jest do obliczenia położenia użytkownika.

Fingerprinting

- jest najczęściej stosowaną techniką lokalizacji w budynkach,
- bardziej odporna na zakłócenia niż pozostałe metody,
- **data science!**

Ale:

- wymaga wykonania dokładnych pomiarów pokrywających jak największą część budynku,
- mniej odporny na awarie sieci oraz zmiany w infrastrukturze.

Pomiary

Rysunek 1: Przykład aparatury do zbierania pomiarów [1]

Pomiary

- wykonywane w budynku MiNI
- 570 punktów dostępu (w tym 46 bezpośrednio z infrastruktury wydziałowej)
- zbiór treningowy 2676 punktów w siatce o boku 1.5m
- zbiór testowy 2794 punktów przesunięty o 0.75m względem treningowego
- 40 pomiarów w każdym punkcie
- poza siłą sygnałów mierzone były także inne wielkości, na przykład ciśnienie atmosferyczne, nie były jednak one używane w analizie

Dotychczasowe podejścia

Przy poprzednich analizach przeprowadzanych na danych z naszego wydziału autorzy używali między innymi następujących algorytmów:

- k Nearest Neighbours
- Multilayer Perceptron
- Random Forest

Najlepsze wyniki uzyskane zostały za pomocą lasów losowych.

Ogólny schemat algorytmu

Na danych testowych zostały zbudowane trzy modele - po jednym dla współrzędnych x oraz y oraz dla numeru piętra. Modele szacujące współrzędne horyzontalne były budowane na całych danych, bez podziału na piętra. Dla nowego sygnału dokonujemy po prostu niezależnej predykcji każdej z trzech zmiennych.

Dodatkowo zaproponowany został system obsługi awarii punktów dostępu.

Lasy losowe

- możliwość szybkiego uczenia modeli i dokonywania predykcji
- pozwalają wyznaczyć istotność zmiennych
- ogólne cechy modeli opartych na drzewach decyzyjnych dobrze pasują do postawionego problemu:
  - brak konieczności zakładania rozkładu szacowanej wartości
  - wykrywanie nieliniowych zależności
  - wykrywanie głębokich interakcji między zmiennymi niezależnymi

XGBoost - wprowadzenie

- metoda iteracyjna
- oparta o drzewa decyzyjne (komitet drzew)
- kolejne drzewa są budowane tak, aby poprawić predykcję z poprzedniego kroku
- szczególny przypadek tzw. Gradient Boosting Machines (GBM)

GBM - ogólna idea

Załóżmy, że operujemy na zbiorze danych $\mathcal{D} = \{(x_i, y_i) : i = 1 \dots n,\ x_i \in \mathbb{R}^m,\ y_i \in \mathbb{R}\}$. Dla komitetu modeli $\hat{y}$ możemy napisać

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i). \tag{1}$$

W przeciwieństwie do drzew losowych, w k-tym kroku budujemy drzewo f_k w taki sposób, aby zminimalizować ogólną funkcję celu:

$$\mathcal{L}(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{2}$$

GBM - ogólna idea

$\mathcal{L}$ jest tutaj zadaną przez użytkownika funkcją straty, a $\Omega(f_k)$ to funkcja penalizująca złożoność drzewa f_k:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2, \tag{3}$$

gdzie T to ilość liści drzewa f_k, a w to wektor wag w liściach.

Wprowadzenie kary za złożoność drzewa jest pierwszą funkcją odróżniającą algorytm XGBoost od ogólnych GBM.

XGBoost vs GBM

Pozostałe modyfikacje wprowadzone przez autorów to m. in. :

- możliwość zmniejszenia wpływu pojedynczego drzewa poprzez przeskalowanie jego wag
- możliwość budowania pojedyńczego drzewa na losowym podzbiorze zmiennych (analogicznie do lasów losowych)
- wielowątkowe szukanie optymalnego punktu podziału w drzewie
- możliwość przybliżonego znajdowania punktu podziału

Najważniejszymi cechami algorytmu XGBoost są ogólna szybkość działania i możliwość zrównoleglenia obliczeń. Twórcy udostępnili biblioteki do najpopularniejszych języków: R, Python, Java, Scala. Dostępna jest także biblioteka dedykowana do użycia w środowisku Spark.

Procedura testowa

Poza zmianą samego algorytmu, wprowadziliśmy także jedną różnicę w całej procedurze. Dla każdego piętra budujemy oddzielny model do predykcji współrzędnych x oraz y. Dzięki temu model jest w stanie lepiej dopasować się do rozkładu sił sygnałów na każdym z pięter.

Porównanie modeli przeprowadziliśmy zarówno na wszystkich punktach dostępu, jak i na ograniczonym zbiorze 46 punktów z sieci wydziałowej. Ostatnie podejście jest zarówno bardziej wymagające od samego modelu, jak i bardziej uzasadnione z praktycznego punktu widzenia.

Dopasowanie modelu porównywaliśmy w oparciu o średnią, medianę i kwantyl rzędu 80% błędów horyzontalnych oraz o odsetek błędnych klasyfikacji numeru piętra.

Wyniki

Dla modeli opartych o wszystkie dostępne dane otrzymaliśmy następujące wyniki:

model	HME	mean error	80%	ACC
XGBoost (2 models)	2.44	3.32	4.54	0.95
XGBoost (12 models)	2.34	3.37	4.42	0.95
Random forest	2.78	3.80	5.14	0.94
kNN	2.39	3.00	4.19	0.93

Wyniki

Dla modeli opartych o dane z sieci wydziałowej otrzymaliśmy następujące wyniki:

model	HME	ACC
XGBoost(2 models)	2.92	0.93
XGBoost(12 models)	2.81	0.93
Random forest	4.47	0.93
kNN	3.13	0.91

Tabela 1: Results of localisation on 46 access points

Podsumowanie

Podsumowując:

- udało nam się poprawić otrzymane wcześniej rezultaty
- szczególną poprawę można było zaobserwować w najbardziej praktycznym przypadku - czyli przy użyciu infrastruktury wydziałowej
- omówiliśmy jednak tylko jedną metodę, i być może otrzymane wyniki da się jeszcze bardziej poprawić

Bibliografia

- Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In 22nd SIGKDD Conference on Knowledge Discovery and Data Mining, 2016. https://arxiv.org/abs/1603.02754
- Grzenda, M.: On the prediction of floor identification credibility in rss-based positioning techniques. Lecture Notes in Computer Science, vol. 7906, pp. 610-619. Springer (2013), http://dx.doi.org/10.1007/978-3-642-38577-3_63
- Karwowski, J., Okulewicz, M., Legierski, J.: Application of particle swarm optimization algorithm to neural network training process in the localization of the mobile terminal. Communications in Computer and Information Science, vol. 383, pp. 122131. Springer (2013), http://dx.doi.org/10.1007/978-3-642-41013-0_13
- Górak, R., Luckner, M.: Malfunction Immune Wi-Fi Localisation Method
- Górak, R., Luckner, M.: Comparison of Floor Detection Approaches for Suburban Area