Map matching theo thời gian thực

Bài toán:

Trên ứng dụng VCar, có nhóm tính năng hỗ trợ người lái xe trên hành trình.

Cụ thể là 2 tính năng:

- F1: Báo vượt tốc: khi xe đi vượt quá tốc độ quy định sẽ báo với chủ xe. Tốc độ quy định sẽ theo dữ liệu biển báo tốc độ (tối thiểu, tối đa), hoặc theo quy định chung của pháp luật cho khu vực dân cư, hoặc ngoài dân cư.

- F2: Nhắc biển phía trước: nhắc nhở khi xe đi gần đến các biển báo để điều chỉnh hành vi. Ví dụ phía trước vận tốc tối đa 80km/h, hoặc lối rẽ bên trái vận tốc 60 km/h

Để làm được tính năng đó, cần liên tục xác định cung đường mà xe đang đi, qua đó so sánh vận tốc hiện tại với tốc độ giới hạn của cung đường đang đi, hoặc kiểm tra trên cung đường phía trước có các biển báo nào để báo lại cho người lái xe. Việc xác định chính xác cung đường là cần thiết, tránh nhầm lẫn cung đường phía dưới hoặc 1 đường song song vì tốc độ cho phép và biển báo có thể rất khác nhau, ví dụ Đại lộ Thăng Long, đường tốc độ cao được đi 100 km/h, tuy nhiên đường gom chỉ được đi 60km/h. Thuật toán cần chạy nhanh đẻ có thể xử lý được càng nhiều xe đồng thời càng tốt.

Input

Dữ liệu đầu vào gồm 2 loại:

- Bản tin REPORT_GPS: 1 mảng khoảng 7-8 phần tử, mỗi phần tử chứa thông tin tọa độ (lat, lng) và hướng di chuyển thu thập từ thiết bị OBD, gắn trên xe ô tô. Tần suất 8 giây báo lên 1 lần. Mỗi phần tử trong mảng kể trên, tương ứng vị trí của xe trên từng giây.

- Bản tin REPORT_VEHICLE: có thông tin tọa độ (lat, lng), vận tốc, và 1 số thông tin khác về thông số của xe. Tần suất gửi lên 5 giây/lần.

Hiện GPS có sai số trung bình khoảng 12m, một số mẫu có sai số lên tới 90m.

Thông tin về đường được lấy từ OpenStreetMap, được tiền xử lý gán biển báo giao thông thu thập ở Việt Nam. Kết quả là các cung đường đã được gán tốc độ tối đa, tốc độ tối thiểu (nếu có).

Output

Xác định CUNG ĐƯỜNG tương ứng của xe mỗi khi ghi nhận thêm bản tin (1 trong 2 loại bản tin kể trên). Sau đó đưa ra 1 trong 2 loại thông báo (nếu có): báo vượt tốc và nhắc biển phía trước.

Cách xử lý

Hiện hệ thống đã giải bài toán này theo 2 cách khác nhau:

- Cách 1: Xử lý theo batch.

Cụ thể là với mỗi thông tin tọa độ GPS mới được đưa vào, sẽ lấy **n** điểm GPS cuối cùng được ghi nhận, hoặc các điểm GPS ghi nhận trong thời gian **t** giây gần nhất. Với các điểm này, sẽ cố gắng tìm cung đương khả dĩ nhất (xác suất cao nhất) dựa theo các GPS đầu vào ghi nhận.

Việc matching từ GPS vào đường theo hường offline có nhiều opensource đã làm. Có thể kể đến như Graph Hopper (Java), OSRM (C++), Valhalla (C++). Đặc điểm chung đều là làm theo mô hình Hidden Markov Model, dựa theo các điểm quan sát được (là các GPS) tìm trạng thái thật (state - chính là các điểm trên đường). Chi tiết paper gốc tham khảo ở đây: Hidden Markov Map Matching Through Noise and Sparseness

Nhược điểm của phương pháp này là nếu chọn **n hoặc t** lớn thì có thể tăng độ chính xác khi match vào đường, tuy nhiên sẽ tăng thời gian xử lý. Tuy nhiên nếu chọn n hoặc t nhỏ sẽ ảnh hưởng tới độ chính xác nhưng giảm thời gian xử lý, 1 lí do có thể giải thích là do tập điểm nhỏ không capture được các vị trí quan trọng như lối rẽ, gây nhầm lẫn với đường song song hoặc đường phía dưới.

- Cách 2: Xứ lý online, sử dụng variable sliding window thay vì fixed sliding window như cách 1.

Chi tiết paper gốc tham khảo ở đây: Online map-matching based on Hidden Markov model for real-time traffice sensing applications

Thuật toán VCar đang sử dụng, là bản đã sửa (tối ưu 1 số cấu trúc dữ liệu và chỉnh sửa dữ liệu đầu vào cho phù hợp) từ opensource Barefoot, implement theo tư tưởng ở paper trên. Repo thực tế phần core matching online ở đây: thucdx/or-tracker

Ưu điểm: Độ chính xác về cung đường tương chạy với chạy offline. Có thể đánh đổi 1 chút độ chính xác bằng cách tinh chỉnh tham số về trạng th

Nhược điểm: Hiện VCar đã sử dụng cách này để chạy, độ chính xác tương đương khi chạy offline và cover được hầu hết trường hợp. Tuy nhiên vẫn cần nhiều cải tiến về hiệu năng để có thể phục vụ đồng thời nhiều xe hơn nữa. Benchmark chung, 1 máy cấu hình CPU tương đương AMD Ryzen 7 3700x (3.6 GHz, 8 nhân 16 luồng), có thể phục vụ được 100 xe chạy online đồng thời (mỗi GPS point cần khoảng 10ms để xử lý để update tính toán lại vị trí của xe).

Các vấn đề có thể tối ưu được

Các vấn đề đã làm:

1. Giãn các điểm cần cập nhật vị trí, thay vì 1s hoặc 2s update 1 lần thì có thể 4s hoặc 5s mới ccần update 1 lần. Hoặc có cơ chế dynamic ví dụ như nếu xe đi nhanh thì tần suất dày hơn, xe đi chậm thì thưa hơn.
2. Tinh chỉnh tham số quét bán kính để tìm tập điểm trên đường khả dĩ có thể là vị trí thực tế của điểm tọa độ được report lên. Theo thực tế đo kiểm tại Hà Nội, trung bình GPS sai số khoảng 12m, nhưng tồn tại nhiều điểm gần nhà cao tầng sai số lên tới 90m. Do vậy tham số quét bán kính này nên là d=100m. Nếu chọn d lớn sẽ có nhiều candidates thừa làm chậm thuật toán, nếu d nhỏ có thể thiếu candidate thực tế.

Các vấn đề có thể đề xuất:

1. Tìm opensource khác hoặc 1 cách tiếp cận mới.