UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Gº en Ingeniería Informática

TFG Ingeniería Informática:

GII17.0T Trayectorias Semánticas

Presentado por Hector Cogollos Adrian
en Burgos el 2 julio de 2019
Tutores D. Bruno Baruque Zanón
y D. Santiago Porras Alfonso

D. Bruno Baruque Zanón y D. Santiago Porras Alfonso, profesores del departamento Ingeniería Civil, área de Lenguajes y Sistemas Informáticos

Exponen:

Que el alumno D. Hector Cogollos Adrian, con DNI 71295969V, ha realizado el TFG Ingeniería Informática titulado: GII17.0T Trayectorias Semánticas.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, se autoriza su presentación y defensa.

En Burgos a 2 de julio de 2019

Vº. Bº del Tutor

Vº. Bº. del Tutor

D. Bruno Baruque Zanón

D. Santiago Porras Alfonso

Resumen

En este documento se exponen las características del trabajo de fin de grado “GII17.0T Trayectorias Semánticas”. El cual trata de aportar una herramienta para el análisis de datos GPS generados por dispositivos móviles. Estos datos son tomados a intervalos regulares formando rutas. El objetivo es convertirlos en un modelo que facilite su comprensión denominado Trayectorias Semánticas. También se busca analizar la eficacia de este modelo tratando de implementar sobre este, un algoritmo de predicción que sirva para predecir los movimientos de los usuarios. Y utilizaremos métodos de *clustering* para comprobar si podemos obtener características de los usuarios con este modelo. Por último, se desarrollará una pequeña aplicación web para poder probar algunos de estos algoritmos.

Descriptores

Trayectoria semántica, geoespacial, minería de datos, PostGIS, clasificador

Abstract

In this document the characteristics of the final degree work "GII17.0T Semantic Trajectories" are exposed. Which tries to provide a tool for the analysis of GPS data generated by mobile devices. These data are taken at regular intervals forming routes. The objective is to convert them into a model that facilitates their understanding called Semantic Trajectories. It also seeks to analyse the effectiveness of this model trying to implement on this, a prediction algorithm that serves to predict the movements of users. And we will use clustering methods to check if we can obtain user characteristics with this model. Finally, a small web application will be developed to test some of these algorithms.

Keywords

Semantic trajectory, geospatial, data mining, PostGIS, classifier.

Índice General

Índice de figuras

1. Introducción

En este documento se exponen las características del trabajo de fin de grado “GII17.0T Trayectorias Semánticas”. El cual trata de aportar una herramienta para el análisis de datos GPS generados por dispositivos móviles. Para esto partimos de datos muy básicos que puede capturar cualquier dispositivo con GPS que son latitud, longitud y el instante de tiempo en el que se toman. Estos datos tienen que estar almacenados en el orden en el que se toman para poder construir trayectorias.

Para llevar a cabo este proyecto se va a crear una pequeña aplicación que permita transformar los datos, en lo que en el artículo “*Semantic trajectories: Mobility data computation and annotation*[1]” denomina modelo semántico. Para llegar a este modelo primero se va a realizar unos pasos previos que consisten en limpiar los datos y en detección de paradas para crear un modelo conceptual. Con estos datos tendremos que obtener el punto de interés y las características de una geolocalización para crear el modelo semántico. Este modelo es el que utilizara para el análisis posterior.

Se van a realizar dos análisis de los datos en este proyecto con el objetivo de comprobar si el modelo semántico puede resultar de utilidad. El primero es analizar si con los datos de la aplicación podemos predecir la clase o el tipo de lugar al que se dirige un “usuario” a partir de una trayectoria. El segundo análisis es poder categorizar las rutas mediante un algoritmo de *clusterin*. Esto se hace con el objetivo de poder analizar las rutas más representativas de cada clúster para poder saber que representa una categoría del clúster.

También se va a desarrollar una interfaz web para la aplicación. El objetivo es que un usuario pueda cargar sus datos en la base de datos de la aplicación. Después queremos que el usuario pueda seleccionar aquellos datos que desea según su criterio para que sean cargados en memoria. Y una vez el usuario a cargado los datos en memoria pueda configurar los clasificadores según su criterio para analizar los datos.

2. Objetivos del proyecto

El objetivo principal que se persigue en este proyecto es poder analizar trayectorias semánticas. Para ello se va a realizar una aplicación para la cual se fijan los siguientes objetivos funcionales:

- Diseñar e implementar una base de datos en la cual almacenar los datos que posteriormente utilizara la aplicación. Esta base de datos debe ser compatible con datos geográficos.
- Desarrollar una funcionalidad que permita cargar los datos en la base de datos de la aplicación. Los datos que se cargaran en la aplicación deben formar trayectorias coherentes.
- Implementar un algoritmo que permita detectar que puntos de la trayectoria corresponden a paradas del usuario. Y determinar la localización geográfica en la que se ha parado un usuario.
- Implementar una funcionalidad que permita identificar qué puntos de interés hay en los puntos geográficos en los que se ha parado un usuario. Y las características de estos puntos de interés en los que se ha parado el usuario.
- Implementar un clasificador que permita predecir la clase o el tipo de la siguiente parada en una trayectoria. De este clasificador tenemos que ser capaces de evaluar los resultados del entrenamiento.
- Implementar un algoritmo que nos permita agrupar las trayectorias por similitud con el fin de poder analizar los grupos y determinar sus características. En definitiva, que nos permita obtener información de los datos.
- Implementar una interfaz web para que un usuario pueda utilizar la aplicación.

Los objetivos técnicos son los siguientes:

- Se persigue familiarizarse con el desarrollo de aplicaciones Web, más concretamente con el Framework para el lenguaje Python, Flask.
- Familiarizarse con el uso de técnicas de minería de datos y su posterior representación de los resultados.
- Iniciación a la planificación de proyectos mediante metodologías agiles más concretamente Scrum, pero adaptado a las particularidades del proyecto. Para ayudarme en la planificación y la utilización de esta metodología me ayudare de GitHub con el plugin de ZenHub que nos facilita la planificación de tareas. Y como cliente Git utilizo GitKraken que permite gestionar las versiones de forma fácil.
- Por último, también se persigue aplicar los conceptos y experiencia adquiridas durante el grado. Para ello utilizaremos los conocimientos en distintas áreas del grado. Algunas de las más relevantes son la gestión de sistemas, gestión de bases de datos, técnicas de desarrollo software y minería de datos.

3. Conceptos teóricos

A continuación, se desarrollan los conceptos teóricos necesarios para la correcta compresión del proyecto.

Trayectoria

Una trayectoria es una serie de puntos formados por latitud, longitud e instante de tiempo. Estos puntos se obtienen de usuarios desplazándose entre distintas ubicaciones. Los puntos se obtienen en intervalos cortos de tiempo mediante un dispositivo que disponga de GPS. Con esto obtenemos una serie de coordenadas ordenadas de tal forma que se puede ver como se ha ido desplazando el usuario. Por usuario no solo se puede entender una persona física, también se puede tratar de un vehículo animal o en general cualquier cosa que se desplace entre puntos dentro del planeta.

En este proyecto vamos a distinguir entre tres tipos de trayectorias que parten de esta definición y en las cuales van “evolucionando” de las anteriores para poder obtener información.

Trayectoria en Bruto

Estas son esencialmente Trayectorias como las definidas anteriormente pero que contienen errores o pueden ser inconexas. Esto puede deberse a fallos de sincronización del GPS, una ausencia de regularidad en la toma de los datos o escasez de información.

Dado que en este proyecto no se basa en la toma de datos no vamos a entran en los detalles concretos que pueden causar estos fallos, solo en cómo abordarlos.

Estas trayectorias en bruto para que puedan ser interpretadas correctamente primero se deben eliminar los fallos mencionados anteriormente. Esto se hace troceándolas en trayectorias coherentes y desechando aquellas cadenas de puntos que sean demasiado cortas o incoherentes. La razón por la cual desechamos las que son demasiado cortas es porque no puede aportar información.

Ilustración 1Trayectoria en bruto

Trayectorias Conceptuales

Una trayectoria conceptual se obtiene mediante una transformación de una trayectoria en bruto que han sido procesadas para eliminar los fallos. Al hacer esta transformación lo que obtenemos es otra vez una serie de puntos, pero ahora estos representan paradas. En este caso las paradas son aquellos puntos en los que estimamos que se ha detenido el usuario. Estos están formados por latitud, longitud, tiempo de inicio de la parada y el tiempo de fin de la parada.

En las trayectorias hasta ahora lo que teníamos era una gran cantidad de puntos en los que había estado el usuario y cuando. Ahora lo que tenemos es un numero mucho más reducido de puntos que representan en qué punto geográfico ha estado parado y en que intervalos de tiempo. También sabemos en qué orden se ha movido entre estos puntos.

Ilustración 2 Trayectoria Conceptual

Trayectorias Semánticas

Una trayectoria semántica se construye a partir de una trayectoria conceptual. Lo que se busca en este nivel es dar un significado a las paradas que teníamos en las trayectorias conceptuales. Para ello lo que hacemos es obtener el lugar donde se ha detenido, pero en vez de en coordenadas semánticamente. Planteando lo de otra manera lo que queremos saber es que hay en esas coordenadas físicamente.

Por ejemplo, si ha estado en un bar parado queremos saber que bar, de que calle, de que ciudad, de que país, etc. Todos los datos que podamos obtener de ese sitio donde ha estado parado el usuario.

También queremos seguir manteniendo el orden y agruparemos aquellas paradas que suceden dentro de la misma localización semántica. Con esto lo que buscamos obtener es una ruta con los lugares en los que se ha ido deteniendo el usuario.

Ilustración 3 Trayectoria Semántica

La ilustración anterior trata de representar una trayectoria semántica que no necesariamente tiene que tener las mismas paradas que la conceptual. En este punto lo que tengo es la información de los lugares a los que se mueve un usuario y el orden en el que lo hace.

Minería de datos

La minería de datos[2] lo que persigue es la extracción de conocimiento o información a partir de grandes conjuntos de datos. Para poder obtener conocimiento o información hay que realizar una serie de operaciones tales como selección de los datos, filtrado de datos, procesado de los datos y análisis de los resultados obtenidos.

En minería de datos tenemos dos modelos a la hora de extraer información los modelos predictivos y los modelos descriptivos. En el primero se busca predecir valores futuros para nuevos datos como sucede en los clasificadores. Y en los modelos descriptivos se busca identificar patrones y resumir los datos como es el caso del *clustering*.

Clustering

El *clustering* es una técnica que se utiliza cuando no tenemos las instancias clasificadas por clases, pero queremos agruparlas en grupos naturales. Los grupos que se forman con las técnicas de *clustering* presumiblemente reflejaran un comportamiento o características comunes a las instancias que la forman. Analizando

estos grupos podemos obtener una descripción que describa a las instancias que conforman el grupo.

Distancias

Los algoritmos de *clustering* generalmente tratan de minimizar distancias para determinar su similitud. Para calcular la distancia entre dos instancias se necesita una función que la calcule, a continuación, podemos ver las más comunes.

- Distancia Euclídea: esta forma de medir la distancia lo que hace es calcular la distancia entre los dos puntos con una recta.
- Distancia de Manhattan: mide la distancia entre dos puntos como si el espacio estuviese dividido en cuadriculas te tal modo que mide la distancia en zigzag.
- Distancia de Chebychev: es la distancia máxima que hay entre dos puntos en alguna de las dimensiones.
- Distancia de Mahalanobis: esta distancia no asume que cada atributo es independiente lo que la hace más robusta, pero con peor escalabilidad. Para ello utiliza la matriz de covarianzas.

Hay más funciones para medir las distancias entre instancias, en función del problema se debe elegir una u otra. Incluso se puede definir una función para medir distancias que se especifica al problema y no de propósito general.

KMeans

Es un algoritmo bastante simple de *clustering* el cual es de propósito general y que cosiste en, a partir de n puntos que llamaremos clústeres los cuales situaremos en el espacio de datos de forma aleatoria o por otro método. Ahora los datos pertenecerán a uno de estos clústeres y será al que más cerca se encuentre. Los puntos de los clústeres se moverán en cada interacción al centro de todos los puntos que pertenecían a este. Esto se hace hasta que deje de variar la pertenencia de los datos a los clústeres.

PCA

PCA o análisis del componente principal es una técnica utilizada para reducir la dimensionalidad de los datos. Esta técnica transforma los datos en otros con una dimensionalidad menor, los devuelve en el orden de relevancia y son independientes entre sí.

Clasificador

Un clasificador es un algoritmo que permite clasificar instancias en una clase ya conocida. Los clasificadores son algoritmos de aprendizaje supervisado y requiere tener instancias de entrenamiento en las que ya conozcamos la clase a predecir. Para utilizar un algoritmo de este tipo, primero se le debe pasar los datos de entrenamiento mencionados. Y una vez tenemos el algoritmo entrenado se puede pasar a clasificar nuevas instancias. Habitualmente lo que se hace es reservar algunos de los datos ya clasificados para validar el clasificador y estimar su tasa de aciertos.

Arboles de decisión

Los arboles son un tipo concreto de clasificador, como su propio nombre indica tiene forma de árboles. Estos árboles en los nodos tenemos condiciones que en función del resultado nos llevaran por una u otra rama del árbol. Esta operación se repite hasta llegar a las hojas donde tenemos la clase que queremos predecir.

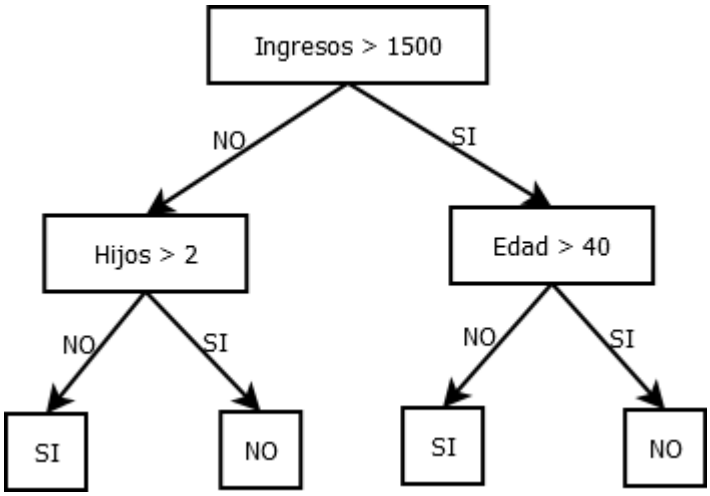

Ilustración 4 Árbol de Decisión

Validación Cruzada

La validación cruzada es un método de validación que permite reducir la dependencia de los datos. Consiste en dividir los datos de entrenamiento en n grupos, de estos grupos n-1 se utilizará para entrenar el clasificador. El grupo que queda es el que utilizaremos para validar el clasificador. Esto hay que hacerlo rotando de tal forma que al final todos los grupos se han utilizado una vez para validar lo que supone n iteraciones. Como resultado de la validación se devuelve la media aritmética de los experimentos realizados.

Base de datos Geoespacial

Las bases de datos geoespaciales[3] son bases de datos normales pero que incluyen tres características que mejoran su rendimiento y usabilidad. La primera es que incorporan objetos especiales como puntos, líneas y polígonos. La segunda es que incluye índices multidimensionales propios para mejorar el procesado de las operaciones espaciales. En tercer lugar, implementa funciones propias de sistemas geoespaciales. Estas características nos permiten operar de forma más eficiente y cómoda con este tipo de datos.

4. Técnicas y herramientas

En este apartado se exponen tanto herramientas utilizadas para el desarrollo software como técnicas de diseño adoptadas.

Herramientas

Python

Este ha sido el lenguaje de programación escogido para desarrollar este proyecto software.

Las ventajas que proporciona este lenguaje frente a otros es la cantidad y calidad de librerías de las que dispone, tanto para manejar datos geoespaciales, como para el procesamiento de datos. Dado que el proyecto tiene como ejes centrales estos dos puntos se optó por esta alternativa.

Como desventaja tenía la gran complicación que supone implementar técnicas de ingeniería del software en este lenguaje.

Alternativas

Java

Java se consideró como posible alternativa, pero se descartó debido a la complejidad y escasa información de las librerías que había, en comparación con las de Python.

Open Street Map

O OSM[4] se trata de un proyecto colaborativo para crear mapas. Estos mapas son creados y actualizados usando información geográfica obtenida por sus colaboradores. Que es almacenada en una base de datos. Esta base de datos se distribuye mediante una licencia abierta de base de datos.

Estos mapas les utilizamos en el proyecto para poder saber que hay en el mundo real en esas paradas que tenemos registradas. Y de esta forma obtener información de las paradas y no solo puntos geográficos.

Alternativas

Google Maps

Se descarto por ser una alternativa distribuida mediante una licencia propietaria y tiene costes económicos. Y dado que esta aplicación está bajo una licencia GPL3 completamente gratuita, es un coste que no se puede asumir.

GeoPandas[5]

Es una librería de Python que extiende de la librería Pandas, permitiendo realizar todas las operaciones que permite Pandas y también aporta una serie de operaciones geoespaciales. En el proyecto es utilizada para hacer operaciones

matemáticas sobre los datos, transformaciones en los sistemas de coordenadas y cargar datos. Nos permite cargar datos geoespaciales directamente de la base de datos.

Pandas[6]

Es una librería de código abierto que sirve para realizar análisis de datos permitiendo de forma fácil hacer operaciones sobre los datos.

En el proyecto principalmente utilizamos pandas a través de GeoPandas, pero hay operaciones de las cuales no dispone GeoPandas como es la carga de datos desde un CSV. Y esta funcionalidad es necesaria para introducir nuevas rutas.

PostgreSQL[7]

Es un motor de bases de datos de código abierto. Este motor de bases de datos cuanta con una extensión que es PostGIS que permite trabajar con datos geoespaciales. La escogimos porque el resto de las bases de datos que pueden trabajar con estos tipos de datos son de pago. A demás es un motor de bases de datos estable y del que hay gran cantidad de información.

En el proyecto la utilizamos para guardar la base de datos de Nominatim, la de la propia aplicación y otra de ost2psql. Todas ellas necesitan del complemento de PostGIS.

PostGIS[8]

Se trata de una extensión de PostgreSQL la cual facilita el trabajo con datos geoespaciales. Añade nuevos tipos de datos geoespaciales, mejora la búsqueda de estos y añade funciones para trabajar con este tipo de datos.

Se utiliza para poder almacenar los datos espaciales de las tres bases de datos que utilizamos. También la utiliza GeoPandas para cargar datos de la base de datos en la aplicación.

Nominatim[9]

Es una Herramienta que nos permite buscar en una base de datos completa de OSM. Tiene una modalidad de búsqueda que es “Reverse” con la que podemos buscar elementos de OSM por id, o por coordenadas. Esta búsqueda inversa por coordenadas nos permite buscar por capas y podemos escoger como queremos que nos devuelva la información. Esto va desde el formato al idioma o el nivel de detalles.

En este proyecto la utilizamos para buscar coordenadas y obtener el punto más cercano de las paradas que tenemos en la aplicación. A demás también la usamos posteriormente para obtener los detalles de una para de la que ya tenemos guardada la id de OSM.

Esta herramienta tiene una API a la cual se pueden hacer las peticiones, pero debido a la limitación de una petición por segundo hemos tenido que optar por instalarla.

Osm2pgsql[10]

Esta es una herramienta que nos permite cargar los XML de OSM en una base de datos de PostgreSQL. Para atízala necesitamos tener una base de datos creada con la extensión de PostGIS. Esta Herramienta también la utiliza Nominatim para crear su base de datos. Osm2pgsql por defecto crea una base de datos simplificada con los datos de OSM, pero permite crear bases de datos con los datos completos como hace Nominatim mediante sus opciones de configuración.

Flask[11]

Es un *framework* de Python para diseñar páginas web y aplicaciones web. Flask cuanta con el *framework* básico para crear las aplicaciones y además cuenta con multitud de extensiones para añadir funcionalidades. Una de sus ventajas es que solo necesitas tener Python instalado ya que no necesita de ninguna aplicación de servidor adicional lo que facilita su despliegue.

En el proyecto se utiliza para crear la aplicación web para que un usuario pueda interactuar con la aplicación.

Bootstrap

Es una herramienta de código abierto para desarrollo web en HTML, CSS y JS. Permite agregar funcionalidades y dar diseño a una página web de forma sencilla. En este proyecto utilizamos fundamentalmente la parte de estilos. Que permite agregar estilos unificados a la aplicación simplemente con añadir clases a las etiquetas HTML.

Técnicas

MVC

En la aplicación se utiliza el patrón de diseño Modelo Vista Controlador el cual ayuda a separar las distintas partes del proyecto. El modelo encapsula aquellas partes del programa que forman la estructura de los datos. La vista encapsula las partes que son visibles para el usuario y con las que interactúa, El controlador contiene la lógica del programa e interactúa con las dos anteriores. También en este proyecto se añade el apartado de datos ya que interactuamos con una base de datos y una API que nos suministra también información en tiempo de ejecución.

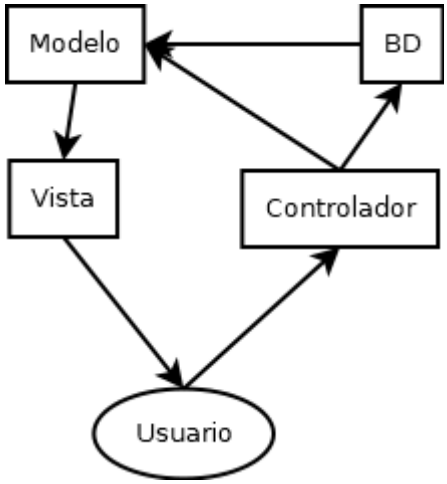

Ilustración 5 MVC Comunicación simplificada entre componentes

Singleton

Se trata de un patrón de diseño creacional cuya funcionalidad es evitar que se creen instancias de un objeto de forma incontrolada. Este patrón lo que hace es evitar que se puedan crear instancias nuevas evitando que se pueda usar su constructor directamente. Y te obliga a llama a una función que te devuelve la instancia del objeto si existe y si no la crea y te la devuelve.

En el proyecto se usa en dos ocasiones para almacenar información global de la aplicación que se requiere en distintos momentos evitando la necesidad de irse pasando la información de uno a otro.

5. Aspectos relevantes del desarrollo del proyecto

En este apartado se habla de algunas de las partes más relevantes del proyecto y de las dificultades que han aparecido. Esta separado por puntos que se encuentran en el mismo orden en el que se desarrolló el proyecto.

Obtención de datos[12]

Este punto es fundamental y condiciona el proyecto debido a que se centra en el tratamiento de datos. La idea es partir de unos datos lo más genéricos posibles para no condicionar el proyecto a una aplicabilidad especifica. Se ha optado por unos datos que han sido obtenidos por diferentes aplicaciones móviles capaces de tomar ubicaciones geoespaciales en tiempos regulares. Estos datos son de voluntarios de origen chino los cuales grabaron sus rutas las cuales carecen de contesto.

Se barajo la utilización de otros datos los cuales si tenían un contexto como taxis, autobuses urbanos, deportistas. El problema de estos es que son demasiado dirigidos y se enfocaría en resolver un problema propio de ese contexto. Por el contrario, con datos más generales podemos centrarnos en obtener información independientemente del contexto siendo más fácil de generalizar a otros problemas.

El primer problema que encontramos en estos datos es que tiene una gran cantidad de errores y un error de precisión bastante amplio. El segundo fue que los datos eran chinos y estaban centrados en ciudades chinas especialmente Pekín. Esto tiene la problemática de que china a pesar de su gran tamaño y población la base de datos de OSM es más pequeña que la de España. Por lo que tiene bastantes vacíos en el mapa en los cuales no hay apenas información.

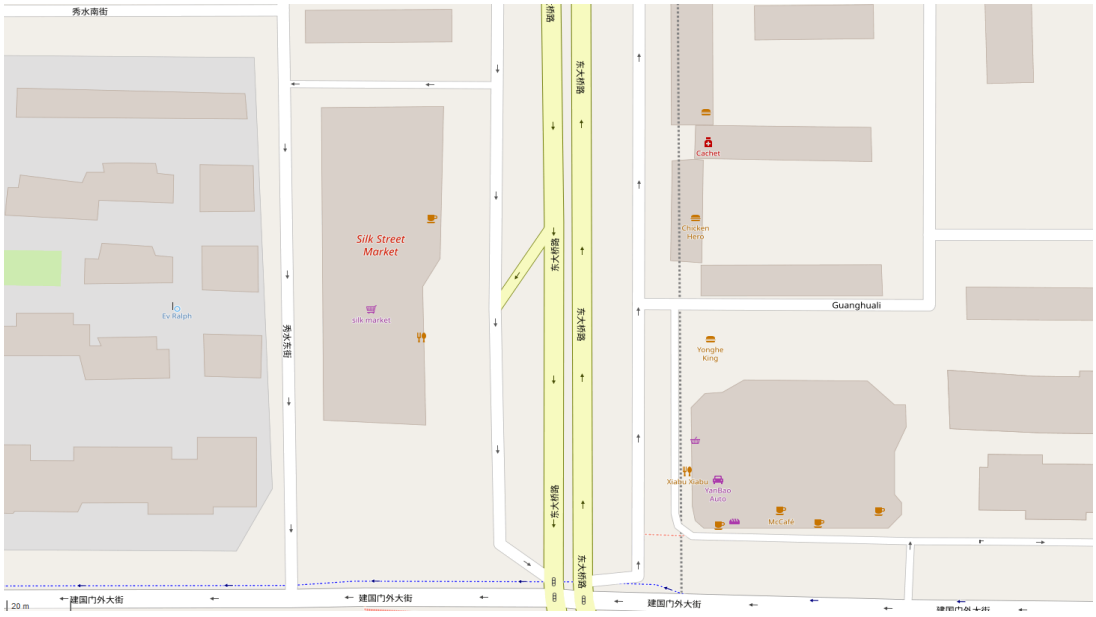

Ilustración 6 China Mercado de la seda

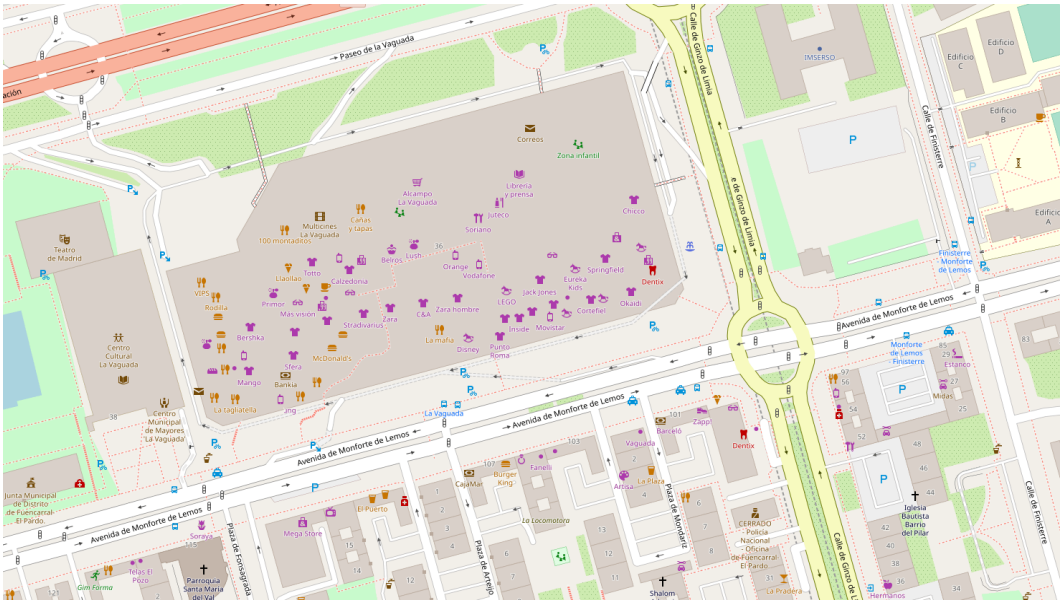

Ilustración 7 La vaguada Madrid

La primera ilustración es del mercado de la seda en Pekín una de las principales zonas comerciales y la segunda es la vaguada en Madrid. Como se puede observar la diferencia de puntos de interés es abismal y muestra claramente la escasez de datos que hay de china. Esto no solo pasa en OSM en Google Maps pasa lo mismo.

Detección de paradas

Ese es el principal punto en el que se basa la aplicación y es detectar cuando está parado un usuario dentro de una trayectoria. A qui tenemos por un lado la problemática de que los datos no están etiquetados por lo que realmente no sabemos cuándo está parado y cuando no. Para paliar un poco esta situación tome datos con la aplicación para Android A-GPS Tracker de mis propios movimientos. De esta forma, aunque seguían sin estar etiquetados si podía saber dónde estuve parado y donde no, si lo representaba en un mapa. Esto fue muy útil para poder contrastar si se estaban detectando las paradas bien o no.

Otra problemática es que los datos son series temporales, y no datos independientes unos de otros por lo que un algoritmo de *clustering* común tampoco era útil. A esto hay que sumarle la gran cantidad de datos la cual no se podía almacenar completamente en memoria.

Lo primero que se hizo para poder trabajar con los datos fue determinar a que se iba a considerar una trayectoria. Para que una serie de puntos se considerase que sigue siendo una trayectoria tiene que cumplir con un mínimo de puntos y de tiempo. Las series de puntos se dividen cunado hay vacíos de datos en el tiempo y un desplazamiento notable. Esto sirve para eliminar errores y problemas futuros a la hora de determinar si está parado.

Para determinar donde se encuentra parado un usuario primero probamos una solución simple que consistía en dos reglas. La primera es que se considera parada si la velocidad a la que se mueve entre dos puntos es inferior a 0.6 m/s (2.15 km/h). La

segunda es que si a pesar de la velocidad no nos desplazamos más de 30 metros con respecto a una media móvil de las ubicaciones comprendidas en el intervalo de los tres minutos previos y un minuto previo. Esta aproximación detectaba las paradas de forma errónea cuando había curvas pronunciadas en las trayectorias.

La segunda solución que implementamos para detectar paradas la extrajimos del artículo "*Identifying stops from mobile phone location data by introducing uncertain segments*[13]". Este articulo propone coger intervalos de tiempo y calcular que distancia se recorre entre el último punto y todos los anteriores del intervalo de tiempo. En este algoritmo hay tres estados, parado, en movimiento y desconocido. Estos dependen de la distancia que se recorra en el intervalo de tiempo. En este caso el estado de desconocido se asume como que está en movimiento debido a que la solución que propone el documento es dependiente del contexto. Este algoritmo dio muy buenos resultados cuando utilizamos las rutas que grabe de mis movimientos.

Detección de puntos de interés

El objetivo de este punto es que a partir de las paradas anteriores saber qué puntos de interés hay entorno a esas ubicaciones. A pesar de que teníamos una base de datos con datos de OSM nos encontramos con que era difícil determinar en qué punto se encuentra un usuario. Para esto utilizamos una herramienta de software libre llamada Nominatim esta herramienta nos permite que a parir de la latitud y la longitud obtener el punto con mayor relación entre cercanía e importancia.

Ilustración 8 Captura de Nominatim Reverse

En la ilustración anterior podemos ver un ejemplo de lo que hace Nominatim. En rojo una coordenada y en azul lo que detecta que hay en esa coordenada.

Ya se mencionó anteriormente que la base de datos de china es muy pequeña y tiene información. Lo que deja muchas zonas sin puntos de interés y hace que muchas paradas estén en calles donde aparentemente no hay nada. Hay otra problemática y es que los requisitos del servidor que aloja Nominatim aumentan considerablemente en función de los países que agregamos a la aplicación. Por esta razón solo está cargada la información de china en el servidor de Nominatim debido a que comparte recursos

con resto de los componentes del proyecto y este servidor es muy limitado. Debido a esto se pierde información de algunos usuarios que viajan por el mundo.

Predicción de próximos puntos de interés

En este punto se busca obtener la clase de la próxima parada que hará un usuario a partir de una ruta. La clase determina si un punto de interés en un lugar turístico, una tienda, un lugar de ocio, etc. Para obtener un predictor que sea capaz de predecir esto utilizamos un algoritmo de aprendizaje supervisado similar a un árbol. Este algoritmo se ha extraído de *"Semantic trajectory mining for location prediction*[14]*"* en el cual se platea crear un árbol de clasificación, pero diferente. Lo que se hace es entrenar este con la secuencia y subsecuencias de clases que forman las trayectorias de entrenamiento. Con esto obtenemos un árbol al cual en cada nodo se le da un peso en función de su soporte (valor entre 0 y 1 que determina en cuantas rutas aparece). Con este árbol no hay que baja hasta la hoja, sino que con la ruta que queremos predecir se recorren todos los caminos y se evalúa su soporte en función del número de nodos con los que coincida. Y nos quedamos con el que nos da la cifra más alta.

Los resultados de este algoritmo con los datos de los que disponemos no se pueden extrapolar a otros individuos de forma indiscriminada debido a las diferencias que existen entre ellos. Solo es capaz de predecir para el individuo de entrenamiento u otro individuo similar nuevas rutas.

Agrupación de usuarios

Aunque por falta de tiempo este apartado no está dentro de la aplicación web sí que están desarrolladas algunas de sus partes en controlador de la aplicación. A demás se han hecho pruebas para estudiar su viabilidad de desarrollo y ha servido para ver algunos patrones en los datos ya no de comportamiento de los usuarios sino también de la calidad de los datos.

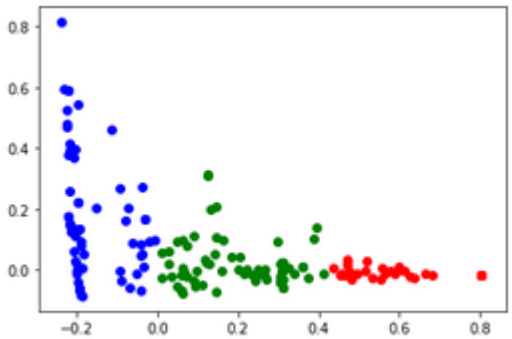

Ilustración 9 Representación en dos dimensiones mediante PCA de la clasificación de KMeans

Para aplicar técnicas de clústeres a los datos se ha utilizado una representación mediante grafos dirigidos de todas las rutas de un usuario. De este grafo sacamos la matriz de adyacencia que es la que se utilizara en los algoritmos de *clustering*. Esta matriz se representa en forma de lista y nos quedamos con las columnas que tiene información. A este resultado se le ha aplicado KMeans con tres clústeres que se ha decidido a partir de la representación en dos dimensiones de los datos mediante PCA.

Podemos observar que es el eje de las “x” el que define mayoritariamente a que clúster pertenece. Al obtener los centros de cada clúster hemos visto que la componente principal es el porcentaje de veces que un usuario va de una vía (carreteras o calles) a otra. En el clúster azul representa el 1.146% de los enlaces, en el verde es el 38.892% de los enlaces y en el rojo es el 86.098% de los enlaces. Teniendo en cuenta que ir de una vía a otra no nos aporta información relevante, se puede concluir a falta de más pruebas que los usuarios de estudio más interesantes son aquellos que pertenecen al grupo azul.

Interfaz Web

El principal problema con la interfaz Web es mi desconocimiento del Framework Flask y de Java Script, así como del hecho de que la aplicación no es una página Web. La mayor dificulta ha sido que el usuario pueda cagar nuevos datos en la aplicación debido a que el navegador deja de atender al servidor si no recibe una respuesta rápida por parte del servidor. Para solucionar esto se necesita el uso de procesos o hilos esto también pasa con el clasificador. También ha anido dificultades con las comunicaciones entre el servidor y el cliente. Debido a que las páginas web no están diseñadas para que sea el servidor el que tome la iniciativa en el paso de mensajes. Por el contrario, requiere de alguna acción por parte del usuario para desencadenar una serie de acciones en el servidor.

6. Trabajos relacionados

En este apartado se exponen proyectos similares con mayor o menor relación con respecto al de este trabajo de fin de grado.

Carto[15]

Es una herramienta SaaS (software como servicio con computación en la nube) que engloba una gran cantidad de herramientas para el tratamiento de datos geoespaciales. Tiene varios apartados como carga de datos de múltiples orígenes, el enriquecimiento de datos, análisis de datos visualización de los datos e integración en aplicaciones. La herramienta es completamente de pago que en su versión profesional tiene un coste mensual de 199$ con la suscripción profesional anual la cual tiene bastantes limitaciones en cuanto a capacidad. Y tiene otra versión empresarial sin restricciones, pero con costes por uso.

En general es una aplicación muy potente que te permite hacer todo tipo de análisis y aplicaciones basadas en datos geoespaciales con resultados muy profesionales, pero con un coste importante.

Power BI[16]

Es una herramienta de Microsoft para visualización de datos y análisis general. Esta herramienta permite utilizar datos geoespaciales para sus análisis y trata de poner en valor su facilidad de uso y su rápida visualización para la toma de decisiones en empresas. Es una aplicación de pago cuyo precio varía entre los 8,40€ al mes la aplicación personal a los 4.212€ que cuesta la versión completa con acceso a cualquier usuario y recursos de almacenamiento y calculo en la nube.

Arcgis[17]

Es una herramienta de análisis de datos para la toma de decisiones apoyándose en datos geoespaciales que permitan agrupar por áreas. Por ejemplo, posibles zonas o poblaciones donde captan nuevos clientes. Incluye una gran cantidad de herramientas dependiendo del sector empresarial de tu empresa.

7. Conclusiones y líneas de trabajo futuras

En este apartado se exponen las conclusiones generales de los resultados del proyecto y posibles líneas de trabajo futuras.

Conclusiones

En la fase de detección de paradas consideramos que los resultados son bastante buenos, aunque se pueden mejorar analizando aquellos tramos dudosos. También habría que valorar la posibilidad de que las rutas sen de un día y no en función de los cortes temporales, aunque esto podría suponer algunos cambios en el algoritmo.

En la fase en la que detectamos que puntos son más cercanos creemos que podríamos obtener mejores resultados en zonas con mayor número de datos. Ya que como se ha visto en el apartado de *clustering* nos encontramos con que hay muchos usuarios cuyas rutas contiene una gran cantidad de enlaces entre calles. Habría que investigar más afondo esto, pero podría deberse a falta de información de las zonas en las que viven.

También creemos que tener trayectorias más regulares como sugería con la idea de partirlas por día podría mejorarla coherencia de las rutas y con ello su análisis tanto en *clustering* como en predicciones.

En las predicciones creemos que hay que ajustar las categorías y tipos debido a que algunas son demasiado generales como "*shop*" o "*amenity*" en el caso de las categorías y en los tipos hay demasiadas las cuales se pierden con el soporte o son demasiado específicas como para que una combinación de ellas pueda aparecer más de una vez.

líneas de trabajo futuras

De este proyecto se pueden sacar bastantes líneas de trabajo futuras debido a que todavía queda por explorar muchas opciones de explotación de estos datos. Partiendo de la base de este proyecto son las trayectorias semánticas se pueden plantear las siguientes líneas de trabajo:

- Terminar de implementar el módulo de *clustering* para analizar trayectorias en la aplicación.
- Mejorar el algoritmo de detención de paradas añadiendo reglas para determinar que son con mayor precisión los puntos con estado desconocido.
- Eliminar la dependencia de Nominatim para mejorar la eficiencia de búsqueda sin tener que hacer peticiones HTTP.
- Implementar una interfaz para la aplicación que permita una mejor interacción del usuario con la aplicación.

- Mejorar la seguridad de las comunicaciones entre el cliente y el servidor de la aplicación y cifrar los ficheros con la configuración de las bases de datos.
- Permitir que la aplicación pueda ser utilizada por múltiples usuarios al mismo tiempo.
- Añadir nuevos orígenes y formatos de datos para introducir en la aplicación.
- Implementar nuevas funcionalidades basadas en las trayectorias semánticas tales como sistemas de recomendación.

Bibliografía

[1] Zhixian Yan et al., «Semantic Trajectories: Mobility Data Computation and Annotation», *ACM Transactions on Intelligent Systems and Technology* 4, n.º 3 (1 de junio de 2013): 1, https://doi.org/10.1145/2483669.2483682.

[2] I. H. Witten et al., *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann Publishers Inc, 2016), https://www.dawsonera.com:443/abstract/9780128043578.

[3] «2. Introduction — Introduction to PostGIS», accedido 20 de junio de 2019, https://postgis.net/workshops/postgis-intro/introduction.html.

[4] «ES:Acerca de OpenStreetMap - OpenStreetMap Wiki», accedido 30 de junio de 2019, https://wiki.openstreetmap.org/wiki/ES:Acerca_de_OpenStreetMap.

[5] «GeoPandas 0.4.0 — GeoPandas 0.4.0 documentation», accedido 10 de abril de 2019, http://geopandas.org/.

[6] «Python Data Analysis Library — pandas: Python Data Analysis Library», accedido 10 de abril de 2019, https://pandas.pydata.org/.

[7] «PostgreSQL: About», accedido 30 de junio de 2019, https://www.postgresql.org/about/.

[8] «2. Introduction — Introduction to PostGIS».

[9] «Nominatim 3.3.0», accedido 18 de mayo de 2019, http://nominatim.org/release-docs/latest/.

[10] «Osm2pgsql - OpenStreetMap Wiki», 2, accedido 18 de mayo de 2019, https://wiki.openstreetmap.org/wiki/Osm2pgsql.

[11] «Welcome to Flask — Flask 1.0.2 documentation», accedido 30 de junio de 2019, http://flask.pocoo.org/docs/1.0/.

[12] «GeoLife GPS Trajectories», Microsoft Download Center, accedido 26 de marzo de 2019, https://www.microsoft.com/en-us/download/details.aspx?id=52367.

[13] Zhiyuan Zhao et al., «Identifying stops from mobile phone location data by introducing uncertain segments», *Transactions in GIS* 22, n.º 4 (1 de agosto de 2018): 958-74, https://doi.org/10.1111/tgis.12332.

[14] Josh Jia-Ching Ying et al., «Semantic Trajectory Mining for Location Prediction», en *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '11* (the 19th ACM SIGSPATIAL International Conference, Chicago, Illinois: ACM Press, 2011), 34, https://doi.org/10.1145/2093973.2093980.

[15] CARTO, «Data Ingestion & Management — CARTO», accedido 30 de junio de 2019, https://carto.com/platform/data-ingestion-management/.

[16] «¿Qué es Power BI? | Microsoft Power BI», accedido 30 de junio de 2019, https://powerbi.microsoft.com/es-es/what-is-power-bi/.

[17] «ArcGIS», *Esri España* (blog), accedido 30 de junio de 2019, https://www.esri.es/arcgis/.