ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

"Análisis y Detección de Puntos de Interés Casa y Trabajo en Puntos GPS"

Trabajo de Titulación Previo a la Obtención del Título de:

MAGISTER EN CIENCIAS COMPUTACIONALES

Autor:

KARINA BEATRIZ JIMENES VARGAS

Tutor:

PhD. DANIEL ERICK OCHOA DONOSO

Co-tutor:

Mgtr. ÁNGEL JAVIER LÓPEZ AGUIRRE

Guayaquil – Ecuador

2018

Dedicatoria

A mi familia y amigos por la ayuda, apoyo y colaboración que siempre me brindan.

Karina

Tribunal de graduación

PhD. Carmen K. Vaca Ruíz **TITULAR**	PhD. Mónica K. Villavicencio Cabezas **ALTERNO**
PhD. Daniel E. Ochoa Donoso **TUTOR**	Mgtr. Ángel J. López Aguirre **CO-TUTOR**

Declaración Expresa

"La responsabilidad del contenido de este Informe de Proyecto, me corresponde exclusivamente; y el patrimonio intelectual del mismo, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

Art. 12 del Reglamento de Graduación

_____

Karina Beatríz Jimenes Vargas

Resumen

La detección de puntos de puntos de interés (POIs) contribuye en el planteamiento de soluciones a problemas de movilidad a través del análisis de actividades y viajes cotidianos de las personas usando datos GPS; sin embargo su detección a partir de puntos GPS es un problema difícil de resolver, debido a la imprecisión de los sensores GPS. El propósito de esta investigación es evaluar algoritmos basados en densidad, distancia-tiempo y velocidad que permitan la identificación del número mínimo de viajes que se requieren para detectar correctamente puntos de ubicación (PUs) que correspondan a POIs casa y trabajo. Considerando la información geoespacial, temporal y de viajes de entrada/salida se caracterizó los PUs para entrenar un clasificador con los algoritmos de árboles de decisión, bosques aleatorios y máquinas de soporte. Los mejores resultados se obtuvieron con el algoritmo basado en velocidad ya que con un promedio de tres viajes se pudo detectar los PUs hasta en un 90% de exactitud. Mientras que, el mejor clasificador fue entrenado con el algoritmo de bosques aleatorios el cual permitió clasificar un PU en un POI hasta con una exactitud del 86.6%.

Palabras claves: *puntos de interés, viajes, casa, trabajo, puntos GPS*

Abstract

The detection of points of interest (POIs) contributes to solve mobility problems through the analysis of everyday activities and trips of people using GPS data; however, the detection of POIs from GPS points is a difficult task to solve due to the inaccuracy of GPS sensors. The purpose of this research is to evaluate algorithms based on density, time-distance and speed that allow to identify the minimum number of trips required to correctly detect location points (LPs) which corresponds to home and work POIs. LPs were characterized by considering geospatial, temporal information and arrival/departure trips in order to train a classifier with a decision tree, random forests and a support vector machine algorithms. The best results were obtained with the speed-based algorithm which could detect LPs with an average of three trips and with an accuracy of up to 90%. Meanwhile, the best classifier was the random forests which allowed classification of an LP in a POI with up to 86.6% of accuracy.

Key words: *point of interest, trips, home, work, GPS point*

Índice General

Dedicatoria ii
Tribunal de graduación iii
Declaración Expresa iv
Resumen v
Abstract vi
Índice General vii
Abreviaturas ix
Índice de Figuras x
Índice de Tablas xi
CAPÍTULO 1 12
1. Introducción 12
CAPÍTULO 2 15
2. Marco teórico 15
2.1. Definiciones 15
2.2. Trabajos relacionados 17
2.3. Propósito de la investigación 20
2.4. Metodología 21
CAPÍTULO 3 23
3. Pre-procesamiento 23
3.1. Datasets 24
3.1.1. Dataset I: Viajes 24
3.1.2. Dataset II: Puntos de Interés 27
3.2. Limpieza de datos 28
3.3. Selección de muestras para experimentos 30
3.4. Validación de viajes 31
CAPÍTULO 4 36

4. Detección de Puntos de Ubicación 36
4.1. Metodología 36
4.1.1. Evaluación 41
4.2. Experimentos 42
4.2.1. Estimación de Parámetros 42
4.2.2. Selección del Número de Viajes Óptimo. 48
CAPÍTULO 5 53
5. Clasificación de Puntos de Ubicación 53
5.1. Metodología 53
5.1.1. Caracterización de PUs 53
5.1.2. Preparación de dataset de PUs 60
5.1.3. Clasificación 61
5.1.4. Evaluación 63
5.2. Experimento 65
5.2.1. Árbol de decisión 67
5.2.2. Bosques aleatorios 70
5.2.3. Máquinas de soporte 73
Conclusiones y trabajos futuros 77
ANEXOS 79
Anexo I: *Detalle de data sets de sujetos más activos luego de validación de viajes* 79
BIBLIOGRAFÍA 80

Abreviaturas

POI	*Point of Interest* (Puntos de interés en español)
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise* (Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido en español).
PU	Puntos de Ubicación
TPR	*True Positive Rate* (Tasa de verdaderos positivos en español)
GPS	*Global Positioning System* (Sistema de posicionamiento global en español)
RFID	*Radio Frecuency Identification* (Identificación de frecuencia de radio en español)
RF	*Random Forest*s (bosques aleatorios en español)
SVM	*Support Vector Machines* (Máquinas de soporte vectorial en español)

Índice de Figuras

Figura 2.1.- Viaje de un sujeto. 15
Figura 2.2.- Segmento de parada 16
Figura 2.3.- Diagrama de flujo de la metodología 21
Figura 3.1.- Gráfica de dispersión de diferencias de la velocidad GPS y velocidad derivada 28
Figura 3.2.- Distribución de número de viajes 31
Figura 3.3.- Puntos GPS cercanos a los POIs 31
Figura 3.4.- Densidad de velocidad en puntos GPS cercanos a los POIs 32
Figura 3.5.- Densidad de distancias de puntos GPS cercanos a los POIs 32
Figura 3.6.- Frecuencia de duración en viajes 34
Figura 3.7.- Frecuencia de distancias recorridas en viajes 35
Figura 4.1.- Flujo para la detección de PUS 37
Figura 4.2.- Proximidad de los PUs a los POIs 41
Figura 4.3.- Densidad de p.distance para puntos cercanos a los POIs 43
Figura 4.4.- Rendimiento de parámetro min_pts en algoritmo basado en densidad. 44
Figura 4.5.- Rendimiento de parámetros timeTh y distTh en algoritmo basado en distancia-tiempo. 45
Figura 4.6.- Desviación estándar de cambio de dirección de segmentos. 46
Figura 4.7.- Rendimiento de parámetro timeTh en algoritmo basado en velocidad.. 47
Figura 4.8.- PUs detectados con timeTh en algoritmo basado en velocidad. 47
Figura 4.9.- Evaluación de viajes con algoritmo basado en densidad 49
Figura 4.10.- Evaluación de viajes en algoritmo basado en distancia-tiempo 50
Figura 4.11.- Evaluación de viajes en algoritmo basado en velocidad. 51
Figura 5.1.- Flujo en un segmento de parada. 54
Figura 5.2.- Matriz de correlación de atributos de PUs 66
Figura 5.3.- Árbol de decisión para clasificación de PUs. 68
Figura 5.4.- Variación del error cuadrático medio al variar el número de árboles 71
Figura 5.5- Importancia de atributos en la deducción de PUs 71

Índice de Tablas

Tabla 3.1.- Atributos de puntos GPS en dataset I. 25
Tabla 3.2.- Nuevos atributos de puntos GPS en dataset I. 25
Tabla 3.3.- Atributos los puntos de interés del dataset II 27
Tabla 3.4.- Resultado de limpieza de datos en dataset I 29
Tabla 3.5.- Resultado de limpieza de datos en dataset II 30
Tabla 3.6.- Análisis de puntos GPS cercanos a los POIs 33
Tabla 3.7.- Resultado de muestra de dataset I: viajes 33
Tabla 3.8.- Resultado de muestra de dataset II: POIs 34
Tabla 4.1.- Subconjuntos de sujetos para selección de umbrales óptimos 42
Tabla 4.2.- Rango de valores para selección de parámetros. 42
Tabla 4.3.- Evaluación de parámetros en algoritmo basado en densidad 44
Tabla 4.4.- Evaluación de parámetros en algoritmo basado en distancia-tiempo. 45
Tabla 4.5.- Evaluación de parámetros en algoritmo basado en velocidad. 48
Tabla 4.6.- Resultados obtenidos con los tres algoritmos de detección de PUs. 52
Tabla 5.1.- Matriz de confusión para evaluación de modelos de clasificación 63
Tabla 5.2.- Conjunto de entrenamiento y prueba para la clasificación 65
Tabla 5.3.- PUs y número de viajes del conjunto de entrenamiento. 65
Tabla 5.4.- Datasets de PUs para evaluación de algoritmos de clasificación 67
Tabla 5.5.- Error estándar relativo con respecto al tamaño del árbol 67
Tabla 5.6.- Resultados de árbol de decisión para exactitud 68
Tabla 5.7.- Resultados de árbol de decisión para precisión y exhaustividad. 69
Tabla 5.8.- Resultados de árbol de decisión para F-score 70
Tabla 5.9.- Resultados de bosques aleatorios para exactitud 72
Tabla 5.10.- Resultados de bosques aleatorios para precisión y exhaustividad 72
Tabla 5.11.-Resultados de bosques aleatorios para F-score 73
Tabla 5.12.- Resultados de máquinas de soporte para evaluación de kernels 73
Tabla 5.13.- Resultados de máquinas de soporte para exactitud 74
Tabla 5.14.- Resultados de máquinas de soporte para precisión y exhaustividad 74
Tabla 5.15.- Resultados de máquinas de soporte para F-score 75
Tabla 5.16.- Resultados obtenidos con los tres algoritmos de clasificación 75

CAPÍTULO 1

1. Introducción

El análisis de datos del movimiento de las personas representa una oportunidad para la comprensión de patrones y actividades de las personas en los estudios de movilidad [1]. Una práctica común antes de la existencia de los teléfonos inteligentes era realizar un diario o encuestas donde se solicitaba a una muestra poblacional que proporcione un reporte de sus actividades y sus viajes, los cuales eran utilizados después para el análisis de su comportamiento [48]. Sin embargo, la recopilación de datos a través de estos medios es costoso y los datos resultantes son limitados e imprecisos ya que un sujeto puede informar sobre unos cuantos días o no se tiene la certeza de que la información proporcionada sea la correcta.

Posteriormente, se empezó a recopilar datos de geo-localización con dispositivos de rastreo GPS (*Sistema de Posicionamiento Global*) [11] portátiles o a través de teléfonos inteligentes que generan datos geo-localizados usando múltiples sensores: GPS, radio, RFID, radar, redes celulares, Wi-Fi, Bluetooth y otros. Dado que a diario decenas de millones de personas se mueven a lo largo de millones de trayectorias [17] y que los datos proporcionados por los sensores GPS son más precisos en el tiempo de registro y la ubicación de los viajes [40] que la recolección manual; la popularidad de los sensores GPS y los teléfonos móviles ha aumentado y consecuentemente la cantidad de datos recolectados.

Sin embargo, los datos recopilados por los sensores consisten únicamente en coordenadas geográficas con marcas de tiempo y carecen de información semántica. La información semántica es útil para estudiar el comportamiento de las personas, la provisión de servicios de ubicación y el diseño de aplicaciones para ordenadores portátiles y teléfonos móviles [3] [15] [50]. Así como también, brindar información a los gobiernos para preparar planes de movilidad que contribuyan al desarrollo de una ciudad. En los últimos años, varios autores han explorado formas de extraer información útil de estos datos de forma rápida y precisa [15] [47] [48] con el propósito de agregar significado a las posiciones y viajes de los sujetos.

En efecto, el uso de dispositivos móviles con GPS incorporado ha permitido a más personas acceder a aplicaciones de localización, y seguimiento; por lo que, el análisis de *puntos de ubicación frecuentes (PUs*) que corresponden a *puntos de interés (POIs*) a partir de los datos recopilados ha despertado el interés de los investigadores. PU se refiere a las paradas en donde un sujeto se detiene con frecuencia; mientras que los POIs son lugares donde el sujeto reportó que se detiene para realizar alguna actividad en dicho lugar o cerca de él. Sin embargo, los escasos recursos energéticos de un dispositivo móvil son una limitación importante para las aplicaciones de seguimiento que requieren calcular dinámicamente dicha información y enviar correcciones GPS [43].

La detección de POIs es una de las tareas clave al estudiar el comportamiento de las personas ya que representan lugares en donde las personas pasan una cantidad considerable de tiempo [12] o los visita con frecuencia [21]; lugares como por ejemplo casas, trabajos, restaurantes, hospitales, entre otros. Indudablemente, identificar cómo y por qué se mueven las personas a los POIs es crucial para una gama de políticas y decisiones en áreas como las telecomunicaciones y la infraestructura de transporte [19]; así como también, para servicios de predicción, navegación y recomendaciones personalizadas [27]. Al respecto, varios autores han investigado formas de detectar PUs que corresponden a POIs en las trayectorias GPS considerando distintos métodos [6] [15] [27]; sin embargo, pocos trabajos se han desarrollado con el propósito de dar significado a los PUs de forma automática [28] [1].

Cabe señalar que para objeto de esta investigación se ha delimitado los POIs únicamente a casa o trabajo; por ello, este trabajo consiste en el análisis automático de viajes cotidianos para identificar PUs que corresponden a dichos POIs a través de: 1) aplicar, analizar y evaluar tres algoritmos para la detección de PUs casa y trabajo y 2) examinar, contrastar y comparar tres algoritmos tradicionales de clasificación para dar significado a los PUs empleando aprendizaje supervisado. Se emplearon datos GPS recopilados a través de la aplicación móvil *Routecoach* que fue creada por la *Universidad de Ghent* como parte de una campaña de *crowdsourcing* en la provincia de *Flemish-Brabant (Bélgica)*; con el objetivo de desarrollar herramientas de evaluación y planificación para los proyectos de movilidad, que sean transferibles y que puedan ser adoptados por los planificadores urbanos de la provincia [26].

El resto del documento está organizado de la siguiente manera: en el *capítulo 2* se presentan algunos aspectos preliminares como definiciones, trabajos relacionados y

se define el propósito de este trabajo, en el *capítulo 3* se describe el proceso para la preparación de los *datasets* utilizados, en el *capítulo 4* se desarrolla la evaluación de algoritmos para la detección de PUs que correspondan a los POIs casa y trabajo, en el *capítulo 5* se entrenan algoritmos de clasificación para asignar un significado a los PUs detectados. Finalmente, se presentan las conclusiones generales y trabajos futuros.

CAPÍTULO 2

2. Marco teórico

2.1. Definiciones

A continuación se presentan definiciones de las entidades que se van a utilizar en el resto del manuscrito.

Punto GPS (p): es un par de coordenadas geográficas que representan una posición en el espacio en cierto instante de tiempo $p(x, y, t)$, donde x es longitud, y es latitud y t es la *fecha y hora* en que se registró el punto. En la práctica los dispositivos GPS reportan atributos adicionales (ver *Tabla 3.1*).

La longitud está en un rango de -180 y 180°, los valores negativos son para ubicaciones al este del meridiano de *Greenwich* y los valores positivos al oeste del meridiano de *Greenwich*. La latitud está en un rango -90° y 90°, los valores negativos son para ubicaciones en el hemisferio sur, los valores positivos en el hemisferio norte y el valor 0° en el Ecuador.

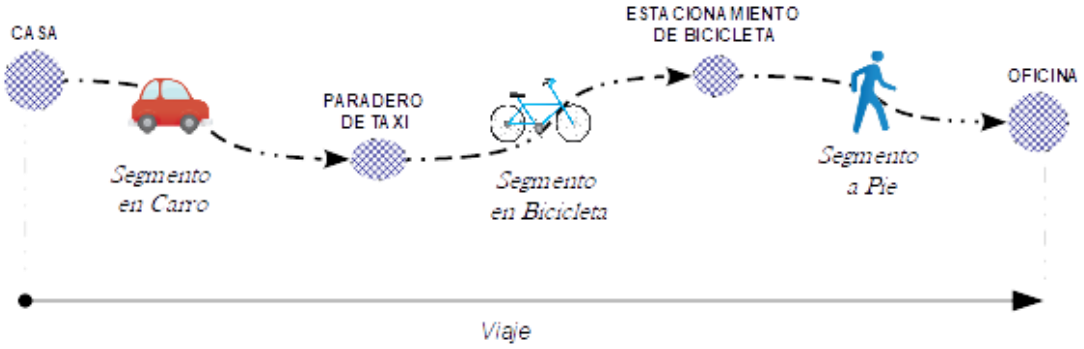


Figura 2.1.- Viaje de un sujeto.

Segmento (S): es un conjunto puntos GPS de un viaje S= $\{p_1, p_2, p_3, \dots, p_m\}$ $\therefore m < n$; donde $S_i \epsilon \tau$ para $i = 1,2,3, \dots, m$, τ es un viaje, m es el número total de puntos de un segmento y n es el número total de puntos de un viaje. El segmento representa el desplazamiento de un sujeto que se mueve entre dos paradas usando sola un medio de transporte. Por ejemplo, en la *Figura 2.1* un sujeto se desplazó de su casa al paradero de taxi utilizando un vehículo (segmento en carro).

Modo de transporte: es la forma como se transporta el sujeto; este puede ser: auto, tren, bicicleta, otros.

Viaje (τ): es una secuencia de n puntos GPS ordenados cronológicamente en el tiempo con un propósito definido $\tau = \{p_1, p_2, p_3, \dots, p_n\}$; donde $p_i \in \tau$ para $i = 1,2,3, \dots, n$. Un viaje puede ser una trayectoria unimodal o multimodal. La *Figura 2.1* muestra el viaje de un sujeto cuyo propósito es movilizarse de su casa a su oficina usando varios modos de transporte.

Trayectoria unimodal: son viajes que incluyen a un único medio de transporte. Por ejemplo: los datos recolectados con dispositivos GPS instalados en vehículos; los cuales son encendidos/apagados simultáneamente con el motor.

Trayectoria multimodal: son viajes que incluyen más de un modo de transporte. Por ejemplo, los datos recolectados con aplicaciones móviles con *smartphones* GPS embebidos en que una persona se desplaza de casa a trabajo manejando una bicicleta y caminando.

Parada (L): es una región donde un sujeto permanece de forma estática o moviéndose lentamente durante un tiempo y área determinados al realizar un viaje en lugares como: casas, paraderos de taxi, estacionamientos de bicicletas, otros.

Segmento de parada ($\mathcal{W}$): es un conjunto de puntos GPS contenido en una parada (*Figura 2.2*), donde p_a es el punto de arribo del sujeto a la parada y p_b es el punto de salida.

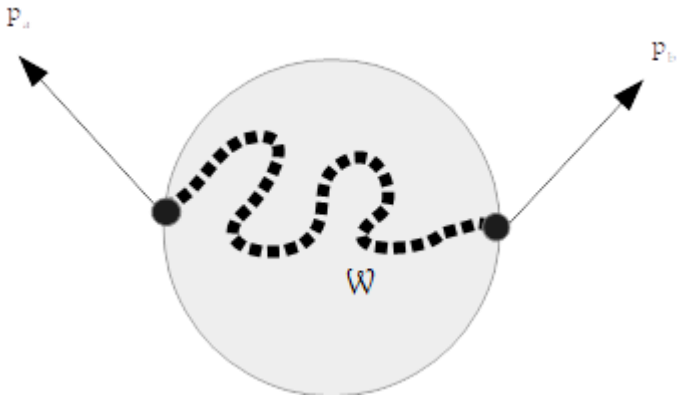


Figura 2.2.- Segmento de parada

Punto de una parada (C): es el *centroide* de una parada. Se representa con una tripleta $\mathcal{C}(\{x, y\}, t_a, t_b)$ que representa una parada; donde $c.x$ y $c.y$ son coordenadas geográficas (ver cálculo en *Sección 5.1.4*), $c.t_a$ es el tiempo de arribo, y $c.t_b$ es el tiempo de salida.

Punto de ubicación (PU): es el centroide $\{x, y\}$ de uno o varios puntos de parada $PU = \{\mathcal{C}_i, \dots, \mathcal{C}_j, \dots, \mathcal{C}\}$ [52]; donde $\mathcal{C}$ es un punto de parada y $\{x, y\}$ son coordenadas geográficas (ver cálculo en Sección 5.1.4).

Punto de Interés (POI): es la ubicación $(\{x, y\}, sem)$ declarada por el sujeto con una descripción semántica [35]; donde $\{x, y\}$ representan a las coordenadas geográficas latitud y longitud; mientras que, *sem* es su descripción semántica (ver *Tabla 3.3*).

2.2. Trabajos relacionados

Se han explorado varios métodos para la detección de PUs que corresponden a POIs en puntos GPS; los cuales se enfocan en: densidad, distancia-tiempo y velocidad.

Los métodos basados en densidad suponen que en las paradas la cantidad de puntos GPS es mayor que en el resto del viaje y fueron propuestos inicialmente por Ashbrook y Starner [3] y Zhou, et al. [54] y desarrollados en base a algoritmos de agrupamiento tradicionales como el *K-means* y el *DBSCAN (Density-based spatial clustering of applications with noise en inglés)* que agrupan los puntos GPS empleando medidas tales como la distancia entre puntos GPS o densidad de conectividad en un espacio cartesiano bidimensional. *K-means* [29] es un algoritmo basado en el análisis de varianzas que permite agrupar un conjunto de datos en un, valor *k*, número predefinido de grupos. *DBSCAN* [13] es un algoritmo que utiliza el concepto de puntos núcleo para encontrar los grupos, no requiere un valor k y permite localizar grupos de alta densidad indistintamente de su tamaño.

Así, en el caso de [3] se observó que los dispositivos GPS no funcionaban bien en el interior de los edificios ya que la recepción de la señal GPS es bloqueada por la estructura; por tanto, trataban los periodos de datos perdidos como paradas. Una vez extraídas, estas paradas se agruparon utilizando una variación del algoritmo de *k-means* cuya entrada es un valor *k* y un radio para lo cual se realizó el agrupamiento múltiples veces en diferentes parámetros. En cambio, en [54] se propuso una variación del algoritmo *DBSCAN*, denominado *DJ-Cluster,* el cual emplea la noción de conectividad entre puntos y el concepto de puntos compartidos para encontrar grupos. Los puntos GPS que se registran en una misma área pero en diferentes momentos son fusionados en un grupo.

En general los métodos basados en densidad consideran la dimensión espacial pero ignoran las características secuenciales temporales ya que trabajan sobre trayectorias completas; por ejemplo, los lugares que no portan un significado semántico como cruces de carreteras en donde varios sujetos pasan de forma iterativa y generan regiones densas.

Con respecto a los métodos basados en distancia-tiempo propuestos por Kang, et al.[21], y Zheng, et al. [52], los autores consideran como paradas a los puntos GPS ordenados cronológicamente que están dentro de un umbral predefinido de distancia (*distTh*) y tiempo (*timeTh*). En [21] el algoritmo denominado *"time-based clustering"* procesa los puntos gradualmente para encontrar paradas y utiliza los umbrales de timeTh y distTh para eliminar paradas falsas. Y en [52] el algoritmo identifica paradas caracterizadas por puntos GPS consecutivos dentro de un umbral predefinido de timeTh y distTh. Ambos métodos son adecuados para extraer paradas con forma circular como las paradas de autobuses, no lo son para la extracción de paradas más grandes en lugar de paradas con formas arbitrarias; por ejemplo, un sujeto caminando lentamente en un centro comercial.

Por otra parte, los métodos basados en velocidad propuestos por Palma, et al. [34] y Bhattacharya, et al. [7] asumen que, en las paradas, los puntos GPS registran una velocidad mínima. De acuerdo a, [34] el algoritmo *CB-SMoT (Clustering-Based Stops and Moves of Trajectories en inglés)* genera grupos de puntos en donde la velocidad es inferior a un umbral dado durante un tiempo corto. Mientras que en [7] se tomó en cuenta que los dispositivos GPS causan pseudo-movimientos pequeños debido a errores de posicionamiento incluso cuando el usuario está en reposo, y propuso un algoritmo que considera el cambio de dirección, la velocidad y la aceleración para extraer paradas en base a *segmentos estáticos* o *con poco movimiento*.

Otro problema para detectar paradas es la escasa cobertura de la señal GPS en entornos urbanos, en particular dentro de edificios. Bhattacharya, et al. [6] propuso un método basado en el problema clásico de la aguja de *Buffon-Laplace* [2]. Primero por cada punto GPS impreciso se generan puntos aleatorios (en tres dimensiones: latitud, longitud y tiempo) dentro de un círculo de incertidumbre de acuerdo a la estimación de error. Luego, se obtiene segmentos de línea uniendo puntos GPS individuales en orden cronológico para identificar paradas con la hipótesis de que si un sujeto se encuentra dentro de un POI, el número de segmentos de línea GPS que intersecan será mucho mayor que si el usuario simplemente se mueve alrededor del POI.

Recientemente, se han propuesto métodos híbridos por Fu, et al. [15] y Luo, et al. [27]. En [15] emplearon un algoritmo denominado *TDBC (spatio-temporal clustering based on time and distance)* para encontrar paradas según tres definiciones: a) el punto inicial p_0 y punto final p_m de cada trayectoria $L = (p_0, p_m) < \delta_d$, b) una región donde una persona permanece en un lugar durante un largo tiempo sin pérdida de señal $L = \{p_a, p_{a+1}, \dots, p_b\}$ donde $a \leq i \leq b$, $distance(p_a, p_b) < \delta_d$ y $|p.t_b - p.t_a| > \delta_t$, y c) una región donde la señal se pierde debido a barreras, edificios o problemas de equipo $L = \{p_a, p_{a+1}, \dots, p_b, \dots, p_c, p_{c+1}, \dots, p_d\}$ donde $a \leq i \leq d$, $distance(p_a, p_d) < \delta_d$ y $|p.t_d - p.t_a| > \delta_t$; siendo p un punto GPS, δ_d un umbral de distancia y δ_t un umbral de tiempo.

En [27] se propone otra variación del algoritmo *DBSCAN* cuya base es que en una parada hay una menor *capacidad de movimiento* y una mayor densidad. *Capacidad de movimiento* se define como la distancia punto a punto y la curvatura del punto inicial y final del segmento de una parada. Asimismo, para medir la densidad alrededor de un punto GPS, se introduce la *Teoría de Campo de Datos* [24] que plantea una función que permite calcular automáticamente un umbral de densidad de puntos.

Los trabajos permiten encontrar paradas que luego se representan como puntos de parada y como PUs sin *significado semántico*. Lv, et al. [28] y Andrienko, et al. [1] han tratado de asignar una semántica a PUs de forma automática. En [28] se exploraron características a nivel de visitas (calculadas en base a cada visita del sujeto al PU: día de la semana, hora del día, duración, índice de respuesta) y lugares (calculadas en base a valores estadísticos de las características a nivel de visitas) para clasificar los PUs en tipos definidos (casa, trabajo, restaurante, tienda, supermercado, recreación, negocios, turismo, aire libre) con datos de 10 sujetos que registraron sus trayectorias en un periodo de 2 meses. Primero, se entrenaron los clasificadores de *redes bayesianas (BN)*, *regresión logística (LR)* y *bosques aleatorios (RF)* en base a los lugares reportados por 8 sujetos y posteriormente los probaron los otros dos sujetos restantes; luego se entrenó un modelo mejorado en base al modelo estadísticos de los *Modelos Ocultos de Markov (HMM).* Los mejores resultados se obtuvieron con RF con una exactitud promedio del 75%; y con el modelo mejorado con una exactitud del promedio del 83%.

En [1] dan significado a los PU a través del análisis visual interactivo de un experto para lo cual crearon *firmas temporales* que caracterizan la distribución temporal de los PU con respecto a los ciclos de tiempos diarios y semanales, y las líneas de tiempo.

Primero, un analista dio significado (casa, trabajo, centro comercial, transportes, restaurantes, deportes) a un grupo de PUs seleccionados en base a sus *firmas temporales*, luego se clasificó el resto de PUs en base a un análisis de similaridad de sus firmas temporales. Para automatizar la asignación semántica en base a los ejemplos iniciales se estimó la similitud entre series de tiempo por medio de la distancia euclidiana. Antes de calcular las distancias se aplica un suavizado temporal y luego se transformó los valores absolutos en desviaciones normalizadas de las medias. Finalmente, probaron y demostraron su hipótesis con datos *GSM*, *GPS* y *Twitter*. Desafortunadamente, debido a un acuerdo de acceso de datos no se publicaron los resultados del experimento que realizaron con datos GPS sobre 38 sujetos en Suiza y una persona en Estados Unidos.

2.3. Propósito de la investigación

La falta de inversión y la infraestructura inadecuada son los principales problemas a los que se enfrentan las personas para movilizarse. Adicionalmente, el aumento de la demanda de transporte y de tránsito vial ha causado, particularmente en las ciudades grandes, congestión, demoras, accidentes y problemas ambientales [42].

Descubrir potenciales problemas en la infraestructura de transporte de una región es importante para plantear propuestas de solución de manera oportuna. Esto se puede realizar a través del análisis de actividades y viajes cotidianos de las personas usando datos GPS [48] [47]. Los viajes cotidianos generalmente tienen origen o destino a la casa y el trabajo. Las personas suelen movilizarse de la casa al trabajo por la mañana y del trabajo a casa por la noche [36]. En el aplicativo *Routecoach*, los viajes son reportados por los usuarios que indican el inicio y fin de cada segmento de viaje y su propósito.

Conocer el propósito del viaje o la actividad en el destino sigue siendo un problema difícil de resolver [47]. La ubicación de los PUs, en general, no coincide con los POIs declarados por el sujeto debido a errores en los datos GPS, falta de registro de inicio, fin o propósito de viaje, etc. Múltiples PUs pueden ocurrir en un viaje y sólo en ciertos casos se pueden asociar a actividades [16] que determinen su propósito. En este trabajo se evalúan tres algoritmos para detectar PUs con número cada vez mayor de viajes como datos entrada. Este parámetro es importante para las campañas de *crowdsourcing* pues nos indica cuántas veces un sujeto debe llegar o salir de un PUs para que éste sea detectado de manera consistente. Luego, para inferir cuáles PUs de

los detectados son casa o trabajo se compararon tres algoritmos de clasificación: árboles de decisión, bosques aleatorios y máquinas de soporte. Los PUs identificados se validaron usando su proximidad a los POIs casa o trabajo marcados por cada sujeto. Los resultados se resumen usando las métricas de *accuracy*, *precisión*, *recall y F-score*.

2.4. Metodología

La identificación de PUs que corresponden a POIs se realiza en tres fases (*Figura 2.3*): 1) pre-procesamiento de datos, 2) detección de PUs, y 3) clasificación de PUs en POIs casa y trabajo.

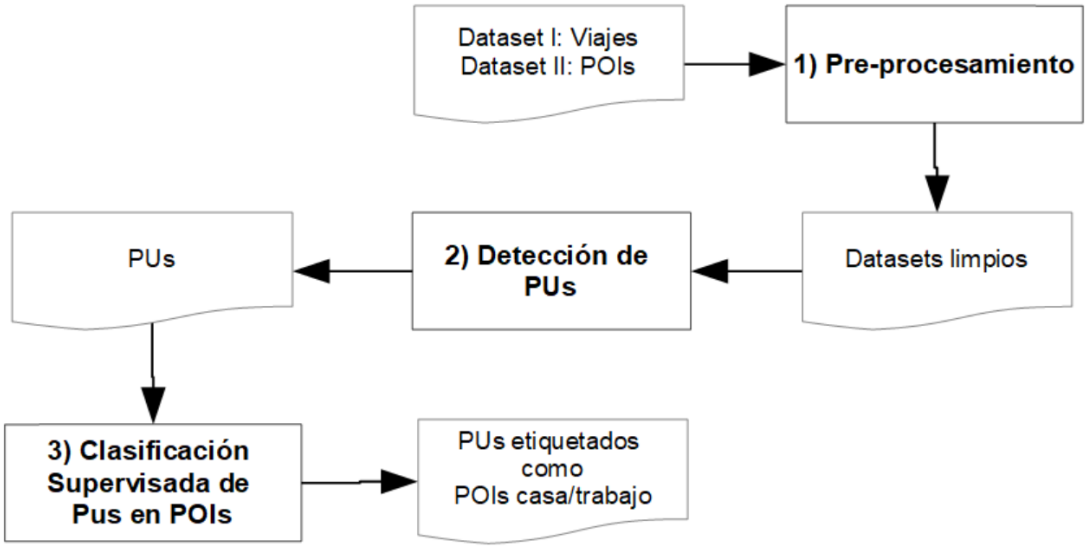


Figura 2.3.- Diagrama de flujo de la metodología

En la primera fase se descartan datos en base a reglas que consideran velocidades poco probables, puntos solitarios, puntos fuera de Bélgica, deterioro y ambigüedad de registros. Luego, se selecciona los sujetos que reportaron más viajes cercanos a los POIs.

En la segunda fase, para la detección de PUs que correspondan a los POIs casa y trabajo, se evalúa tres algoritmos con enfoques diferentes: el primero en base a la densidad de puntos, el segundo en base a un umbral de distancia-tiempo, y el tercero en base a la velocidad. Se determina también experimentalmente los mejores parámetros para cada uno de los algoritmos; y la cantidad de viajes mínimos requeridos para detectar PUs.

Finalmente, en la tercera fase, se calcularon atributos derivados de los puntos GPS y las visitas del sujeto a un PU. Luego se aplicaron tres algoritmos de clasificación basados en la semejanza o diferencia de sus atributos para etiquetar los PUs como casa y trabajo.

CAPÍTULO 3

3. Pre-procesamiento

Se cuenta con dos *datasets*, uno de viajes (*dataset I*) con puntos GPS y otro de POIs (*dataset II*) generados usando la aplicación móvil, *Routecoach*[1]. El proceso de recolección de ambos datasets se realizó durante un periodo de 6 meses entre el *2015-01-02* al *2015-06-23*. Para propósitos de este trabajo se seleccionó la muestra de sujetos considerando que los que reportaron sus viajes cotidianos de entrada y salida a casa/trabajo para el *dataset I*, y aquellos que reportaron la ubicación real de su casa/trabajo para el *dataset II*. Por ello, de un total de *8 303 sujetos*, se consideró en primera *instancia a 94 sujetos* que fueron los únicos que reportaron la ubicación de su casa o trabajo; sin embargo, algunos de ellos habían registrado menos de 5 viajes validos a casa o trabajo o simplemente los viajes reportados no iniciaban/terminaban en las casas o trabajos reportados; por lo cual se realizó un proceso de limpieza datos en tres fases donde se obtuvo la muestra resultante de *22 sujetos* que reportaron *423 viajes multimodales* para la experimentación ya que (ver detalle en secciones 3.2-3.4).

Por otra parte, aunque, cada sujeto reportó el modo de transporte en el que se movilizaba de algunos viajes, este atributo se descartó debido a que los sujetos tienden a marcar el modo de transporte antes o después del inicio o fin de un segmento de viaje; por ejemplo, un sujeto pudo haber marcado como modo de transporte *"tren"* mientras caminaba hacia la estación de tren.

En el *dataset I*, se identificaron dos problemas para el análisis automático de puntos GPS: 1) Puntos GPS faltantes debido a la falta de energía en la batería [35] o cuando el sujeto no reporta su localización a propósito. La aplicación *Routecoach* no puede identificar estos eventos y por lo tanto no es posible recuperar dichos puntos. 2) Puntos GPS erróneos debido a la imprecisión propia de los sensores GPS de teléfonos móviles [50]; por ejemplo, en áreas urbanas debido a la interferencia de la señal de satélites causada por los edificios. Para obtener la latitud y longitud se requiere al menos de tres satélites y de cuatro satélites para agregar la altitud [39] [31].

Los puntos GPS erróneos se identificaron de varias maneras. Una es a través del análisis de las velocidades poco probables; por ejemplo, un punto GPS donde la

[1] http://routecoach.ugent.be/

velocidad reportada sea superior a 300 km/h (límite de velocidad de un tren en Bélgica). Otra es midiendo la diferencia entre la velocidad derivada (*p.speedcalc*, ver cálculo en sección 3.1.1) y la velocidad reportada por el sensor GPS (*p.speed*) [22] ya que usualmente una diferencia grande puede ser causada por puntos GPS aislados, que generan un error considerable en el colado de las distancias acumuladas [47].

En el *dataset II* se extrajeron los POIs de las subcategorías casa o trabajo para evaluar su detección automática. En este *dataset* se detectó también cuatro problemas: 1) No todos los POIs son reportados, 2) Al reportar un POI la subcategoría asignada es incorrecta, 3) Hay POIs que estan ubicados a gran distancia de los puntos GPS de los viajes registrados en el *dataset I*. 4) Hay POIs de un sujeto con la misma subcategoría. Además, se detectó casos donde no se reportó POIs y por ende en dichos casos es imposible determinar si el resultado de la detección es correcto. Por otro lado, no se conoce la ubicación exacta de casa o trabajo de un sujeto. En la práctica, hay múltiples POIs casa o trabajo pero su ubicación no coincide. Para tener POIs que permitan validar la detección, primero se eliminan POIs aislados, luego se agrupan POIs que tiene la misma subcategoría y están asociados (existen puntos GPS cercanos) a viajes en el *dataset I*. Luego se descartan los grupos con baja densidad espacial. Cada grupo restante se representa con un único POI, ubicado en el *centroide* del grupo.

A continuación, se detalla los campos de los *datasets* utilizados este trabajo. En la *sección 3.2* se presentan las reglas utilizadas para preparar los *datasets* y los resultados de la limpieza de datos. En la *sección 3.3* se explica cómo se seleccionó la muestra de ambos *datasets* para los experimentos. Finalmente en la *sección 3.4* se realizó una validación de viajes con el objeto de garantizar que estos permitan la detección de los PUs que correspondan a los POIs.

3.1. Datasets

3.1.1. Dataset I: Viajes

El primer *dataset* contiene puntos de tipo GPS y FUSED de 693 viajes capturados a una frecuencia de *1 Hz*. En total el dataset I, contiene 927 016 puntos GPS (ver sección 2.1) y 859 080 puntos FUSED. Los puntos FUSED se refieren a la localización basada en múltiples sensores que usan además del GPS señales de sensores de radio, redes celulares, Wi-Fi, y Bluetooth. Debemos anotar que las especificaciones de cómo se generan las posiciones no fue provista por *Universidad de Ghent*.

Para evaluar si había diferencias entre los puntos GPS y FUSED se calculó el trayecto recorrido cada uno de los viajes y luego se los comparó a través del *coeficiente de correlación de Pearson*. Los resultados demostraron una correlación de *r=0.99* con un *p-value* $< 2.2 \times 10^{-16}$ lo cual significa que no existe una diferencia significativa. Consecuentemente, únicamente se utilizó los puntos GPS en nuestros experimentos. Los atributos de puntos GPS y FUSED se describen en la *Tabla 3.1*.

Tabla 3.1.- Atributos de puntos GPS en dataset I.

ATRIBUTOS	DESCRIPCIÓN	UNIDADES	FUENTE
p.type	Tipo de registro	FUSED, GPS	Automático
p.locid	Identificador del registro.	Valor numérico	Automático
p.deviceid	Identificador del dispositivo móvil.	Valor numérico	Teléfono
p.userid	Identificador de sujeto.	Valor numérico	Automático
p.tripid	Identificador de viaje.	Cadena de caracteres	Automático
p.x	Longitud	Grados	GPS
p.y	Latitud	Grados	GPS
p.altitude	Altitud	metros	GPS
p.speed	Velocidad reportada por el GPS logger	km/h	GPS
p.accuracy	Precisión reportada por el GPS logger	metros	GPS
p.t	Fecha y hora	yyyy-mm-dd HH:MM:SS	GPS
p.transportmode	Modo de transporte	BIKE, BUS, DRIVER, FOOT, PASSENGER, TRAIN, null	Sujeto
p.starttime	Fecha y hora de inicio del viaje	yyyy-mm-dd HH:MM:SS	Sujeto
p.stoptime	Fecha y hora de fin del viaje	yyyy-mm-dd HH:MM:SS	Sujeto

A los atributos del *dataset I* se agregaron los atributos adicionales que se presentan en la *Tabla 3.2*.

Tabla 3.2.- Nuevos atributos de puntos GPS en dataset I.

ATRIBUTOS	DESCRIPCIÓN	UNIDADES
p.distance	Distancia punto a punto	metros
p.timediff	Diferencia de tiempo punto a punto	segundos
p.speedcalc	Velocidad derivada	km/s
p.heading	Heading (dirección)	Grados [0, 360]
p.headingdiff	Cambio de dirección	Grados

A continuación se describe el cálculo de estos atributos:

— La **distancia punto a punto** (*p.distance)* se calculó aplicando la fórmula *Harvesine* [14] la cual asume que la tierra es una esfera y calcula la distancia desde el punto actual p_i al punto anterior p_{i-1} considerando las coordenadas longitud *p.x* y latitud *p.y* y el radio terrestre ecuatorial (R) de *6' 378 145 metros*.

$$\Delta p.y = |p.y_i - p.y_{i-1}|; \Delta p.x = |p.x_i - p.x_{i-1}|$$

$$\theta = sin^2\left(\frac{\Delta p.y}{2}\right) + \cos(p.y_i) \times \cos(p.y_{i-1}) \times sin^2\left(\frac{\Delta p.x}{2}\right)$$

$$p.distance_i = R \times \left(2 \times atan2\left(\sqrt{\theta}, \sqrt{1-\theta}\right)\right) \therefore i > 1$$

$$p.distance_1 = NA \therefore i = 1$$

— La **diferencia de tiempo** (*p.timediff*) diferencia entre la diferencia la hora y la fecha de registro del punto actual $p.t_i$ y la hora y fecha del punto anterior $p.t_{i-1}$ en segundos.

$$p.timediff_i = p.t_i - p.t_{i-1} \therefore i > 1$$

$$p.timediff_1 = NA \therefore i = 1$$

— La **velocidad derivada** *(p.speedcalc)* es el cociente entre la relación *distancia punto a punto* y la *diferencia de tiempo* [47] entre pares de puntos consecutivos p_i y p_{i-1}.

$$p.speedcalc_i = \frac{p.distance_i}{p.timediff_i} * 3.6 \therefore i > 1$$

$$p.speedcalc_1 = NA \therefore i = 1$$

— El **heading** *(p.heading)* es la dirección del punto actual con respecto al anterior. Se calcula a través del *azimuth* (ángulo inicial) siguiendo la ruta más corta de un elipsoide entre ambos puntos p_i y p_{i-1}:

$$p.bearing_i = atan2\ (A, B) \therefore i > 1$$

$$p.bearing_1 = NA \therefore i = 1$$

Donde A y B es igual a:

$$\Delta p.x_i = p.x_i - p.x_{i-1}; \Delta p.y_i = p.y_i - p.y_{i-1}$$

$$A = \sin\ \Delta p.x \times \cos\ p.y_i$$

$$B = \cos\ p.y_{i-1} \times \sin p.y_i - \sin p.y_{i-1} \times \cos\ p.y_i \times \cos\ \Delta p.x$$

Luego, el θ[-180°, 180°] se convirtió en θ[0°, 360°] con:

$$p.heading_i = (\theta_i + 360)\%360$$

— El **cambio de dirección** *(p.headingdiff)* [51] es la diferencia entre la dirección del punto actual *p.heading*$_i$ y el punto anterior *p.heading*$_{i-1}$.

$$p.headingdiff_i = \Delta p.headding = p.headding_i - p.headding_{i-1}$$

3.1.2. Dataset II: Puntos de Interés

El segundo *dataset* contiene 9 488 puntos de interés, para propósito de este trabajo se utilizaron aquellos POIs cuya categoría fue definida como clave (*key*) y la subcategoría fue casa (*home)* o trabajo (*work*). La lista completa de atributos de los POIs del *dataset II* se muestra en la *Tabla 3.3*.

Tabla 3.3.- Atributos los puntos de interés del dataset II

ATRIBUTOS	DESCRIPCIÓN	UNIDADES	FUENTE
I.category	Categoría del POI	GROCERY, KEY, LEISURE, MOBILITY	Automático
I.subcategory	Subcategoría del POI	**GROCERY:** shop, fuel, shoother library **KEY:** home, work, school, family, key-other childcare, other, business, null **LEISURE:** sport, lei-other, restaurant, drink, music, museum, cinema **MOBILITY:** station, bus-stop, parking, bicycle, parking	Sujeto
I.id	Identificador del POI.	Valor numérico	Automático
I.ownerid	Identificador de sujeto. Este campo es análogo a p.userid en un punto GPS.	Valor numérico	Automático
I.x	Longitud	Grados	Automático
I.y	Latitud	Grados	Automático
Para nuevos POIs (ver *sección 3.2*).			
I.density	Número de POIs de un grupo	Cantidad	-

3.2. Limpieza de datos

Para eliminar datos erróneos en el *dataset I* se aplicaron las siguientes reglas de forma secuencial:

1. Remover los puntos en donde los sensores GPS registraron velocidades superiores a: 50 km/h en áreas urbanizadas, 30 Km/h en áreas escolares, 120 km/h en autopistas, entre 70 y 90 en carreteras nacionales, y entre 160 - 300 km/h en vías ferroviarias [4][45].

2. Remover puntos cuya diferencia entre velocidad derivada y la velocidad reportada por el GPS es mayor a 25000 km/h (*Figura 3.1*).

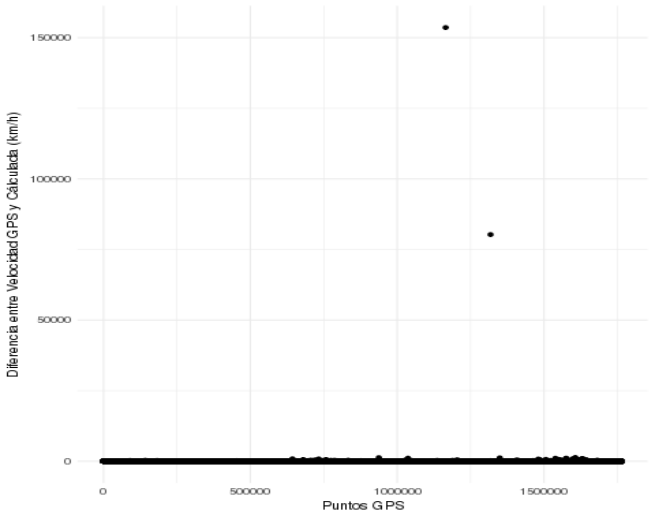


Figura 3.1.- Gráfica de dispersión de diferencias de la velocidad GPS y velocidad derivada

3. Remover puntos ubicados fuera de Bélgica.

4. Remover puntos de viajes de sujetos que no registraron los POIs casa o trabajo.

Es importante destacar que no se aplicaron otras reglas como la cantidad de satélites o el valor *Horizontal Disolution of Precision (HDOP)* [41][9] y los valores de altitud [38] por dos razones: 1) los valores de HDOP y cantidad satélites no están disponibles, y 2) la precisión de la altitud suele ser menor que la de las coordenadas latitud y longitud.

En el *dataset II* para definir un único POI por sujeto para las sub-categorías casa y trabajo se creó una nueva tabla siguiendo estas reglas en orden:

1. Descartar POIs con coordenadas nulas.

2. Descartar POIs sin una categoría y subcategoría registrada.

3. Descartar POIs cuya ubicación se identificó fuera de Bélgica.

4. Descartar POIs con coordenadas duplicadas

5. Descartar POIs con etiquetas de categorías distintas a *"clave"* y subcategorías *"casa"* y *"trabajo"*.

6. Calcular el centroide de los POIs de una misma subcategoría por sujeto y calcular la distancia entre cada POI y el centroide.

7. Aplicar *DBSCAN* usando un radio de 100 metros con un número mínimo de puntos igual a dos debido a que se buscaba agrupar a dos o más POIs cercanos para obtener una sola ubicación. En la mayoría de los casos los POIs están a una diferencia inferior a ese valor.

8. Para los grupos resultantes calcular el *centroide*, y usarlo como nuevos POIs, en el atributo *I.density* se guarda el número de elementos del grupo.

9. Descartar POIs que se encuentran a más de 0.5 kilómetros de sus puntos GPS en el *dataset I*.

10. Si existe más de un POI por sujeto con la misma subcategoría, mantener los POIs con mayor *i.density*. Si todos los POIs tienen el mismo valor de *I.density* se descartan.

Tabla 3.4.- Resultado de limpieza de datos en dataset I

		Datos originales	**Datos resultantes**	**Muestras**
Sujetos		94	61 (64.9 %)	23(24.5%)
Dispositivos		98	64 (65.3 %)	26(26.5%)
Periodo de recolección		6 meses	6 meses	6 meses
Viajes	Total	693	599 (86.4 %)	544(78.5%)
	Máximo	83	82	68
	Mínimo	1	1	4
Días	Total	138	131 (94.9%)	127(92%)
	Máximo	51	51	51
	Mínimo	1	1	3
Puntos GPS		927016	683590 (73.7 %)	622222(67.1%)
Modos de transporte		6	6	6

Al aplicar las reglas mencionadas previamente, el *dataset I* se redujo a 61 sujetos, 599 viajes y 683 590 puntos GPS recopilados en 131 días, ver columna 2 de la Tabla 3.4. El *dataset II* se redujo a 86 POIs casa o trabajo reportados por 61 sujetos, ver columna 2 de la *Tabla 3.5*.

Tabla 3.5.- Resultado de limpieza de datos en dataset II

		Datos originales	**Datos resultantes**	**Muestra**
Sujetos		94	61 (%)	23 (24.5%)
Puntos de interés		9488	89 (0.9 %)	41(0.4%)
Categorías		4	1	1
Subcategorías		26	2	2
Casa	*Total*	62	50 (80.6 %)	22(35.5%)
	Máximo por sujeto	24	1	1
	Mínimo por sujeto	0	0	0
Trabajo	*Total*	50	39 (78 %)	19(38%)
	Máximo por sujeto	6	1	1
	Mínimo por sujeto	0	0	0
Casa + Trabajo	*Máximo por sujeto*	25	2	2
	Mínimo por sujeto	0	1	1

3.3. Selección de muestras para experimentos

Si un sujeto viajó durante días ordinarios (de lunes a viernes), debería haber reportado un mínimo de 5 viajes a su casa o trabajo. Así de los 61 sujetos de los *datasets* resultantes del proceso de limpieza de datos, hay 23 sujetos que reportaron al menos 5 viajes ya que del resto de sujetos existe poca información para evaluar los algoritmos de detección de PUs. Por ende en el resto de este trabajo se utilizaron los registros de ambos *datasets* correspondientes estos 23 sujetos.

Asimismo, al revisar los *datasets* de la muestra se notó que la cantidad de viajes y días reportados para los 23 sujetos era heterogénea; por ejemplo, un sujeto tiene 83 viajes registrados mientras que otros únicamente 5 viajes. La mayoría de sujetos reporto entre 5-30 viajes como lo ilustra la Figura 3.2 y la muestra se formó por un total de 544 viajes con 622 222 puntos GPS reportados en 127 días diferentes (*Tabla 3.4*, columna 3) y 41 POIs, 22 casas y 19 trabajos (*Tabla 3.5*, columna 3).

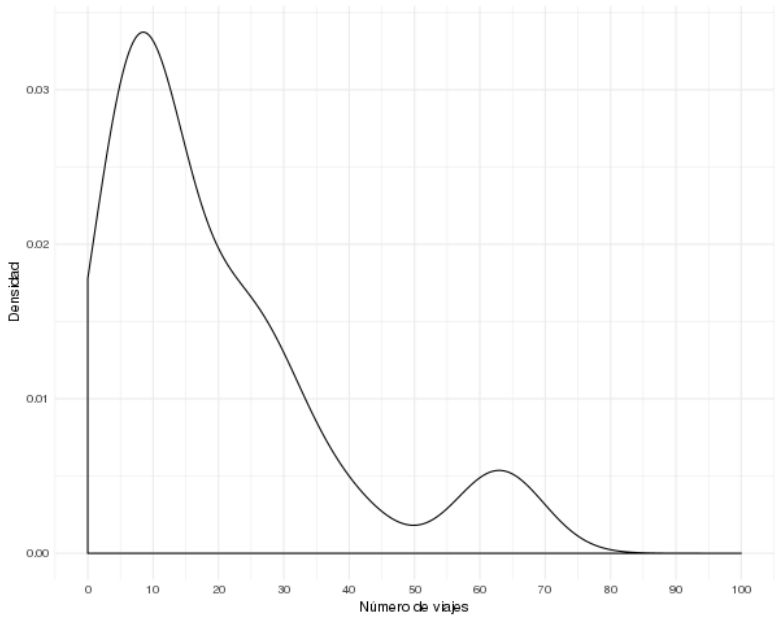


Figura 3.2.- Distribución de número de viajes

3.4. Validación de viajes

No todos los viajes en el *dataset I* permiten la detección de los PUs que corresponden a los POIs casa y trabajo del *dataset II* así por ejemplo, si un sujeto reportó la ubicación de su casa y trabajo, pero sólo registró viajes al trabajo durante tres días consecutivos; se podría identificar las paradas y consecuentemente el PU que corresponde al POI trabajo pero no así las paradas ni el PU que corresponde al POI casa. Para validar la posibilidad de detectar PU que correspondan a POIs se analizó los atributos de los puntos GPS cercanos a cada POIs, es decir, aquellos ubicados a menos de 500 metros, ver *Figura 3.3*.

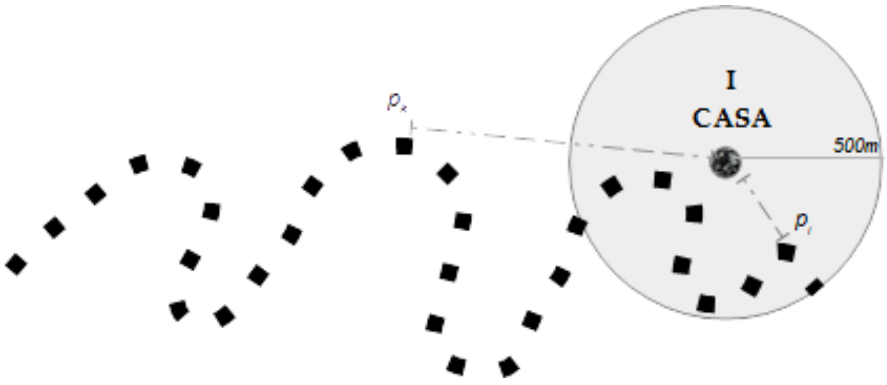
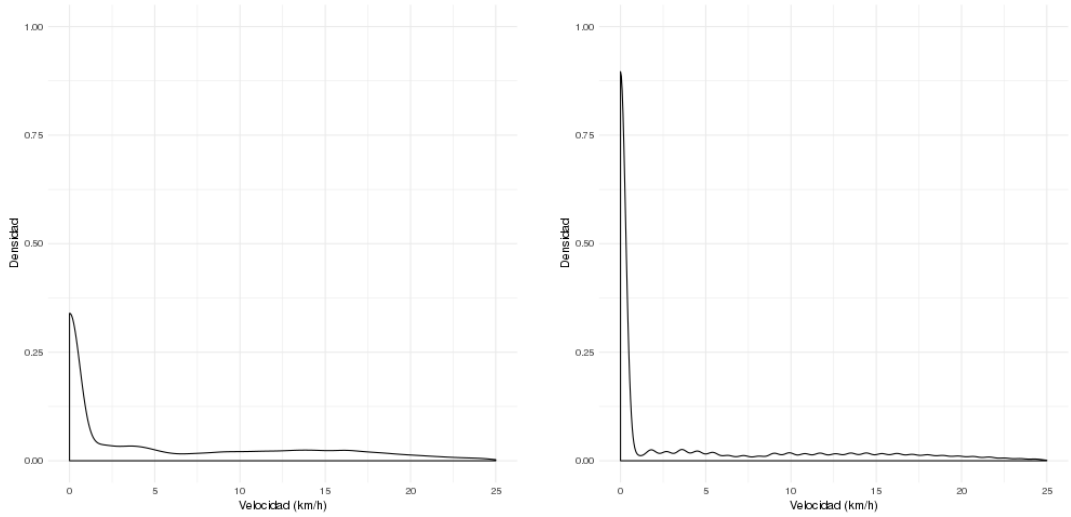


Figura 3.3.- Puntos GPS cercanos a los POIs

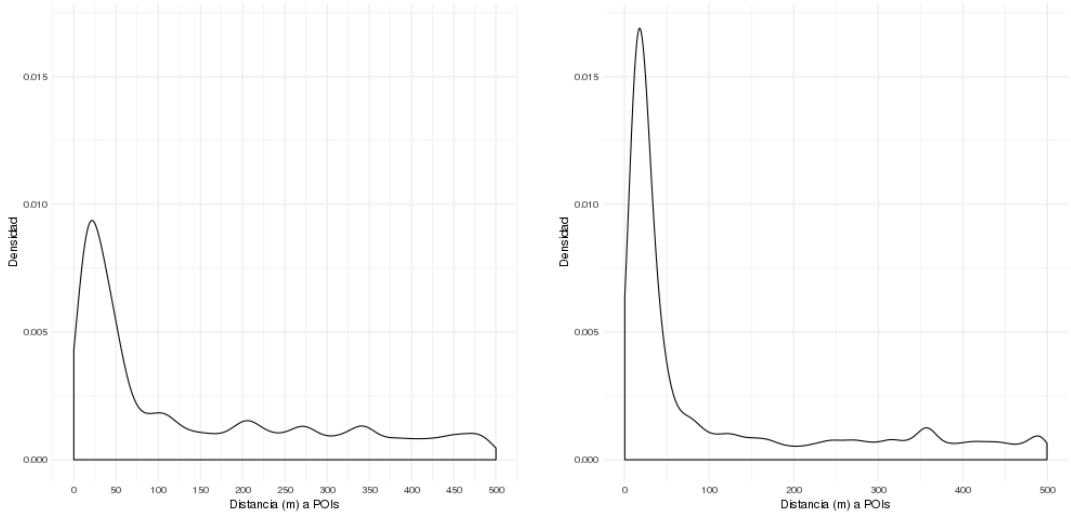
La *Figura 3.4* muestra la densidad de la velocidad de los puntos GPS cercanos a los POIs casa y trabajo en la cual se observa que la mayoría de velocidades son inferiores a 2.5km/h, lo que indica que estos pueden ser PUs. La velocidad promedio de una persona caminando varía entre 4.3 y 5.4 km/h [5][32].



a) Casa b) Trabajo

Figura 3.4.- Densidad de velocidad en puntos GPS cercanos a los POIs

Se examinó también la densidad de las distancias de los mismos puntos y se encontró que la mayoría de estos están ubicados a menos de 100 metros del POI, ver *Figura 3.5*.



a) Casa b) Trabajo

Figura 3.5.- Densidad de distancias de puntos GPS cercanos a los POIs

Además, para los radios de 50, 100, 150 y 500 metros alrededor de los POIs se determinó a qué sujetos y a cuántos viajes corresponden los puntos GPS cercanos. En la *Tabla 3.6* se presentan los resultados, 427 (78.5 %) de los 544 viajes registran puntos GPS para radio mayor o igual a 100 metros, radio a partir del cual se incluye el 100% de POIs (41).

Tabla 3.6.- Análisis de puntos GPS cercanos a los POIs

Radio (metros)	Sujetos		POIs	Viajes		
	Casa	Trabajo		Cantidad	Casa	Trabajo
50	22	18	40	400	333	153
100	**22**	**19**	**41**	**427**	**357**	**172**
150	22	19	41	438	365	187
300	22	19	41	450	384	201
500	22	19	41	451	387	209

En base a estos resultados se seleccionaron los sujetos que tuvieron 5 viajes con al menos un punto GPS a menos de 100 metros de un POI. El *dataset* de viajes válidos tiene un total de 423 viajes con 530559 puntos GPS reportados durante 120 días, los cuales corresponden a 22 sujetos, ver *Tabla 3.7* y *Tabla 3.8*. Es importante mencionar que los viajes válidos resultantes son multimodales aún después del procesamiento del *dataset*. De ahora en adelante el término *dataset I* se referirá a este conjunto datos y el término *dataset II* a los POIs agrupados (ver *Anexo I*).

Tabla 3.7.- Resultado de muestra de dataset I: viajes

		Dataset I: viajes
Sujetos		22
Dispositivos		25
Periodo de recolección		6 meses
Viajes	Total	423 (66.4%)
	Máximo	64
	Mínimo	**5**
Días	Total	120 (%)
	Máximo	44
	Mínimo	3
Puntos GPS		530559 (61.1 %)
Modos de transporte		6

Tabla 3.8.- Resultado de muestra de dataset II: POIs

		Muestra
Sujetos		22 (24.5%)
Puntos de interés		40(0.4%)
Categorías		1
Subcategorías		2
Casas	*Total*	22(35.5%)
	Máximo	1
	Mínimo	0
Trabajos	*Total*	18(38%)
	Máximo	1
	Mínimo	0

Finalmente, al analizar los viajes resultantes del *dataset I* se determinó que la mayoría de estos fueron viajes cortos con una duración entre 6-14 minutos y recorridos inferiores a 5 km como lo demuestran la *Figuras 3.6* y *3.7*.

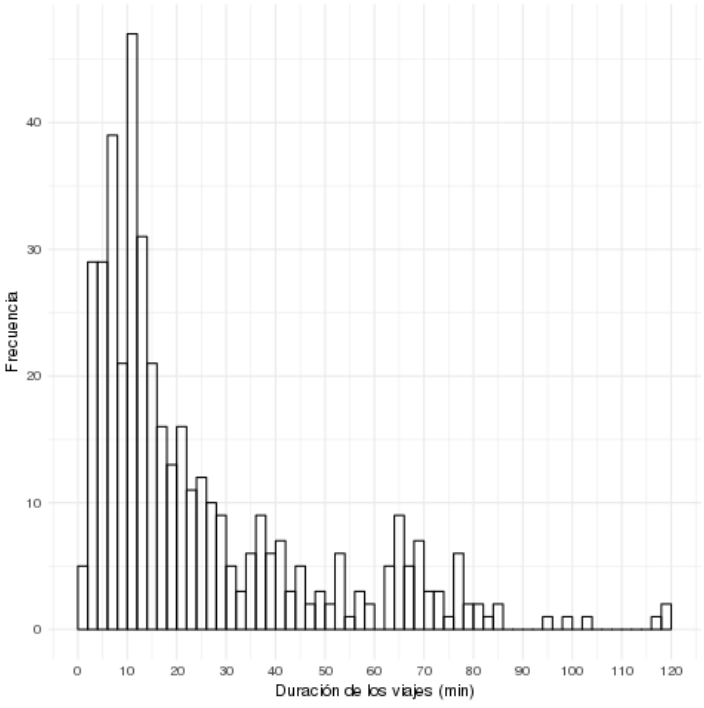


Figura 3.6.- Frecuencia de duración en viajes

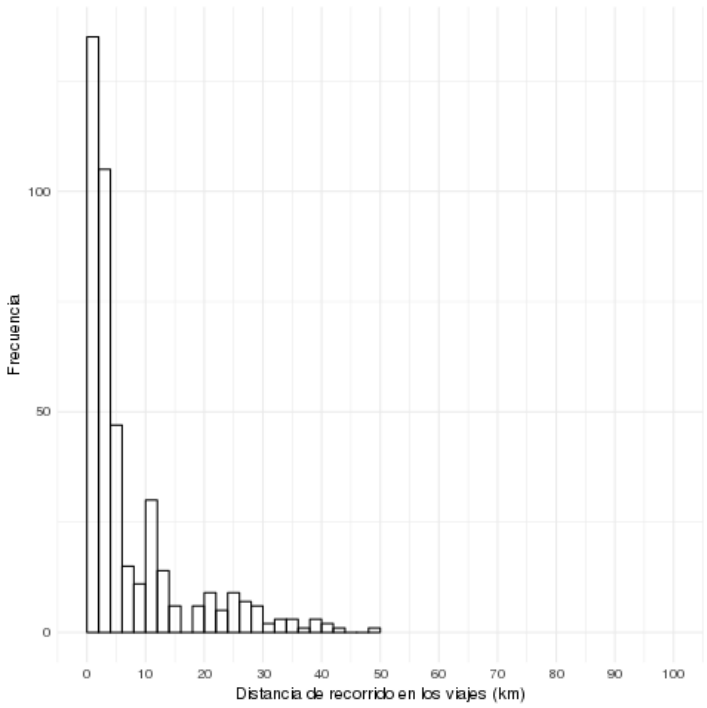


Figura 3.7.- Frecuencia de distancias recorridas en viajes

CAPÍTULO 4

4. Detección de Puntos de Ubicación

Los puntos de ubicación (PUs) representan la permanencia de un sujeto en determinados lugares [27] y proporcionan información para entender el comportamiento, en términos de movilidad, de los sujetos [15] [16] [27] [53]. Este trabajo se enfoca en la detección de PUs que correspondan a la casa y trabajo de un grupo de voluntarios que registraron sus viajes con la aplicación *Routecoach*. Para poder validar los resultados nuestro *ground truth* se asume que un PU cercano a un POI casa o trabajo es un *true positive*.

Este capítulo está organizado de la siguiente forma: en la *sección 4.1* se presenta los algoritmos de detección de PUs, en la *sección 4.2* se describe los experimentos realizados para estimar los parámetros de entrada de los algoritmos y para determinar el número de viajes mínimo para la detección de los PUs. Finalmente, en la *sección 4.3* se presentan los resultados obtenidos en los experimentos.

4.1. Metodología

Se han propuestos varios métodos para detectar PUs; sin embargo, debido a que la frecuencia de muestreo de puntos GPS varía así como el entorno, no existe un método estándar. El método utilizado en este trabajo se basa en la técnica de Fu et al., [15] el cual generaliza la mayoría de los enfoques previos [6] [15] [27] y supone que durante una trayectoria un sujeto puede permanecer *Th* segundos alrededor de un mismo lugar. Las posiciones del grupo de puntos GPS que corresponden a esta acción conforman el área de la *parada* L, cada *parada* se representa con el *centroide* del grupo denominado *punto de parada C*. Aquellos *puntos de paradas* de distintos viajes que están cerca uno de otros son agrupados en *puntos de ubicación U* con un algoritmo de densidad, ver *Figura 4.1*.

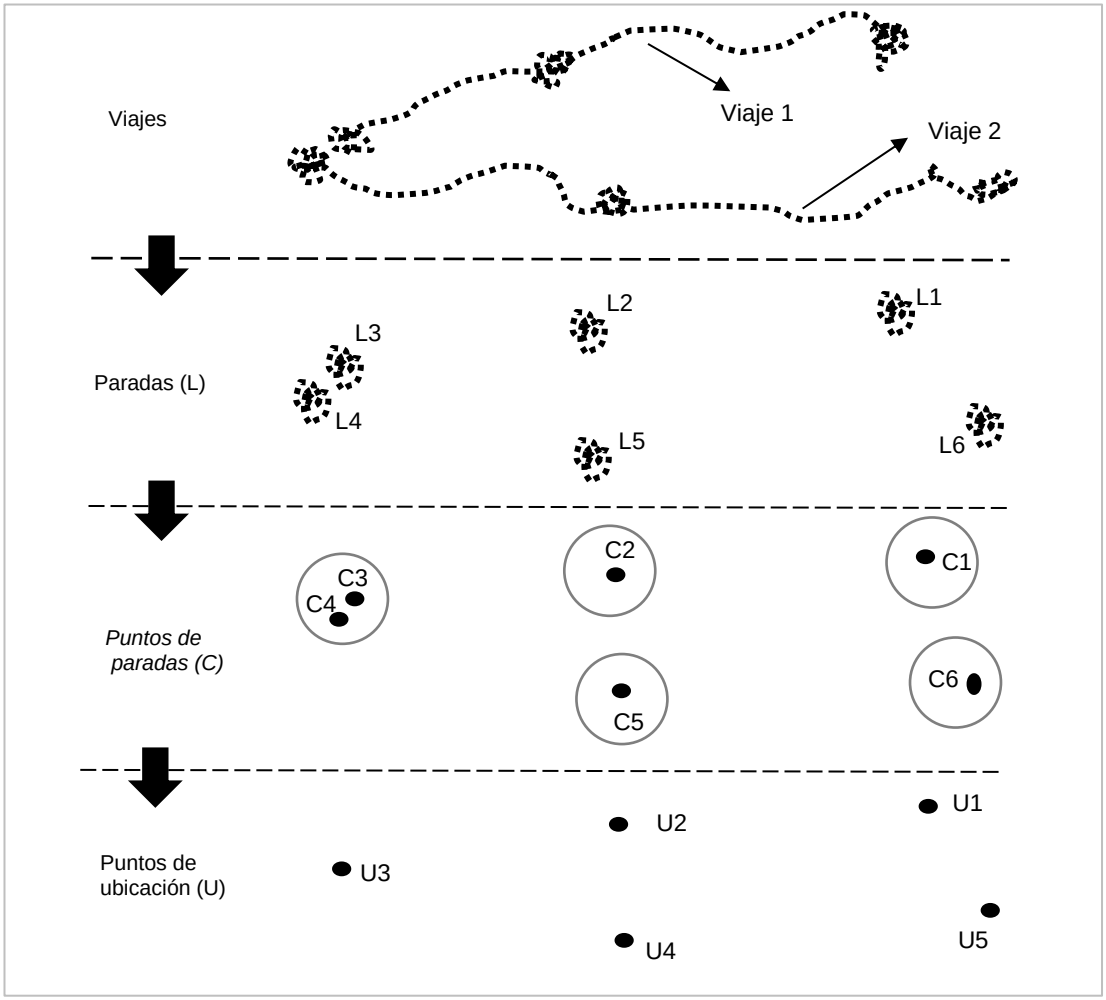


Figura 4.1.- Flujo para la detección de PUS

Los pasos del algoritmo de Fu et al, se listan en el *Algoritmo I*. Toma como entrada una trayectoria con viajes consecutivos en el tiempo y asume que el primer y último punto GPS de la trayectoria son paradas; sin embargo en nuestro *dataset* de viajes esta condición no se cumple ya que no hay control de cuando un sujeto reporta el inicio o fin de un viaje. Así por ejemplo, un sujeto pudo reportar que salió de su casa varios minutos después de ocurrido el hecho. Debido a que esta investigación busca identificar paradas en viajes únicos independientemente de la fecha en que ocurrieron y del tipo I a III [15], se emplearon tres algoritmos para la detección de dichas paradas: densidad de puntos [13], distancia-tiempo de permanencia [52], y velocidad [7].

Algoritmo I

Entrada: Trayectoria T, timeTh, distanceTh
Salida: setParadas

setParadas =NA; previoC=NA;
Si el primer o último p es Tipo I **entonces**
añadirCluster(punto) // Tipo I
Para cada p en T **hacer**
Si distancia(Cluster, p_i)<distanceTh **entonces**
añadir p en Parada; continuar;
Si distancia(Cluster, p_i)>distanceTh y duración(Cluster)>timeTh **entonces**
añadirCluster(Cluster); continuar; // Tipo II
Si distancia(Cluster, p_i)>distanceTh y duración(Cluster)<timeTh **entonces**
revisar(Cluster, previoC); continuar;
Si distancia(Cluster, p_i)<distanceTh y duración(p_{i-1}, p_i)>timeTh **entonces**
añadirCluster({p_{i-1}, p_i}); continuar; // Tipo III
Si distancia(Cluster, p_i)<distanceTh y duración(p_{i-1}, p_i)<timeTh **entonces**
Ignorar();
retornar setParadas

Funcion *revisar*(Cluster, Cluster previo){
Si (intervalotiempo (Cluster, Previous)<timeTh **y** distancia(Cluster, Previous) entonces
Previous=merger(Cluster, Previous);
Si Previous está un tipo II **entonces** añadirCluster(Previous);
Entonces Previous=Cluster;}

Funcion *añadirCluster*(Cluster){
Si (distancia(Cluster, punto de parada previo en setParadas)<distanceTh) **entonces**
Previous =merger(Cluster, Previo)
Sino poner Cluster en setParadas; previous= Cluster;}

Algoritmo basado en densidad de puntos

Detenerse por un periodo de tiempo considerable genera una región con alta densidad relativa de puntos GPS, por ejemplo al llegar o salir de su casa o trabajo. Para detectar dichas regiones se utilizó el algoritmo DBSCAN [13] el cual requiere de dos parámetros *eps* y *MinPts*. *Eps* es la distancia máxima entre dos puntos para que sean considerados vecinos y *MinPts* es el número mínimo de puntos que pueden formar un grupo. El resultado del DBSCAN es el grupo de puntos que forman cada parada, el resto de puntos que no se pueden agrupar se clasifican como ruido.

Algoritmo basado en distancia-tiempo de permanencia

En Zheng, et al., [25] [52] se asume que una parada se forma cuando un sujeto permanece estacionario durante un período de tiempo; así por ejemplo cuando el sujeto ingresa a un lugar cerrado y el sensor GPS pierde la señal del satélite hasta que el sujeto sale nuevamente de ese lugar. Este algoritmo detecta paradas buscando a lo largo de una trayectoria de una región espacial en donde el sujeto permanece un periodo de tiempo mayor a un umbral *timeThreh* y una distancia *distThreh*.

Algoritmo basado en distancia-tiempo de permanencia

Entrada: viajes, distThreh, timeThreh
Salida: Paradas

Para cada viaje **hacer**:
1. i=0; *pointNum*=|p| // Número de puntos
2. **Mientras** i < *pointNum* **hacer**
    a. j=i+1; Token=0;
    b. **Mientras** j < *pointNum* **hacer**
    c. dist = *Distance*(p_i, p_j); //Calcular la distancia entre puntos
    d. **Si** dist > *distThreh* **entonces**
        i. $\Delta t=p_j.t-p_i.t$; //Calcular diferencia de tiempo entre puntos
        ii. **Si** Δt>*timeThreh* **entonces**
            1. L=({p_k|i<=k<=j}) // Extraer parada
            2. paradas.insert(L); // Conservar parada
            3. i=j; Token=1
        iii. break
    e. j=j+1;
    f. **Si** Token!=1 **entonces** i=i+1;
3. **retornar** paradas

Algoritmo basado en velocidad

Este algoritmo [7] permite encontrar paradas bajo el supuesto de que cuando un sujeto se encuentra en una parada, la velocidad es pequeña y la dirección aleatoria. Este algoritmo detecta cuatro tipos de segmentos: 1) *segmentos estáticos dirigidos*, 2) *segmentos estáticos no dirigidos*, 3) *segmentos con poco movimiento dirigidos* y 4) *segmentos con poco movimiento no dirigidos*.

Los puntos GPS de un *segmento estático* poseen una velocidad de *0-3.6 km/h* y una aceleración de *+-1 km/h*^{2} y en un *segmento con poco movimiento* los puntos GPS reportan una velocidad entre *0-7.5 km/h* y no se considera la aceleración. De estos segmentos, se descartan aquellos con duración menor a *timeTh.* Los segmentos se sub-clasifican

en *dirigidos* o *no dirigidos* dependiendo de los valores de la desviación estándar de la dirección $\sigma(p.headingdiff)$. En el caso de segmentos estáticos si *σ(p.headingdiff)* es menor que Sth el segmento es dirigido caso contrario es no dirigido. Para segmentos con poco movimiento se usa *Dth* de forma análoga. *Sth* y *Dth* se determinaron experimentalmente.

Algoritmo basado en velocidad

Entrada: viajes**,** rangos de velocidad y aceleración, Sth, Dth, timeTh
Salida: Paradas

Para cada viaje **entonces**:

1. **Si** $p.speed_i \in \left[0 - 3.6\frac{km}{h}\right]$ & $p.acceleration_i \in \left[\pm 1\frac{km}{h^2}\right]$ **entonces**
    a. $p_i \in$ *segmento estático*
2. **Si** $p.speed_i \in \left[0 - 7.5\frac{km}{h}\right]$ **entonces**
    a. $p_i \in$ *segmento con poco movimiento*
3. Extraer *segmentos estáticos* y *con poco movimiento*.
4. **Para cada** segmento **hacer**
    a. $duración = p.t_b - p.t_a;\ \Delta B = \sigma(p.headingdiff)$
    b. **Si** $duración < timeTh$ **entonces** Descartar(segmentos)
    c. **Si** *segmento es estático* & $\Delta B > Sth$ **entonces**
        i. *El segmento estático* es *no dirigido (B>>0)*
    d. **Sino** El s*egmento estático* es *dirigido (B~0)*.
    e. **Si** *segmento con poco movimiento &* $\Delta B > Dth$ **entonces**
        i. *El segmento con poco movimiento es* no *dirigido (B>>0)*;
    f. **Sino** El s*egmento con poco movimiento* es *dirigido (B~0)*.
    g. Segmentos estático/poco_movimiento no dirigidos ∈ *L*
    h. paradas.insert(L);
5. **retornar** paradas

La salida de los tres algoritmos son un conjunto de paradas. Donde un punto de parada *C=(c.x, c.y)*, es el centroide del grupo[52]:

$$c.x = \sum_{i=1}^{n} \frac{p.x_i}{card(p)}$$

$$c.y = \sum_{i=1}^{n} \frac{p.y_i}{card(p)}$$

Luego se establecen las coordenadas de los *puntos de ubicación PU= (u.x, u.y)* agrupando los *puntos de parada* de tal manera que la distancia entre cada par de puntos sea menor o igual a 100 m. Para agrupar los *puntos de parada* se empleó de nuevo el

algoritmo de *DBSCAN* [13] debido a su capacidad de extraer grupos de forma arbitraria sin que el número de grupos se conozca a priori [49]. Las coordenadas de del P*U* se calculan como la mediana de las coordenadas de los *puntos de parada* agrupados con el objeto de que el *PU* resultante se ubique cerca de la mayoría de los *puntos de parada*.

4.1.1. Evaluación

Para evaluar la detección de PUs, se explota su proximidad hipotética a los POIs [14]. A diferencia de otras propuestas [33] que utilizan la intersección con geometrías de los POIs se construyó una geometría circular con un *radio* variable de 50, 100 y 150 metros alrededor de los POIs casa y trabajo reportados por los sujetos (*dataset II*). Por ejemplo, de los PUs U_k y U_i de la *Figura 4.2*, únicamente el punto de ubicación U_i se encuentra cerca de *CASA*. Una limitación de este enfoque es que para personas que trabajan y viven en el mismo lugar el PU detectado puede tener dos o ninguna etiqueta.

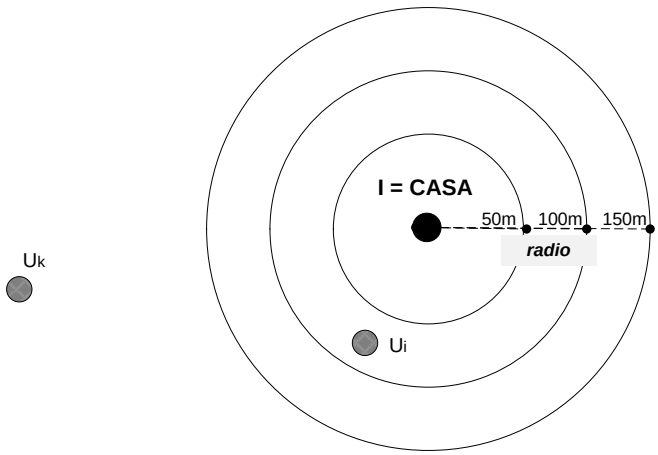


Figura 4.2.- Proximidad de los PUs a los POIs

Para calcular las tasas de detección se utilizó como métrica únicamente el *TPR* y no el *FNR* debido a que el *dataset II* contiene únicamente los POIs que registró cada sujeto. La métrica *True Positive Rate (TPR)* mide el porcentaje de PUs identificados correctamente en un valor que oscila entre los valores 0 y 1, siendo el 1 el valor máximo posible y 0 el mínimo.

$$TPR_r = \frac{TP_r}{TP_r + FN_r}$$

TP corresponde a los *PUs* que están dentro de la geometría circular de radio(*r*) alrededor de los POIs reportados por los sujetos y *FN* son los PUs ubicados fuera de ella.

4.2. Experimentos

Los algoritmos para la detección de paradas mencionados previamente requieren varios parámetros de entrada pero no hay parámetros predefinidos, ni tampoco se conoce cuántos viajes se requieren para detectar correctamente un PU. Para resolver estos problemas se realizaron dos experimentos que se describe a continuación:

4.2.1. Estimación de Parámetros

Se dividió los *datasets (sección 3.3)* de la muestra de los 22 sujetos en dos subconjuntos. El primero con el 30% de los sujetos, se utilizó para extraer los mejores parámetros (entrenamiento). El resto de sujetos para calcular las tasas de detección con los mejores parámetros obtenidos, ver *Tabla 4.1.*

Tabla 4.1.- Subconjuntos de sujetos para selección de umbrales óptimos

Subconjuntos	**Casa**		**Trabajo**	
	Sujetos	**Viajes**	**Sujetos**	**Viajes**
Entrenamiento (30%)	6	105	6	49
Prueba (70%)	16	252	12	119
Total	***22***	***357***	***18***	***168***

Tabla 4.2.- Rango de valores para selección de parámetros.

Algoritmos basados en:	**Parámetros**	**Rango**	**Unidad de medida**
Densidad	*eps1*	0-100	*metros*
	min_pts1	2-120	-
Distancia tiempo	*timeTh*	5-120	*segundos*
	distanceTh	15-150	*metros*
Velocidad	*timeTh*	0-120	*segundos*
	Sth	0-360	*grados*
	Dth	0-360	*grados*

Para establecer *min_pts1*, *timeTh*, *distanceTh* y *timeTh* se aplicó cada algoritmo de detección de paradas sobre el *subconjunto de entrenamiento* con los valores de los parámetros definidos en la *Tabla 4.2* y se calculó el TPR para cada *subconjunto de*

entrenamiento usando radios de 50, 100 y 150m. Así, para cada sujeto del *subconjunto de entrenamiento* se eligió un viaje y se ejecutó el algoritmo de detección. Este procedimiento se repitió 30 veces escogiendo cada vez un viaje aleatorio por cada sujeto y finalmente se calculó el TPR promedio para cada uno de los parámetros con incrementos de 5.

Se seleccionaron los parámetros de cada algoritmo de detección de paradas que reportaron el TPR promedio más alto y para corroborar las tasas de detección se repitió el mismo procedimiento sobre el *subconjunto de prueba*.

El resto de parámetros se definió usando métricas derivadas de las distribución de atributos $p.distance_i$ y $p.headingdiff_i$ como se explica en las *secciones 4.2.1.1* y *4.2.1.3*. Los resultados de este experimento para cada algoritmo se describen a continuación.

4.2.1.1. Algoritmo basado en densidad

Se estableció primero el valor para *eps1*. La Figura 4.3 muestra densidad de $p.distance_i$ (distancia punto a punto) en los puntos que están en el rango *[0 a 100]* metros de los POIs casa y trabajo. Se determinó que *eps1* es igual a *2.5 metros* ya que más del 80% de las distancias son menores o iguales a ese valor.

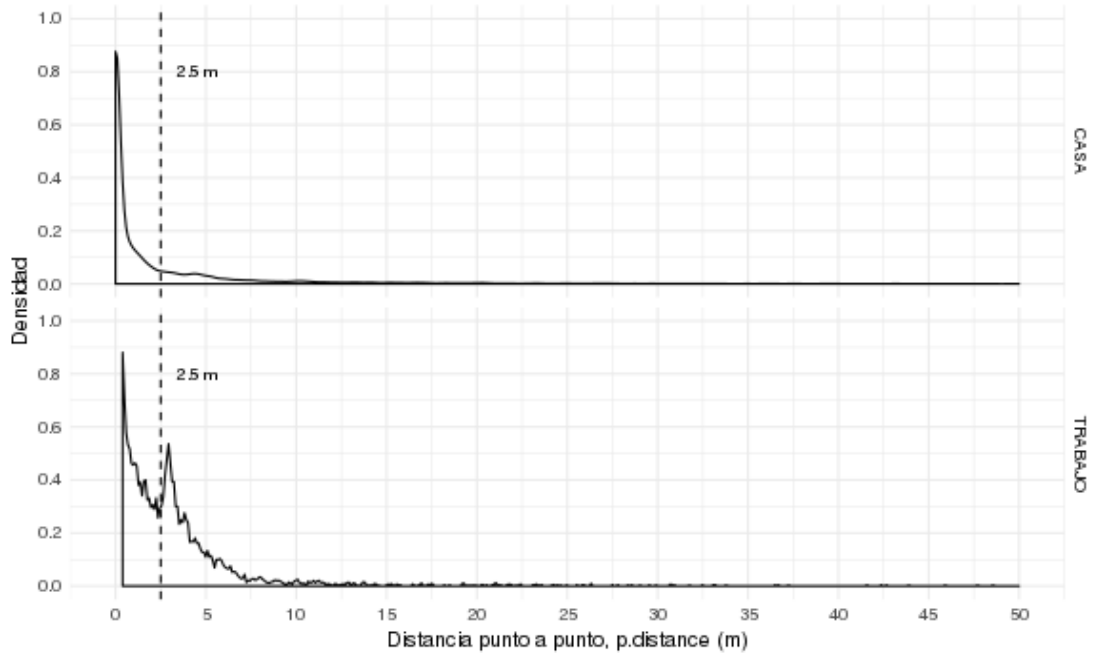


Figura 4.3.- Densidad de p.distance para puntos cercanos a los POIs

Luego, se usó el TPR para determinar el valor de *min_pts1*, encontrando que 5 puntos era el mejor valor para la detección de PUs que corresponden POIs casa y trabajo, *ver Figura 4.4*.

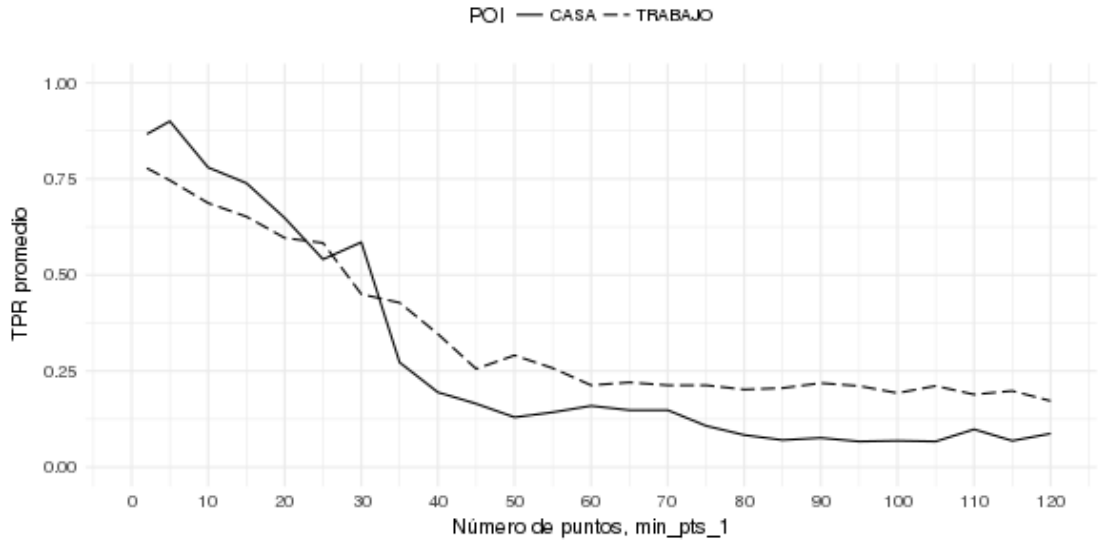


Figura 4.4.- Rendimiento de parámetro min_pts en algoritmo basado en densidad.

Luego, se evaluaron los parámetros establecidos *(eps=2.5m, min_pts=5)* sobre el subconjunto de prueba (70% de sujetos restantes) y se obtuvo los TPR mostrados en la *Tabla 4.3* los cuales revelaron que en *un viaje* se puede detectar en promedio el 71% de PUs que corresponden a POIs casa y 57% de PUs a POIs trabajo. Las tasas de detección para PUs que corresponden a los POIs casa son más altas que los PUs que corresponden a los POIs trabajo y estas se equiparan a medida que el radio aumenta ya que generalmente el área de los lugares de trabajo es más grande que el área de las casas.

Tabla 4.3.- Evaluación de parámetros en algoritmo basado en densidad

Radio (metros)	**TPR**	
	CASA	**TRABAJO**
50	0.62	0.37
100	0.75	0.61
150	0.75	0.73
Promedio	***0.71***	***0.57***

4.2.1.2. Algoritmo basado en distancia-tiempo

Al evaluar las combinaciones con diferentes valores para los parámetros *timeTh* y *distTh* se determinó que *distTh*=30 metros y *timeTh*=15 segundos son los mejores umbrales para este algoritmo ya que representan las mayores tasas para la detección de PUs que corresponden a POIs casa y trabajo como lo ilustra la *Figura 4.5*.

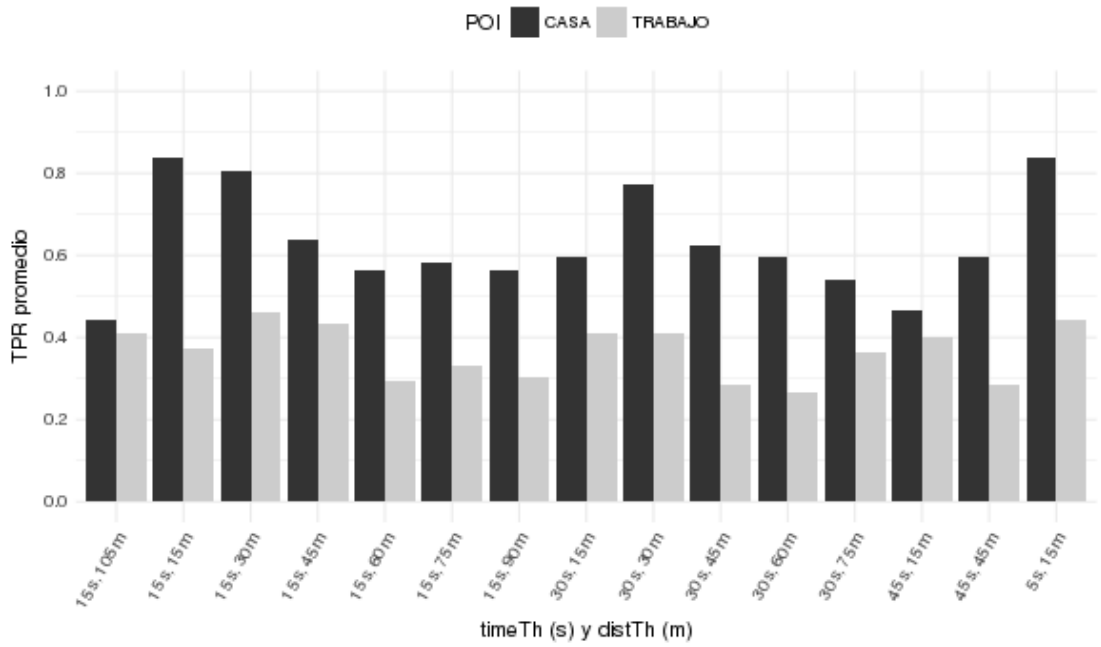


Figura 4.5.- Rendimiento de parámetros timeTh y distTh en algoritmo basado en distancia-tiempo.

Posteriormente, se evaluó los parámetros establecidos *(timeTh = 15s, distTh = 30m)* sobre el conjunto de prueba se encontró que en *un viaje* se puede detectar hasta un 31% de PUs que corresponden a POIs casa y 41% de PUs que corresponden a POIs trabajo.

Tabla 4.4.- Evaluación de parámetros en algoritmo basado en distancia-tiempo.

Radio (metros)	**TPR**	
	CASA	**TRABAJO**
50	0.22	0.25
100	0.22	0.33
150	0.5	0.64
Promedio	***0.31***	***0.41***

4.2.1.3. Algoritmo basado en velocidad

Para identificar los parámetros *Dth* y *Sth* se calculó la desviación estándar del cambio de dirección *p.headingdiff*ᵢ en los *segmentos estáticos* y los *segmentos con poco movimiento* con el objeto de identificar aquellos que representan paradas de los desplazamientos. En las paradas los movimientos son nulos o pequeños con cambios de dirección relativamente grandes; esto ocurre incluso cuando el sujeto está en reposo debido al ruido de la señal GPS, mientras que durante los desplazamientos largos la variación del cambio de dirección es pequeña.

Dth y *Sth* fueron establecidos como el mínimo local en el intervalo de los dos picos más altos de la distribución de la desviación estándar del cambio de dirección. *Dth=35.9°* para el intervalo *[10°, 70°]* en *segmentos con poco movimiento* y *Sth= 18.9°* para el intervalo *[0°, 40°]* en *segmentos estáticos*, ver Figura 4.6.

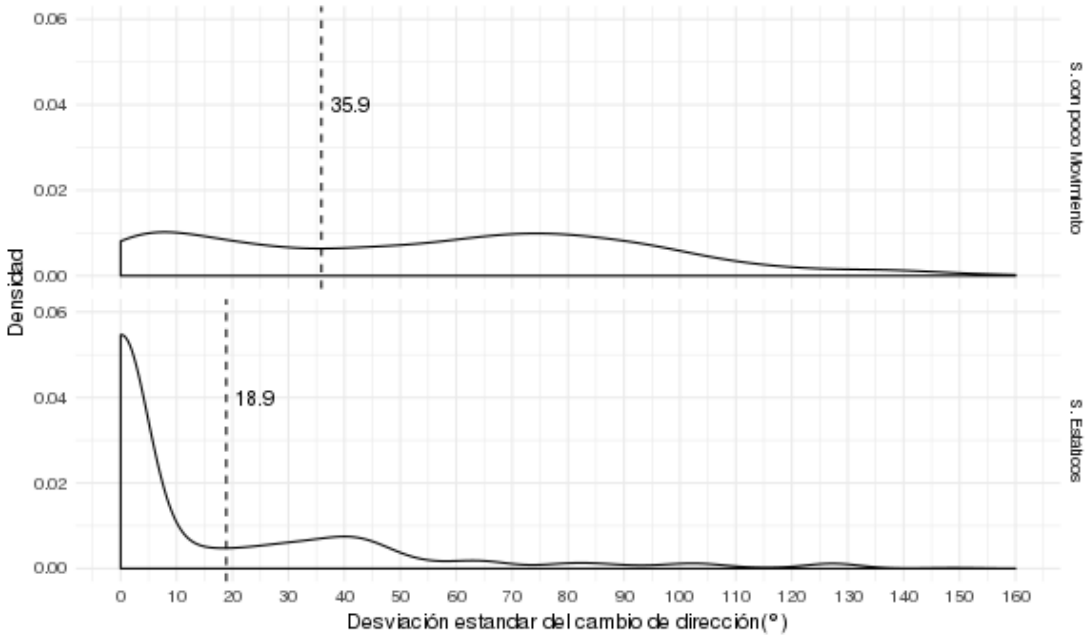


Figura 4.6.- Desviación estándar de cambio de dirección de segmentos.

En cuanto al parámetro *timeTh*, la duración mínima de una parada, los autores en [7] establecieron un *timeTh=1 min(60 s)* ya que estas paradas no son significativas y por tanto se descartan. Sin embargo; en este *dataset* se evidenció que a medida que el *timeTh* se incrementa, el TPR disminuye mientras que al reducir el *timeTh* el TPR aumenta como se ilustra en la *Figura 4.7* porque las paradas registradas por los sujetos en este *dataset* poseen duraciones cortas ya que ellos reportaron los viajes al llegar o

partir de las POIs. Por tanto, se observó que el número de paradas promedio obtenidas con el algoritmo, tanto para las paradas en casa, trabajo y otros, se encuentran con un *timeTh* de *[0s-60s]* como lo ilustra la *Figura 4.8*.

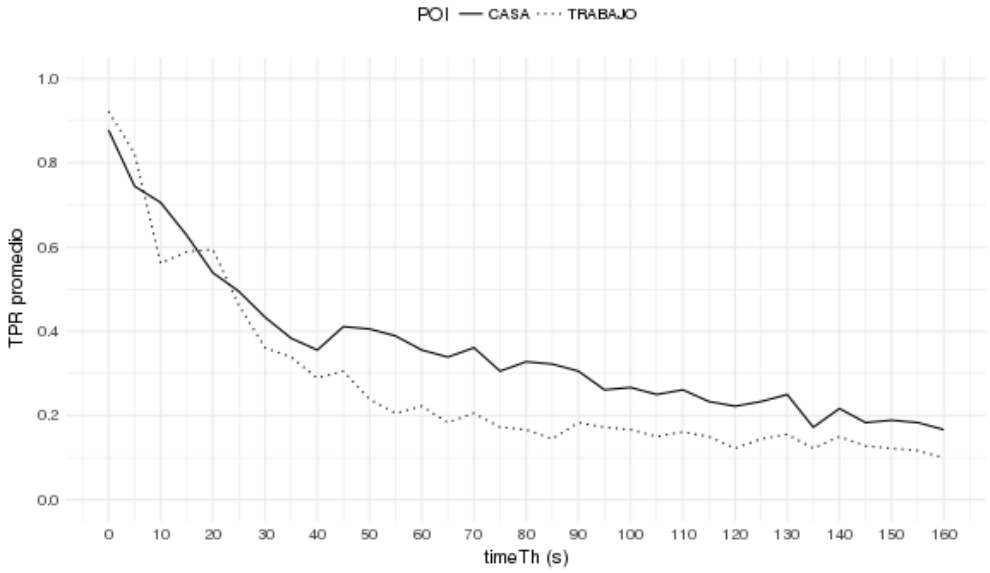


Figura 4.7.- Rendimiento de parámetro timeTh en algoritmo basado en velocidad.

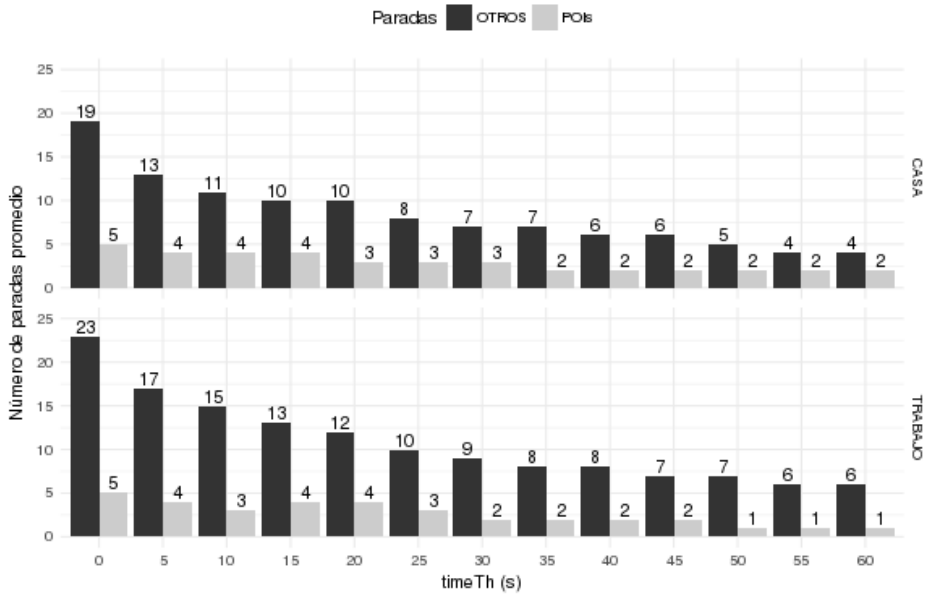


Figura 4.8.- PUs detectados con timeTh en algoritmo basado en velocidad.

De los 6 POIs casas y 6 trabajo, se evidencio que a medida que *timeTh* disminuye el número promedio de paradas aumenta por lo que se estableció que *timeTh=15s* es el más apropiado ya que se puede detectar la mayoría de POIs de interés en casa y trabajo con una cantidad inferior de otros tipos de POIs, lo que disminuiría la cantidad de datos a procesar en unos 50% de paradas en otros tipos de POIs. No se estableció un valor inferior dado que entre más pequeños sean estos umbrales; mayor será la cantidad de paradas cortas que se extraen las cuales no se relevantes para este estudio.

Luego, al evaluar los parámetros escogidos *(Dth=35.9°, Sth=18.9°, timeTh=15s)* sobre el conjunto de prueba en *un viaje* se encontró que con este algoritmo se puede determinar hasta un 53% de los PUs que corresponden a POIs trabajo y un 58% que representan PUs casa, ver Tabla 4.5.

Tabla 4.5.- Evaluación de parámetros en algoritmo basado en velocidad.

Radio (metros)	**TPR**	
	CASA	**TRABAJO**
50	0.44	0.51
100	0.55	0.61
150	0.59	0.62
Promedio	***0.53***	***0.58***

En resumen, las tasas de detección en *un viaje* son inferiores al 80% porque en un solo viaje no siempre aparecen las paradas con las características requeridas por los algoritmos mencionados. Cabe mencionar que *datasets* de los 22 sujetos que fue usado para este experimento tenía como mínimo 5 viajes (ver Tabla 3.7) y por ello estos parámetros podrían variar en un *datasets* de distinta naturaleza o más grande; como lo evidencian los autores de los algoritmos de detección de PUs en [7][25][52].

La hipótesis planteada en el siguiente experimento propone incrementar el número de viajes con el objeto de aumentar la cantidad de información y consecuentemente obtener tasas de detección de PUs más altas.

4.2.2. Selección del Número de Viajes Óptimo.

Se seleccionaron los sujetos con más de 8 viajes registrados en el *dataset II* y se dividieron en dos subconjuntos: el primero con 13 sujetos y 314 viajes que reportaron POIs casa y el segundo con 11 sujetos y 157 viajes que reportaron POIs trabajo. Posteriormente, se aplicó cada algoritmo de detección de paradas con los parámetros definidos en la *Sección 4.2.1* de forma iterativa, incrementando también la cantidad de

viajes de 1 a 8 y seleccionándolos de forma aleatoria para radios de 50m, 100m y 150m. La ejecución de cada algoritmo se repitió 30 veces sobre cada subconjunto y se calculó el TPR en cada ocasión. Finalmente, se calculó y se reportó el TPR promedio para cada radio.

4.2.2.1. Algoritmo basado en densidad

Al incrementar el número de viajes, se puede detectar hasta un 80% y 79% PUs que corresponden a POIs casa y trabajo con un mínimo de 2 y 6 viajes dentro de un radio de 50 metros, ver Figura 4.9. Mientras que, en un radio de 100 m se puede detectar hasta 90% y 80% de LP con un mínimo de 2 y 4 viajes respectivamente; y, en un radio de150 m esos porcentajes eran más similares.

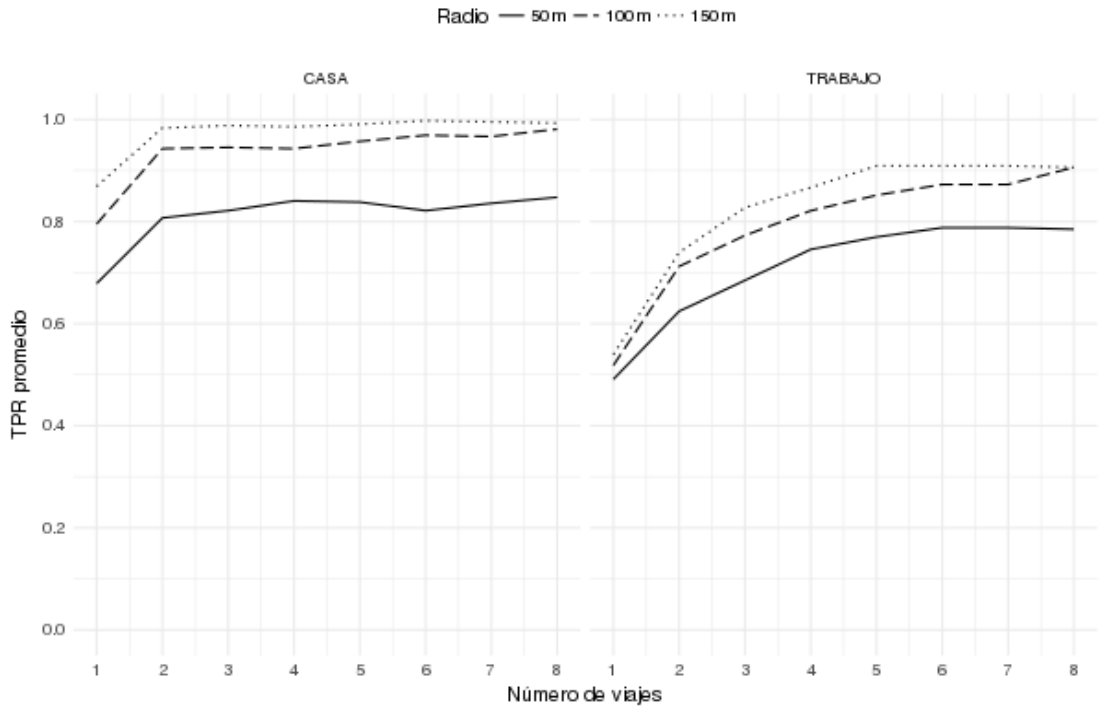


Figura 4.9.- Evaluación de viajes con algoritmo basado en densidad

Las *paradas* de cada viaje se identifican para luego ser agrupadas si están a menos de *100 metros* para representar a un PUs casa o trabajo. Sin embargo, si este enfoque fuese aplicado sobre múltiples trayectorias a la vez, algunas regiones sin significados semánticos como cruces de carreteras y avenidas donde un sujeto pasa iterativamente [25][52] podrían ser extraídas lo que disminuiría significativamente las tasas de detección de PUs.

Los resultados de la *Figura 4.8* demuestran que la detección de PUs considerando las regiones densas es posible sin considerar la información temporal y presenta mejores resultados para la identificación de los PUs que corresponden a casa que a los de trabajo; esto probablemente debido a que el área de las casas es mucho menor a la de los trabajos y en consecuencia la permanencia de un sujeto en casa registra una región densa con los puntos GPS que puede ser identificada a partir de algoritmos de densidad espacial. La aplicación de este enfoque en viajes largos posee un coste computacional alto ya que la identificación de regiones densas requiere el cálculo de una matriz de distancias geodésicas que involucra a todos los puntos GPS de los viajes a evaluar.

4.2.2.2. Algoritmo basado en distancia tiempo

Al incrementar el número de viajes se evidenció que este algoritmo permite la detección de hasta un 55% y 45% de PUs que corresponden a POIs casa y trabajo con un máximo de 2 y 3 viajes en un radio de 50m como lo ilustra la *Figura 4.10*. Estos porcentajes mejoraron dentro de una distancia de 100m a 80% y 75% con un máximo de 2 y 4 viajes y dentro de una distancia de 150m a 90% y 80% con un máximo de 2 y 6 viajes.

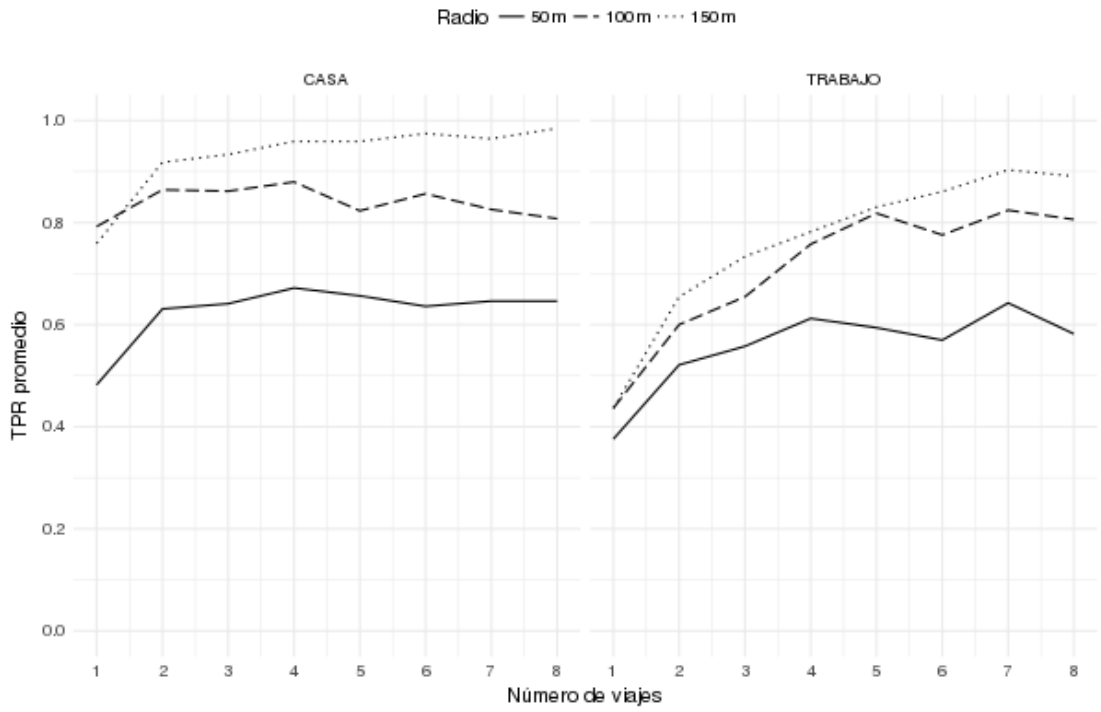


Figura 4.10.- Evaluación de viajes en algoritmo basado en distancia-tiempo

Los resultados muestran que no se puede identificar paradas, con este algoritmo, en la mayoría de los viajes ya que no es posible identificar la región espacial en donde el sujeto permanece un periodo de tiempo en base a umbrales tan pequeños; por lo tanto, la tasa de detección es errática debido al error GPS y consecuentemente los resultados no mejoran aun cuando se incremente el número de viajes.

4.2.2.3. Algoritmo basado en velocidad

Este algoritmo permite la detección de hasta el 80% de PUs que corresponden a POIs casa con un máximo de 3 viajes dentro de un radio de 50m alrededor de los POIs como lo demuestra la *Figura 4.11*. En un radio de 100m, detectó hasta 90% de PU dentro de un máximo y estos porcentajes no cambiaron en un radio de 150 m.

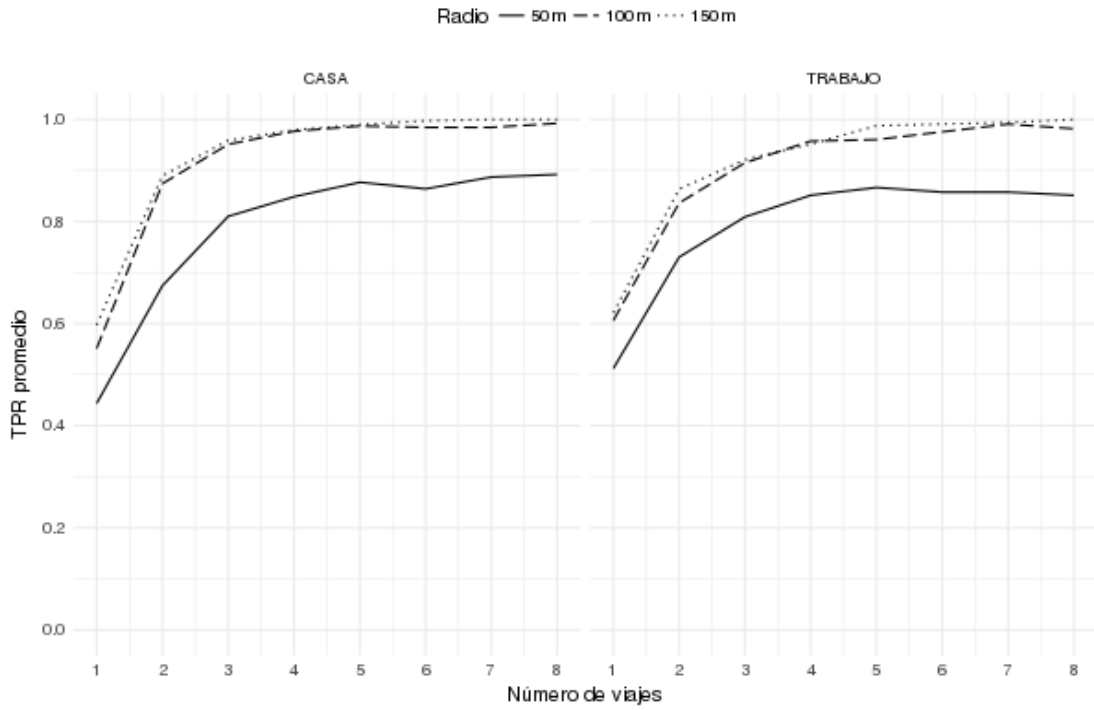


Figura 4.11.- Evaluación de viajes en algoritmo basado en velocidad.

La extracción de segmentos donde el sujeto permanece estático o se desplaza con poco movimiento permitió detectar los PUs casa y trabajo de forma efectiva en un radio de 100m lo que evidencia que al acercarse a un POI los sujetos tienden a caminar o a permanecer en un punto fijo dejando un rastro en los puntos GPS que permiten la identificación de los PUs.

Al comparar los resultados obtenidos con los tres algoritmos para múltiples viajes, *Tabla 4.6*, se encontró que el algoritmo basado en *velocidad* arroja las mejores tasas de detección de PUs tanto en casa como en trabajo en un radio de 100m; dado que al incrementar el número de viajes de partida o llegada, la posibilidad de detectar los PUs a partir de los puntos GPS mejoró considerablemente y se estableció con un promedio de tres viajes se puede detectar hasta el 90% de PUs. Por otro lado, a pesar de que el algoritmo basado en *densidad* presentó también buenos resultados; este no es mejor que el algoritmo basado en velocidad ya que es más costoso computacionalmente y requiere un mayor número de viajes para detectar los PUs en trabajo.

Tabla 4.6.- Resultados obtenidos con los tres algoritmos de detección de PUs.

Algoritmo basado en:	**CASA**		**TRABAJO**	
	Max. TPR	**No. Viajes**	**Max. TPR**	**No. Viajes**
Densidad	0.94	2	0.82	4
Distancia-Tiempo	0.88	2	0.75	4
Velocidad	0.95	3	0.91	3

Es importante mencionar que en este capítulo únicamente se detecta la ubicación de casa y trabajo. En el siguiente capítulo se realiza la clasificación PUs en POIs a partir de características específicas y aprendizaje supervisado.

CAPÍTULO 5

5. Clasificación de Puntos de Ubicación

La clasificación tiene por objeto construir un modelo que capture las características intrínsecas de la clase (POIs) para poder predecir la clase de un PU desconocido [30]. La asignación semántica automática de PUs se ha realizado en trabajos previos en base a características temporales requieren de al menos una semana de recolección de viajes a días consecutivos para la visualización de patrones [1] o para establecer un conjunto de probabilidades en una red bayesiana [28]. Nuestro *dataset* de viajes no tiene una resolución temporal alta y por ende se buscarán técnicas alternativas.

En este trabajo se evaluarán tres métodos de aprendizaje supervisado (*árbol de decisión*, *máquinas de soporte*, *árboles aleatorios*) para la clasificación automática de PUs en POIs casa o trabajo. Este capítulo se organiza de la siguiente manera: en la *sección 5.1* se describe el proceso aplicado para entrenar el clasificador con los tres algoritmos, en la *sección 5.2* se detalla el experimento ejecutado y finalmente en la *sección 5.3* se presentan los resultados obtenidos.

5.1. Metodología

La clasificación de PUs se realizó en 4 etapas: 1) extracción de características de PUs, 2) preparación de PU, 3) aplicación del algoritmo de clasificación y 4) evaluación.

5.1.1. Caracterización de PUs

Para extraer las características de los PUs, se calcula un conjunto de atributos para los *puntos de parada (C)* considerando sus *segmentos de paradas (w)* que contienen puntos GPS con la información de los viajes de entrada y salida del sujeto a la *parada (L)*. Estos atributos son:

Flujo (*C.M*): es un valor binario [0, 1] que describe el flujo de los viajes de entrada y salida de una parada. El valor del flujo depende de la posición relativa de la *parada* con respecto al primer y último punto GPS del viaje. El flujo es 0 si el *segmento de parada* está cerca (ha recorrido menos del 50% del viaje) al primer punto (p_1) de *un viaje* y 1 si la parada está cerca (ha recorrido más del 50% del viaje) del último punto (p_n) de un

viaje. Por ejemplo, en la *Figura 5.1*, la *parada* está más cerca al primer punto del viaje (p_a) y describe a un viaje de llegada; por ende, *C.M = 0*.

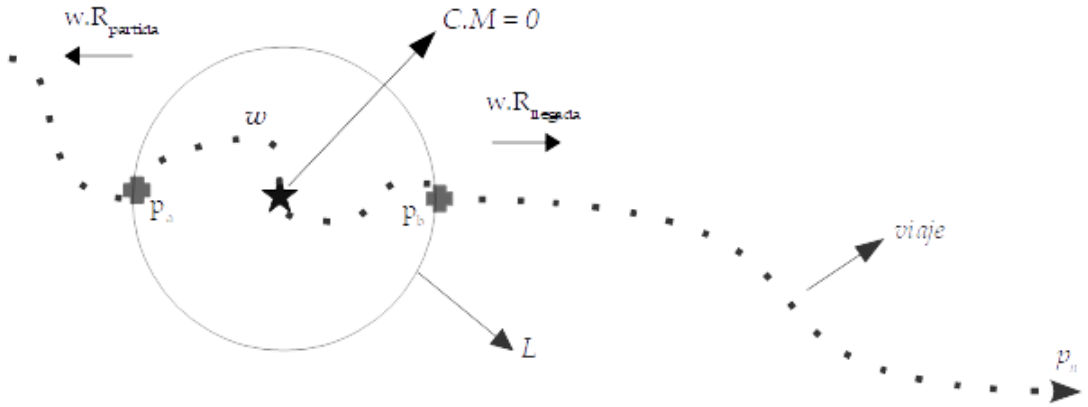


Figura 5.1.- Flujo en un segmento de parada.

Para determinar si la *parada* está más cerca del inicio o fin del viaje se calcula el recorrido desde la parada al primer y último puntos GPS del viaje tal como se ilustra en la *Figura 5.1* Los recorridos $w.R_{partida}$ y $w.R_{llegada}$ se calculan sumando las distancias punto a punto ($p.distance_i$) desde cada extremo del viaje al primer (p_a) y último punto (p_b) del *segmento de parada* (*w)* de la siguiente forma:

$$w.R_{partida} = \sum_{i=a}^{1} p.distance_i$$

$$w.R_{llegada} = \sum_{i=b}^{n} p.distance_i$$

El flujo de la parada *C.M* se define de acuerdo a las siguientes reglas:

$$\textbf{\textit{Si}}\ w.R_{partida} < w.R_{llegada} \implies C.M = 0$$

$$\textbf{\textit{Si}}\ w.R_{partida} > w.R_{llegada} \implies C.M = 1$$

Fecha (C.*date*): es la fecha [dd/mm/yy] en que se registró el primer punto del segmento de parada *w*.

Hora (*C.h*): es un entero [1-24] tomado a partir de la hora registrada en el primer punto del segmento de parada.

$$C.h = \frac{\sum_{i=1}^{m} trunk(p.time_i)}{m}$$

Tamaño (*C.distance*): es la distancia calculada con el valor máximo y mínimo de la latitud y longitud.

$$C.distance = distance([max(p.x), max(p.y)], [\min(p.x), min(p.y)])$$

Duración (C.*duration*): es el tiempo que permanece el sujeto en una parada. Se calcula a partir de la diferencia de tiempo entre el último ($p.time_b$) al primer ($p.time_a$) punto del segmento de la parada.

$$C.duration = p.time_b - p.time_a$$

Velocidad de llegada ($C.speed_{llegada}$).- es el promedio de la velocidad (km/h) del sujeto en los 5 puntos del viaje marcados por el sensor antes de entrar a una parada.

$$C.speed_{llegada} = \frac{\sum_{i=a}^{n} p.speed_i}{|p|} \therefore n = a - 5$$

Velocidad de salida ($C.speed_{partida}$): es el promedio de la velocidad (km/h) del sujeto en los 5 puntos del viaje marcados por el sensor luego de salir de una parada.

$$C.speed_{partida} = \frac{\sum_{i=b}^{n} p.speed_i}{|p|} \therefore n = b + 5$$

Precisión de llegada ($C.speed_{llegada}$).- es el promedio de la precisión GPS del sujeto en los 5 puntos del viaje marcados por el sensor antes de entrar a una parada.

$$C.accuracy_{llegada} = \frac{\sum_{i=a}^{n} p.accuracy_i}{|p|} \therefore n = a - 5$$

Precisión de salida ($C.speed_{partida}$): es el promedio de la precisión GPS del sujeto en los 5 puntos del viaje marcados por el sensor luego de salir de una parada.

$$C.accuracy_{partida} = \frac{\sum_{i=b}^{n} p.accuracy_i}{|p|} \therefore n = b + 5$$

En base a las propiedades descritas anteriormente y a los atributos (*sección 3.1.1*) definidos para puntos GPS se deriva un conjunto de características específicas para los *puntos de ubicación (PUs)*. Debemos recordar que un PU es el *centroide* de los *puntos*

de parada (C) que a su vez son los *centroides* de los puntos GPS de las *paradas (L)* (*ver sección 4.1*). Las características de los *PUs* se clasifican en tres tipos:

5.1.1.1. Derivadas de información geoespacial

Velocidad (*U.speed*): se calcula con el promedio de las velocidades ($p.speed_i$) reportadas en los puntos GPS de todas las *paradas* asociadas a un PU.

$$U.speed_p = \overline{p.speed} = \frac{\sum_{i=1}^{n} p.speed_i}{|p|}$$

Dispersión de la velocidad (*U.speed.sd*): es la desviación estándar de las velocidades ($p.speed_i$) reportadas en los puntos GPS de todas las *paradas* asociadas a un PU.

$$U.speed.sd_p = \sigma(p.speed) = \sqrt{\frac{\sum_{i=1}^{n}(p.speed_i - \overline{p.speed})^2}{|p|}}$$

Dispersión de la distancia entre puntos consecutivos (*U.distance.sd*): es la desviación estándar de la distancia punto a punto ($p.distance_i$) de los puntos GPS consecutivos de todas las *paradas* asociadas a un PU.

$$U.distance.sd_p = \sigma(p.distance) = \sqrt{\frac{\sum_{i=1}^{n}(p.distance_i - \overline{p.distance})^2}{|p|}}$$

Dispersión de la precisión (*U.accuracy.sd*): es la desviación estándar de la precisión ($p.accuracy_i$) reportada por los puntos GPS de todas las *paradas* asociadas a un PU.

$$U.accuracy.sd_p = \sigma(p.accuracy) = \sqrt{\frac{\sum_{i=1}^{n}(p.accuracy_i - \overline{p.accuracy})^2}{|p|}}$$

Tamaño (*U.distance*).- es la distancia geodésica de las longitudes ($p.x_i$) y latitudes ($p.y_i$) máxima y mínima de los puntos GPS de todas las *paradas* asociadas a un PU.

$$U.distance_p = distance([max(p.x), max(p.y)], [\min(p.x), min(p.y)])$$

5.1.1.2. Derivadas de información temporal

Puntos en días ordinarios (*U.weekdays*): porcentaje de puntos GPS de todas las *paradas* asociadas a un PU registrados en días ordinarios [L,…,V].

$$U.weekdays_p = \frac{card(p.date\ \epsilon\{L,..,V\})}{|p|}$$

Puntos en fines de semana (*U.weekends*): porcentaje de puntos GPS de todas las *paradas* asociadas a un PU registrados en fines de semana [S, D].

$$U.weekends_p = \frac{card(p.date\ \epsilon\{S,D\})}{|p|}$$

Hora más frecuente (*U.h*): es la hora [1-24] en que se registra el mayor porcentaje de puntos GPS en el PU.

$$U.h_p = \arg\max_x \quad U.H(x)$$

Donde el porcentaje de puntos durante cada una de las 24 horas del día se calcula de la siguiente manera.

$$U.H(x) = \frac{card(\{trunk\ (p.hora) = x\}\ )}{|p|} \therefore x\ \epsilon\ \{1,2,3,4,\ldots\ldots 24\}$$

Puntos en días sábado (*U.Sat*): es el porcentaje de puntos GPS registrados en los días sábados (S).

$$U.Sat_p = \frac{card(p.date\ \epsilon\{S\})}{|p|}$$

Puntos en días domingos (*U.Sun*): es el porcentaje de puntos GPS registrados en los días domingos (D).

$$U.Sun_p = \frac{card(p.date\ \epsilon\{D\})}{|p|}$$

Flujo en días consecutivos (*U.M.cd*): se calcula el promedio del flujo (C.*M*) de las paradas registradas en días consecutivos.

$$U.M_{cd} = \overline{C.M} = \frac{\sum_{i=1}^{n} C.M_i}{|p|}$$

Duración en días consecutivos (*U.duration*): se calcula el promedio de la duración (*L.duration*) de las paradas registradas en días consecutivos.

$$U.duration_{cd} = \overline{C.duration} = \frac{\sum_{i=1}^{n} C.duration_i}{|p|}$$

5.1.1.3. Derivadas de información de viajes de entrada y salida

Flujo promedio (*U.M.t*): es un valor de 0 a 1 que se calcula con el promedio del flujo (C.M_i) de las paradas asociadas a un PU. Donde $U.M = 0.5$ indica que la cantidad de viajes de entrada y salida es igual; $U.M < 0.5$ indica que el punto de ubicación fue definido por más viajes de entrada que de salida; y viceversa si $U.M > 0.5$.

$$U.M_t = \overline{C.M} = \frac{\sum_{i=1}^{n} C.M_i}{|p|}$$

Duración promedio (*U.duration.t*): es el promedio de la duración cada parada (*L.duration*$_i$) que define a un PU.

$$U.duration_t = \overline{C.duration} = \frac{\sum_{i=0}^{n} C.duration_i}{|p|}$$

Tamaño promedio (*U.distance.t*): es el promedio de los tamaños de las paradas (*L.distance*$_i$) que determinan un PU.

$$U.distance_t = \overline{C.distance} = \frac{\sum_{i=0}^{n} C.distance_i}{|p|}$$

Promedio de la dispersión de la velocidad (*U.speed.sd.t*) es el promedio de la dispersión de la velocidad (*C.speed.sd*$_i$) de las paradas que determinan un PU.

$$U.speed.sd_t = \overline{C.speed.sd} = \frac{\sum_{i=0}^{n} C.speed.sd_i}{|p|}$$

Promedio de la dispersión de la distancia (*U.distance.sd.t*) es el promedio de la dispersión de la distancia (*C.distance.sd*$_i$) de las paradas que determinan un PU.

$$U.distance.sd_t = \overline{C.distance.sd} = \frac{\sum_{i=0}^{n} C.distance.sd_i}{|p|}$$

Promedio de la dispersión de la precisión (*U.accuracy.sd*) es el promedio de la dispersión de la dispersión (*C.accuracy.sd*$_i$) de las paradas que determinan un PU.

$$U.accuracy.sd_t = \overline{C.accuracy.sd} = \frac{\sum_{i=0}^{n} C.accuracy.sd_i}{|p|}$$

Hora de entrada promedio (*U.h.t*$_{entrada}$): es el promedio de la hora registrada *C.h* por las paradas con un flujo *C.M==0*.

$$U.h_{t_{entrada}} = \overline{C.h} = \frac{\sum_{i=0}^{n} C.h_i}{|p|} \therefore C.M == 0$$

Hora de salida promedio (*U.h.t*$_{salida}$): es el promedio de la hora registrada *C.h* por las paradas con un flujo *C.M==1*.

$$U.h_{t_{salida}} = \overline{C.h} = \frac{\sum_{i=0}^{n} C.h_i}{|p|} \therefore C.M == 1$$

Velocidad de llegada promedio (*U.speed*$_{llegada}$): es el promedio de la velocidad de llegada (*C.speed*$_i$) del sujeto en los 5 puntos marcados por el sensor antes de que un sujeto llegue a una parada.

$$U.speed_{llegada} = \overline{C.speed_{llegada}} = \frac{\sum_{i=1}^{n} C.speed_{llegada_i}}{|p|}$$

Velocidad de salida promedio (*U.speed*$_{salida}$): es el promedio de la velocidad de salida (*C.speed*$_i$) del sujeto en los 5 puntos marcados por el sensor luego de que un sujeto salga a una parada

$$U.speed_{salida} = \overline{C.speed_{salida}} = \frac{\sum_{i=1}^{n} C.speed_{salida_i}}{|p|}$$

Precisión de llegada promedio (*U.accuracy*$_{llegada}$): es el promedio de precisión ($C.accuracy_{llegada_i}$) del sujeto en los 5 puntos marcados por el sensor antes de que un sujeto llegue a una parada.

$$U.accuracy_{llegada} = \overline{C.accuracy_{llegada}} = \frac{\sum_{i=1}^{n} C.accuracy_{llegada_i}}{|p|}$$

Precisión de salida promedio ($U.accuracy_{salida}$): es el promedio de precisión ($C.accuracy_{salida_i}$) del sujeto en los 5 puntos marcados por el sensor antes de que un sujeto salida a una parada.

$$U.accuracy_{salida} = \overline{C.accuracy_{salida}} = \frac{\sum_{i=1}^{n} C.accuracy_{salida_i}}{|p|}$$

Finalmente, se utilizó la técnica de la imputación de un valor único en las características que no pueden ser calculadas debido a falta de información. Se reemplaza un valor faltante con un valor único arbitrario, en este caso '-1', ya que se satisfacen dos condiciones [37]: el hecho de que falte un valor depende del valor de la variable de clase y ocurre tanto en el *conjunto de entrenamiento* como el *de prueba*.

5.1.2. Preparación de dataset de PUs

Los PUs derivados de paradas cortas o que representan ubicaciones poco probables se removieron en el *conjunto de entrenamiento* y *prueba* si:

- El tamaño del PU es superior a 1km, es decir abarca una región muy grande para una casa o trabajo.
- La velocidad promedio del PU es superior a 7.1km/h. La velocidad de un sujeto cuando se mantiene en un punto fijo es 0 km/h; sin embargo, cuando la persona se desplaza caminando la velocidad media varía alrededor de 4.5km/h [44] y la velocidad máxima es de 7.5km/h [6].
- El tamaño del PU es 0 metros porque pueden ser puntos de ubicación que representan paradas muy cortas como intersecciones, paradas semáforos, etc.
- Si más del 30% de valores de atributos son iguales a *'-1'*. La ausencia de valores en las características de los PUs pueden afectar la aplicación y evaluación del algoritmo de clasificación de forma significativa.
- *Los PUs* duplicados.
- Además, si dos o más atributos están altamente correlacionados *(>=0.6 o <=-0.6)* se selecciona uno de ellos. Para medir la correlación se utilizó el *coeficiente de correlación de Pearson*.

5.1.3. Clasificación

Se utilizaron los siguientes algoritmos de clasificación: *los árboles de decisión*, *los bosques aleatorios* y las *máquinas de soporte*.

Árboles de decisión

Los árboles de decisión proveen una herramienta de clasificación potente y consisten en una serie de reglas organizadas en una estructura jerárquica que culminan en una clase o valor y se consideran como un método de aprendizaje fácil de utilizar y de entender ya que una decisión se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas.

Un árbol de decisión se construye utilizando un procedimiento de partición binaria [8] que divide repetidamente los datos en grupos. En cada división los datos son partidos en dos o más grupos completamente excluyentes utilizando un criterio de impureza que tiene por objeto particionar grupos homogéneos y a la vez mantener un árbol razonablemente pequeño. Existen varios criterios de impureza como el *error esperado*, *gini*, y el *índice de información o entropía* siendo el criterio de *entropía* el más preciso que *gini* y el *error esperado* [18]. El índice de entropía en el nodo t y se define como:

$$Entropía(t) = \sum_{i=1}^{n} -p(i|t)\log_2 p(i|t)$$

Donde $i = 1, \ldots, n$ es el número de clases de la variable categórica y $p(i|t)$ la probabilidad de clasificación correcta para la clase i en el nodo t. El valor máximo de la entropía depende del número de clases y es 0 si todos los elementos están en una clase.

No obstante, cuando el número de variables es grande, el árbol resultante puede contener un número excesivo de nodos por tanto el modelo requiere ser podado para evitar problemas de sobreajuste. En el procedimiento de poda se descartaron sucesivamente ramas o nodos terminales que incrementan muy poco la precisión del árbol hasta encontrar el tamaño "adecuado" del árbol con un procedimiento de validación cruzada para una mejor capacidad de generalización.

Bosques aleatorios

Los bosques aleatorios (RF, *random forest* en inglés) son árboles de decisión que mejoran el rendimiento de los árboles de decisión tradicionales a través de técnicas como *bootstrapping* y *bagging*. Con la técnica del *bootstrapping* generan varios *pseudo-traning* sets con los que ajustan diferentes árboles de decisión y después promedian los resultados de predicción con un proceso conocido como *bagging*. El proceso de *bagging* consigue mejorar la capacidad predictiva del clasificador en comparación a otros clasificadores basados en un único árbol de decisión; sin embargo, esto tiene un coste asociado ya que la *interpretabilidad* del modelo se reduce ya que tras combinar múltiples árboles ya no es posible obtener una representación gráfica sencilla del clasificador

En otras palabras, en algoritmo RF crea varios árboles que se ajustan de forma repetida empleando muestras generadas por *boostrapping*; en cada muestra se ajusta el clasificador con observaciones y atributos aleatorios donde se usa aproximadamente dos tercios de las observaciones originales también denominado *in-bag*; mientras que, el tercio restante, también denominado *out-of-bag (OOB)*, se utiliza para obtener el *OOB-classification error* que es equivalente a la *tasa de error* [20].

Las muestras OOB son también usadas por los RF para calcular la fuerza de predicción del clasificador; el cual está condicionado a su interacción con el resto de variables con las métricas: *MDG (Mean Decrease Gini)* y *MDA (Mean decrease accuracy)*. El índice MDG se utiliza para medir la homogeneidad que nos proporciona cada atributo a los nodos de los diferentes árboles del RF; es decir, la capacidad de una variable para dividir los datos en particiones lo más puras posibles. Y el índice MDA se utiliza para medir el impacto que tiene cada atributo en la precisión del modelo; esta métrica permite visualizar el impacto relativo que tiene el rendimiento del clasificador al sustraer un atributo en concreto.

Máquinas de soporte

Las máquinas de soporte vectorial [10] son algoritmos de clasificación lineal insensibles al sobre entrenamiento y con la capacidad de tratar datos con gran dimensionalidad. Este clasificador utiliza un hiperplano para separar los puntos de dos clases distintas en un espacio d-dimensional y un kernel que puede modificar la forma en que el hiperplano separa los datos a partir de un producto escalar. El mejor hiperplano queda a mayor distancia de los elementos a separar.

La función *kernel* Φ puede ser lineal, *polinómica, radial o sigmoidal*; sin embargo, para efectos de este trabajo se utilizó la *función kernel polinómica* porque el error estándar relativo calculado con un proceso de validación cruzada es menor (ver *sección 5.2*). La *función kernel polinómica* está dada por la ecuación:

$$\Phi(u, v) = \gamma(u^T v + c_o)^d \therefore \gamma \geq 0$$

Donde u y v son vectores de características calculadas a partir de muestras de entrenamiento o de prueba, c_o es una constante que manipula la influencia de los términos de orden superior frente a los de orden inferior en el polinomio, d permite ajustar el orden polinómico y γ es el margen geométrico o distancia entre los ejemplos de clases.

5.1.4. Evaluación

Para evaluar los algoritmos de clasificación se estableció un conjunto de entrenamiento (30% de los sujetos), otro de prueba (70% de los sujetos) del *dataset* de PUs y se utilizó una matriz de confusión (*Tabla 5.1*) con las métricas: *accuracy*, *precision*, *recall* y *F-score* para la evaluación del algoritmo.

Tabla 5.1.- Matriz de confusión para evaluación de modelos de clasificación

		PREDICCIÓN	
		Positiva	*Negativa*
CLASE	*Positiva*	**TP**	**FN (***Error Tipo II)*
	Negativa	**FP (***Error Tipo I)*	**TN**

Donde *TP*, *TN*, *FP* y *FN* corresponde a las siguientes definiciones considerando que los PUs corresponden a una única *clase (G)* de POIs casa o trabajo.

— *True Positives (TP):* son PUs pertenecientes a la clase *G* que se clasifican correctamente en la clase *G*.

— *True Negatives (TN):* son PUs no pertenecientes a la clase *G* y que no se clasifican como clase *G*.

— *False Positives (FP):* son PUs no pertenecientes a la clase *G* pero que se clasifican como clase *G*.

— *False Negatives (FN):* son PUs pertenecientes a la clase *G* pero que no se clasifican como clase *G*.

Así mismo, con las definiciones establecidas se calcularon las métricas de *accuracy, tasa de error, precisión, recall y F-score*.

Exactitud (*accuracy* en inglés): porcentaje de PUs clasificados correctamente.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{P + N}$$

Tasa de Error (*error rate* en inglés): el número de todas las predicciones incorrectas dividido por el número total del conjunto de datos.

$$ERR = 1 - Accuracy = \frac{FP + FN}{TP + TN + FP + FN} = \frac{FP + FN}{P + N}$$

Precisión (*precision* en inglés): indica la proporción de PUs que se clasificaron como de la clase G correspondiente y que en realidad son de esa clase con respecto al total de PUs.

$$Precision = \frac{TP}{TP + FP}$$

Exhaustividad (*recall* en inglés): indica la proporción de PUs pertenecientes a G que fueron clasificados correctamente.

$$Recall = \frac{TP}{TP + FN}$$

Valor-F (*F-score* en inglés): es una media armónica que combina los valores de *precisión* y *exhaustividad*.

$$F - score = 2 \times \frac{precision \times recall}{precision + recall}$$

Las cuatro métricas oscilan entre los valores 0 y 1, siendo el 1 el valor máximo posible, y 0 el mínimo.

5.2. Experimento

Para entrenar el clasificador se trabajó con un *dataset* de PUs detectados en base a la metodología descrita en la sección anterior con el algoritmo de *velocidad* a los cuales se les asignó una etiqueta de "casa", "trabajo", o "no definido" de acuerdo a su proximidad con los POIs reportados por los sujetos para lo cual se usó un radio de 100 metros tal como se describe en la *sección 4.1.1.* En el *dataset* se descartó los PUs con la etiqueta *"no definido"* y se consideró únicamente los PUs con las etiquetas de casa o trabajo de 20 sujetos que reportaron casa y 12 sujetos que reportaron trabajo. Se consideraron únicamente los PUs de aquellos sujetos que registraron hasta 5 viajes con origen o destino a los POIs casa o trabajo respectivamente.

Tabla 5.2.- Conjunto de entrenamiento y prueba para la clasificación

POIs	**Total**		**Entrenamiento (30%)**			***Prueba* (70%)**		
	Sujetos	*Viajes*	*Sujetos*	*Viajes*	*PUs*	*Sujetos*	*Viajes*	*PUs*
Casa	20	352	6	96	264	14	256	384
Trabajo	12	148	4	42	185	8	106	222

Los algoritmos de clasificación se entrenaron por separado considerando un *conjunto de entrenamiento* de 449 PUs etiquetados como casa o trabajo; los cuales fueron generados a partir de un conjunto de 6 sujetos con 96 viajes que reportaron casa y 4 con 42 viajes que reportaron trabajo. En total, se generó un *dataset* con 268 PUs a partir de 96 viajes con origen o destino casa y 197 PUs a partir de 42 viajes con origen o destino trabajo. Los PUs se extrajeron usando una cantidad variable de viajes 1 a 5 por cada sujeto y el proceso se repitió 30 veces seleccionado viajes aleatorios. La *Tabla 5.2* describe la cantidad de los sujetos, PUs, POIs, y viajes utilizados para generar el conjunto de entrenamiento y la *tabla 5.3* muestra la cantidad de PUs del conjunto de entrenamiento obtenidos al variar los viajes del 1 al 5.

Tabla 5.3.- PUs y número de viajes del conjunto de entrenamiento.

Viajes	**Entrenamiento (30%)**			***Prueba* (70%)**		
	PUs	**Casa**	**Trabajo**	**PUs**	**Casa**	**Trabajo**
1	66	40	26	132	84	48
2	113	73	40	132	84	48
3	99	57	39	126	78	48
4	92	51	41	120	78	42
5	82	43	39	96	60	36
Total	**449**	**264**	**185**	**606**	**384**	**222**

Sobre el conjunto de entrenamiento se calculó la matriz de coeficientes de *correlación de Pearson* en la *Figura 5.2* para identificar correlaciones lineales entre los distintos atributos y se descartó los siguientes: *U.speed.p, U.weekday.p, U.weekend.p, U.Saturday.p, U.Sunday.p, U.distance.sd.t, U.duration.cd, U.distance.p, U.duration.t, U.accuracy.sd.t, U.M.cd, U.speed.sd.t* porque estaban altamente correlacionadas con otros atributos y por tanto son redundantes. Los atributos redundantes tienen un impacto negativo en el rendimiento de los algoritmos de aprendizaje automático [46]. El resto de atributos se conservaron y fueron utilizados para entrenar el clasificador.

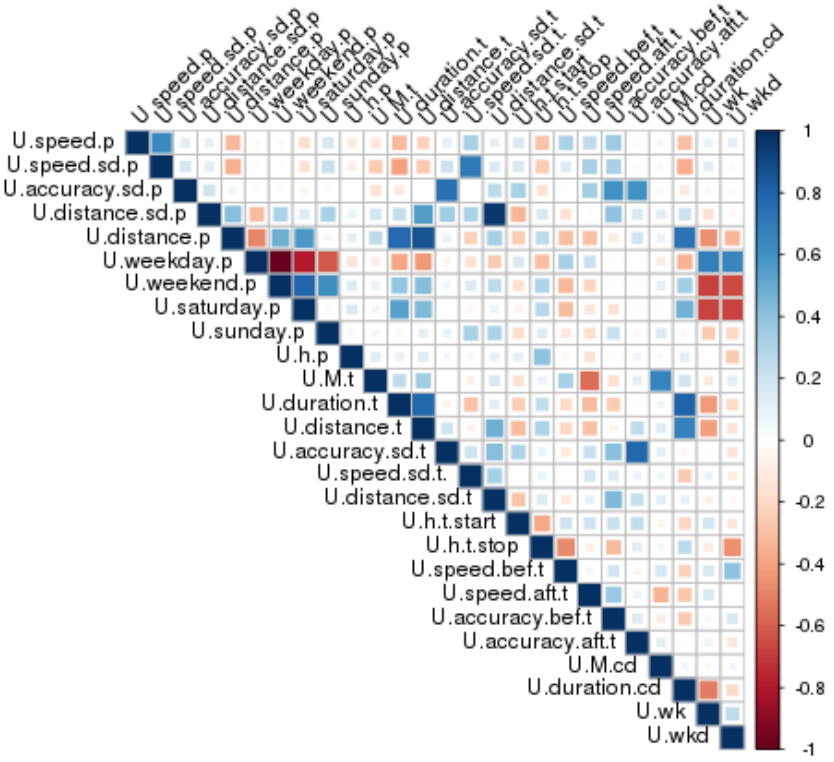


Figura 5.2.- Matriz de correlación de atributos de PUs

La evaluación de los algoritmos de clasificación se construyó un conjunto de prueba con seis *datasets* de 606 PUs obtenidos sobre el 70% de sujetos restantes con 384 viajes para los PUs casa y 222 para los PUs trabajo, ver *Tabla 5.2 y Tabla 5.3*. Cada uno de los seis *dataset*s posee 101 PUs casa y trabajo, los cuales fueron obtenidos variando el número de viajes del 1 al 5 tal como se muestra en la *Tabla 5.5*.

Tabla 5.4.- Datasets de PUs para evaluación de algoritmos de clasificación

Viajes	PUs	Casa	Trabajo
1	22	14	8
2	22	14	8
3	21	13	8
4	20	13	7
5	16	10	6
Total	**101**	**64**	**37**

A continuación se presentan los resultados obtenidos con cada uno de los algoritmos de clasificación:

5.2.1. Árbol de decisión

Con el conjunto de entrenamiento se construyó un árbol de decisión y para evitar el sobreajuste que es uno de los principales problemas de este tipo de clasificadores; en la *Tabla 5.5* se evaluó el *error estándar relativo* con respecto al número de divisiones mediante un proceso de *validación cruzada*. La validación cruzada consiste en dividir el conjunto de datos en *n particiones* y posteriormente ejecutar el algoritmo tantas veces como particiones se hayan generado [46].

Tabla 5.5.- Error estándar relativo con respecto al tamaño del árbol

Número de divisiones	Error estándar relativo
0	1.00000
1	0.37297
2	0.35135
3	0.26946
4	0.18378
6	0.15676
8	0.17838

El *error estándar mínimo* que se obtuvo fue de 0.15676 por lo que árbol completo fue podado para obtener uno nuevo (*Figura 5.3)* con *seis divisiones*. Cada división representa una condición, si la condición especificada en un nodo es satisfecha entonces se toma la rama a la izquierda. El árbol de *decisión podado* es mucho más simple que el *árbol completo*.

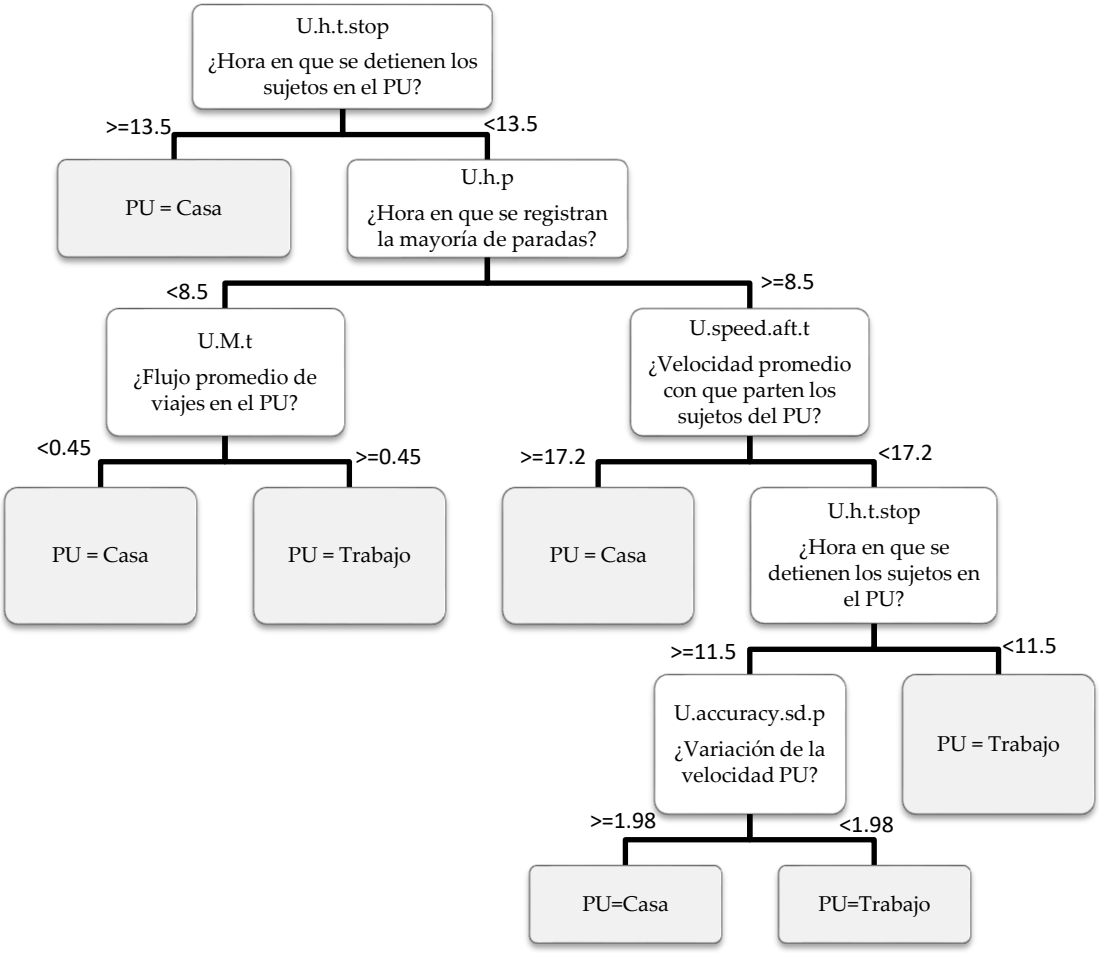


Figura 5.3.- Árbol de decisión para clasificación de PUs.

La *Tablas 5.6* muestran los resultados obtenidos al evaluar el rendimiento del árbol de decisión sobre el *conjunto de prueba*. Se puede clasificar los PUs en casa o trabajo con una *exactitud* de 86% y con una *tasa de error* del 14%.

Tabla 5.6.- Resultados de árbol de decisión para exactitud

Viajes	**Datasets**						***Prom.***
	I	**II**	**III**	**IV**	**V**	**VI**	
1	0.727	0.727	0.727	0.681	0.818	0.772	**0.742**
2	0.954	0.954	0.863	0.863	0.773	0.909	**0.886**
3	0.904	0.904	1	0.905	0.857	0.809	**0.896**
4	0.9	0.9	0.85	0.9	0.9	0.9	**0.891**
5	0.937	0.937	0.875	0.875	0.813	1	**0.906**
1-5	***0.881***	***0.881***	***0.861***	***0.841***	***0.831***	***0.871***	***0.861***

La *precisión* es de 84.9% para predecir PUs en casa y de 88% para trabajo; mientras que la *exhaustividad* es de 92.7% para predecir PUs en casa y 77% para trabajo, ver Tabla 5.7. Los valores de precisión son altos para ambas clases de PUs; sin embargo, al observar los valores de exhaustividad se evidencia que el clasificador es más eficaz para clasificar PUs en casa que en trabajo.

Por otro lado, al clasificar PUs en casa obtenidos con un solo viaje se encontró que la precisión disminuye considerablemente mientras que la exhaustividad aumenta; y al clasificar PUs en trabajo sucede al contrario ya que la exhaustividad disminuye mientras que la precisión aumenta. Al incrementar el número de viajes para detectar un PU, se agrega información a los PUs y por tanto las características del PU mejoran y consecuentemente el rendimiento del clasificador.

Tabla 5.7.- Resultados de árbol de decisión para precisión y exhaustividad.

POIs	Viajes	Precisión							Exhaustividad						
		Datasets						*Prom.*	Datasets						*Prom.*
		I	II	III	IV	V	VI		I	II	III	IV	V	VI	
Casa	**1**	0.642	0.642	0.642	0.714	0.714	0.714	**0.678**	0.9	0.9	0.9	0.769	1	0.909	**0.896**
	2	1	0.928	0.864	0.786	0.714	0.875	**0.861**	0.933	1	0.923	1	0.909	1	**0.96**
	3	0.923	0.846	1	0.846	0.769	0.923	**0.884**	0.923	1	1	1	1	0.8	**0.953**
	4	0.923	0.846	0.875	0.923	0.846	0.923	**0.889**	0.923	1	0.923	0.923	1	0.923	**0.948**
	5	1	1	1	1	0.9	1	**0.893**	0.909	0.909	0.833	0.833	0.818	1	**0.883**
	1-5	**0.89**	**0.843**	**0.859**	**0.843**	0.781	**0.875**	**0.849**	**0.92**	**0.964**	**0.917**	**0.9**	0.943	**0.918**	**0.927**
Trabajo	**1**	0.875	0.875	0.875	0.625	1	0.875	**0.854**	0.583	0.583	0.583	0.556	0.667	0.863	**0.639**
	2	0.875	1	0.875	1	0.875	1	**0.937**	1	0.889	0.78	0.727	0.636	0.8	**0.805**
	3	0.875	1	1	1	1	0.625	**0.916**	0.875	0.8	1	0.8	0.727	0.833	**0.839**
	4	0.857	1	0.875	0.857	1	0.857	**0.907**	0.857	0.78	0.75	0.857	0.778	0.857	**0.813**
	5	0.833	0.833	0.667	0.667	0.667	1	**0.777**	1	1	1	1	0.8	1	**0.967**
	1-5	***0.864***	***0.946***	***0.861***	***0.837***	0.918	***0.864***	***0.881***	***0.82***	***0.78***	***0.78***	***0.756***	0.708	***0.8***	***0.774***

Los valores de *F-score* promedio son de 88.5% para predecir PUs en casa y de 82% para trabajo lo que significa que el clasificador obtenido es estable. Al observar esta métrica con PUs obtenidos con distinto número de viajes se confirma que el equilibrio del clasificador mejora con los PUs detectados en más de dos viajes.

Tabla 5.8.- Resultados de árbol de decisión para F-score

POIs	Viajes	Datasets						Prom.
		I	II	III	V	V	VI	
Casa	**1**	0.749	0.749	0.749	0.740	0.833	0.800	**0.770**
	2	0.965	0.963	0.893	0.880	0.800	0.933	**0.906**
	3	0.923	0.917	1.000	0.917	0.869	0.857	**0.914**
	4	0.923	0.917	0.898	0.923	0.917	0.923	**0.917**
	5	0.952	0.952	0.909	0.909	0.857	1.000	**0.930**
	1-5	**0.905**	**0.899**	**0.887**	**0.871**	**0.854**	**0.896**	**0.885**
Trabajo	**1**	0.700	0.700	0.700	0.588	0.800	0.869	**0.726**
	2	0.933	0.941	0.825	0.842	0.737	0.889	**0.861**
	3	0.875	0.889	1.000	0.889	0.842	0.714	**0.868**
	4	0.857	0.876	0.808	0.857	0.875	0.857	**0.855**
	5	0.909	0.909	0.800	0.800	0.727	1.000	**0.858**
	1-5	**0.841**	**0.855**	**0.819**	**0.794**	**0.799**	**0.831**	**0.823**

5.2.2. Bosques aleatorios

Se entrenó un clasificador sobre el *conjunto de entrenamiento* con el algoritmo de *bosques aleatorios* para lo cual inicialmente se creó múltiples árboles de decisión. Elaborar numerosos árboles y combinar los resultados mejora en gran medida la capacidad predictiva del árbol de decisión elaborado previamente; sin embargo, demasiados árboles pueden conducir a un ajuste o varianza excesiva.

Por ello, para entrenar el clasificador se examinó en la *Figura 5.4* su comportamiento entre *500 árboles de decisión* con la *tasa de error*. La gráfica indica que después indica que después de 100 *árboles*, no hay reducción significativa en la *tasa de error*; es decir, únicamente se necesitan *100 árboles* para optimizar la precisión del clasificador.

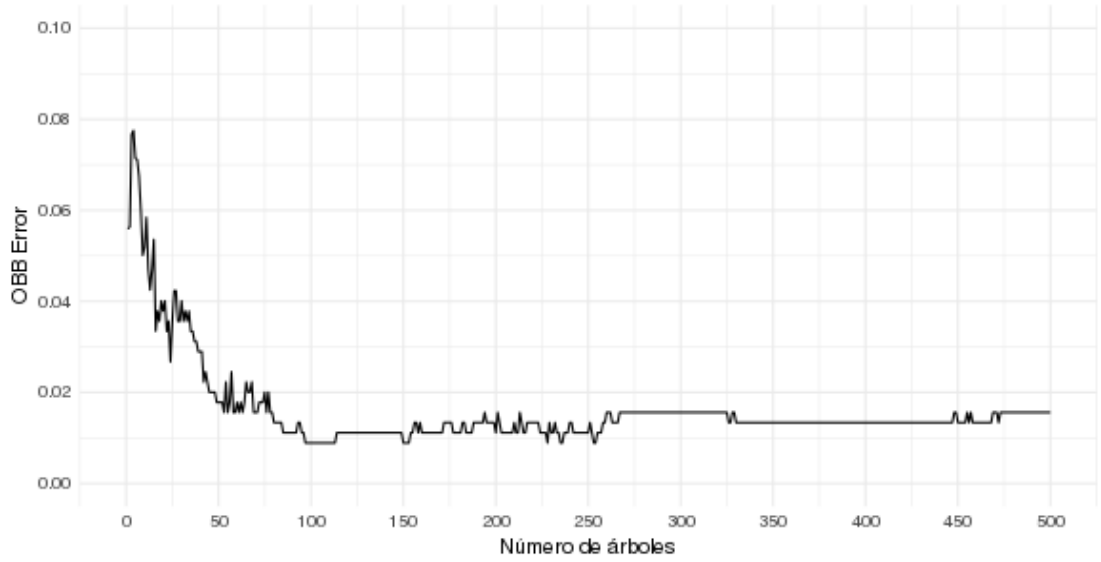


Figura 5.4.- Variación del error cuadrático medio al variar el número de árboles

Y para determinar la cantidad óptima de atributos m_{try} en el algoritmo se evaluó también la tasa de error aumentando la cantidad de atributos de forma progresiva. La *Figura 5.5* indica que al llegar a los *3 atributos* obtiene la *tasa de error* mínima. Por consiguiente, los *bosques aleatorios* se entrenaron con *3 atributos* y 100 árboles sobre el *conjunto de entrenamiento*.

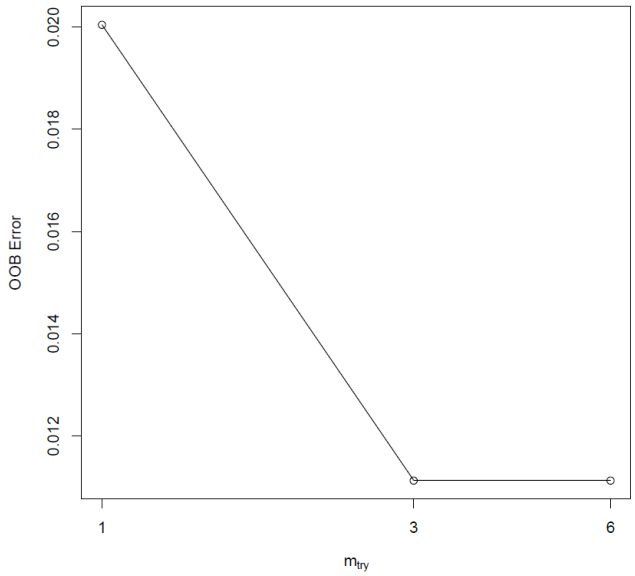


Figura 5.5- Importancia de atributos en la deducción de PUs.

Al evaluar el clasificador con el *conjunto de prueba* fue posible clasificar PUs en POIs casa y trabajo con una exactitud del 86.6% y una tasa de error del 13.4% como se evidencia en la *Tabla 5.9*.

Tabla 5.9.- Resultados de bosques aleatorios para exactitud

Viajes	Datasets						*Prom.*
	I	II	III	IV	V	VI	
1	0.681	0.772	0.727	0.727	0.863	0.818	**0.764**
2	0.818	0.954	0.863	0.864	0.772	0.909	**0.863**
3	0.904	0.905	0.952	0.904	0.904	0.904	**0.912**
4	0.9	0.8	0.85	0.85	0.95	0.9	**0.875**
5	0.875	0.937	0.938	0.875	1	0.93	**0.925**
1-5	***0.841***	***0.871***	***0.861***	***0.841***	***0.891***	***0.89***	***0.866***

Se puede clasificar los PUs casa con una precisión del 94% y los de trabajo en un 73% y con una exactitud de 85.9% y 88% respectivamente, ver *Tabla 5.10*. El clasificador también se comporta con mayor *precisión* y *exhaustividad* clasificando PUs en casa. La clasificación de los PUs es buena, ya que los valores de exhaustividad son superiores al 85% en ambos casos de PUs pese a que la precisión es mayor para clasificar PUs casa que PUs trabajo. Igualmente se evidencia que los valores de exhaustividad mejoran al clasificar PUs obtenidos con más de dos viajes.

Tabla 5.10.- Resultados de bosques aleatorios para precisión y exhaustividad

	Viaje	Precisión							Exhaustividad						
		I	II	III	IV	V	VI	Prom.	I	II	III	IV	V	VI	Prom.
Casa	**1**	0.714	0.928	0.857	0.857	1	0.928	**0.881**	0.769	0.764	0.75	0.75	0.823	0.813	**0.778**
	2	1	1	1	1	0.928	0.923	**0.975**	0.778	0.933	0.823	0.823	0.764	0.875	**0.833**
	3	0.923	0.923	1	0.923	0.923	1	**0.949**	0.923	0.923	0.928	0.923	0.923	0.867	**0.915**
	4	0.923	0.846	0.923	0.923	0.923	0.923	**0.910**	0.923	0.917	0.857	0.857	1	0.923	**0.913**
	5	1	1	1	1	1	1	**1**	0.833	0.909	0.909	0.833	1	1	**0.914**
	1-5	**0.906**	**0.937**	**0.953**	**0.938**	**0.953**	**0.968**	**0.943**	**0.852**	**0.869**	**0.847**	**0.821**	**0.884**	**0.873**	**0.859**
Trabajo	**1**	0.625	0.5	0.5	0.5	0.625	0.625	**0.563**	0.556	0.8	0.667	0.667	1	0.833	**0.754**
	2	0.5	0.875	0.625	0.625	0.5	0.75	**0.646**	1	1	1	1	0.8	0.857	**0.943**
	3	0.875	0.875	0.875	0.875	0.875	0.75	**0.854**	0.875	0.875	1	0.875	0.875	1	**0.917**
	4	0.857	0.875	0.714	0.714	1	0.857	**0.836**	0.857	0.75	0.833	0.833	0.875	0.857	**0.834**
	5	0.667	0.833	0.833	0.667	1	1	**0.833**	1	1	1	1	1	1	**1**
	1-5	***0.729***	***0.757***	***0.703***	***0.648***	***0.783***	***0.757***	***0.730***	***0.818***	***0.875***	***0.892***	***0.857***	***0.906***	***0.933***	***0.88***

Además, se observa que los valores de *F-score* son 89.8% y 90.3% para PUs casa y trabajo demuestran que hay un equilibrio en la clasificación, ver *Tabla 5.11*. Se demuestra también que la estabilidad del clasificador es mejor con PUs obtenidos con más viajes puesto que la clasificación de PUs obtenidos con menos de dos viajes ostenta un *F-score* es inferior al 80%.

Tabla 5.11.-Resultados de bosques aleatorios para F-score

POIs	Viajes	Datasets						Prom.
		I	II	III	V	V	VI	
Casa	1	0.740	0.838	0.800	0.800	0.903	0.867	0.825
	2	0.875	0.965	0.903	0.903	0.838	0.898	0.897
	3	0.923	0.923	0.963	0.923	0.923	0.929	0.931
	4	0.923	0.880	0.889	0.889	0.960	0.923	0.911
	5	0.909	0.952	0.952	0.909	1.000	1.000	0.954
	1-5	**0.878**	**0.902**	**0.897**	**0.876**	**0.917**	**0.918**	**0.898**
Trabajo	1	0.588	0.615	0.572	0.572	0.769	0.714	0.638
	2	0.667	0.933	0.769	0.769	0.615	0.800	0.759
	3	0.875	0.875	0.933	0.875	0.875	0.857	0.882
	4	0.857	0.808	0.769	0.769	0.933	0.857	0.832
	5	0.800	0.909	0.909	0.800	1.000	1.000	0.903
	1-5	**0.771**	**0.812**	**0.786**	**0.738**	**0.840**	**0.836**	**0.8**

5.2.3. Máquinas de soporte

Para determinar el tipo de *kernel* más apropiado para entrenar este clasificador se utilizó también un proceso de *validación cruzada* sobre el *conjunto de entrenamiento* y se evaluó el *kernel lineal, radial, sigmoid y polinomial* con los parámetros requeridos por cada uno de ellos. El parámetro *cost* se refiere al nivel de penalización permitido por cada uno de los *kernels* [23] . La *Tabla 5.12* muestra los resultados que revelan que el *kernel polinomial* es el más apropiado ya que el error estándar relativo es menor.

Tabla 5.12.- Resultados de máquinas de soporte para evaluación de kernels

Kernel *(K)*	Parámetros				Error estándar relativo
	Cost	Gamma(γ)	Coef0 (c_o)	Degree *(d)*	
Linear	100	-	-	-	0.09348485
Radial	0.1	0.1	-	-	0.1424242
Sigmoid	0.1	0.1	0.1	-	0.164798
Polynomial	0.1	0.5	2	3	0.02888889

Las máquinas de soporte se entrenaron con un *kernel polinomial (cost=0.1, gamma=0.5, coef0=2, degree=3)*. Al evaluar el clasificador con el *conjunto de prueba* se obtuvo una exactitud de 79.3%; y una tasa de error del 20.7%, ver *Tabla 5.13*.

Tabla 5.13.- Resultados de máquinas de soporte para exactitud

Viajes	Datasets						*Prom.*
	I	II	III	IV	V	VI	
1	0.681	0.818	0.727	0.773	0.909	0.818	**0.787**
2	0.772	0.863	0.727	0.727	0.772	0.681	**0.757**
3	0.809	0.809	0.714	0.809	0.809	0.857	**0.801**
4	0.75	0.75	0.65	0.9	0.85	0.85	**0.791**
5	0.813	0.75	0.937	0.875	0.875	0.813	**0.843**
1-5	***0.762***	***0.80***	***0.742***	***0.812***	***0.841***	***0.802***	***0.793***

La *precisión* del clasificador es del 84.6% los PUs casa y 74.6% los PUs trabajo; y con una *exactitud* del 85.5% los PUs casa y 73% los PUs trabajo (*Tabla 5.14*). Las tasas de *precisión* y *exactitud* para clasificar PUs en trabajo son inferiores al 80% lo que evidencia que el clasificador funciona mejor clasificando PUs en casa.

Tabla 5.14.- Resultados de máquinas de soporte para precisión y exhaustividad

	Viaje	Precisión							Exhaustividad						
		I	II	III	IV	V	VI	Prom.	I	II	III	IV	V	VI	Prom.
Casa	**1**	0.714	0.9280.	0.714	0.857	0.928	0.928	**0.828**	0.769	0.812	0.833	0.8	0.928	0.928	**0.845**
	2	0.857	0.857	0.928	0.714	0.857	0.857	**0.845**	0.8	0.923	0.722	0.833	0.8	0.8	**0.813**
	3	0.846	0.769	0.846	0.846	0.857	0.846	**0.835**	0.846	0.909	0.687	0.846	0.857	0.846	**0.831**
	4	0.692	0.769	0.692	0.923	0.846	0.846	**0.794**	0.9	0.833	0.75	0.923	0.917	0.916	**0.873**
	5	0.692	0.7	1	1	1	1	**0.898**	0.818	0.875	0.909	0.833	0.833	0.833	**0.85**
	1-5	**0.796**	**0.813**	**0.828**	**0.859**	**0.89**	**0.89**	**0.846**	**0.822**	**0.867**	**0.779**	**0.846**	**0.967**	**0.863**	**0.857**
Trabajo	**1**	0.625	0.625	0.75	0.625	0.875	0.875	**0.729**	0.556	0.833	0.6	0.714	0.875	0.875	**0.742**
	2	0.865	0.875	0.375	0.75	0.625	0.625	**0.685**	0.714	0.778	0.75	0.6	0.714	0.714	**0.711**
	3	0.75	0.875	0.375	0.75	0.75	0.75	**0.708**	0.75	0.7	0.6	0.75	0.75	0.75	**0.716**
	4	0.857	0.714	0.571	0.857	0.857	0.857	**0.785**	0.6	0.625	0.5	0.857	0.75	0.75	**0.68**
	5	0.667	0.833	0.833	0.667	0.667	0.667	**0.722**	0.5	0.625	1	1	1	1	**0.854**
	1-5	***0.702***	***0.784***	***0.595***	***0.729***	***0.933***	***0.756***	***0.749***	***0.667***	***0.707***	***0.667***	***0.75***	***0.8***	***0.8***	***0.731***

El *F-score* es 85.1% para clasificar PUs en casa y 73.9% para clasificar PUs en trabajo (*Tabla 5.15*); por otra parte esta métrica no varía en relación al número de viajes a diferencia de los clasificadores mencionados previamente con los PUs trabajo; lo que

indica que el rendimiento del clasificador para PUs que son trabajo no mejora conforme aumenta la cantidad de información.

Tabla 5.15.- Resultados de máquinas de soporte para F-score

POIs	Viajes	Datasets						Prom.
		I	II	III	V	V	VI	
Casa	1	0.740	0.866	0.769	0.828	0.928	0.928	**0.843**
	2	0.828	0.889	0.812	0.769	0.828	0.828	**0.825**
	3	0.846	0.833	0.758	0.846	0.857	0.846	**0.831**
	4	0.782	0.800	0.720	0.923	0.880	0.880	**0.831**
	5	0.750	0.778	0.952	0.909	0.909	0.909	**0.868**
	1-5	0.809	0.839	0.803	0.852	0.927	0.876	**0.851**
Trabajo	1	0.588	0.714	0.667	0.667	0.875	0.875	**0.731**
	2	0.782	0.824	0.500	0.667	0.667	0.667	**0.684**
	3	0.750	0.778	0.462	0.750	0.750	0.750	**0.707**
	4	0.706	0.667	0.533	0.857	0.800	0.800	**0.727**
	5	0.572	0.714	0.909	0.800	0.800	0.800	**0.766**
	1-5	0.684	0.744	0.629	0.739	0.861	0.777	**0.739**

Finalmente, al comparar los resultados de los tres clasificadores en la *Tabla 5.16* se determinó que el clasificador más eficaz fue el de los bosques aleatorios ya que permitió clasificar PUs en casa y trabajo con mayor *exactitud (86.6%)* que el *árbol de decisión* y las *máquinas de soporte*. La *precisión* promedio es mayor para el *árbol de decisión (88.8%)* que para los *bosques aleatorios (83.8%)* y las *máquinas de soporte (79.7%)*; sin embargo, la *exhaustividad* promedio es mayor para los *bosques aleatorios* (87%) que para el *árbol decisión (82.7%)* y las *máquinas de soporte (79%)*. *El F-score* promedio es también mayor para los *bosques aleatorios (86.4%)* que para el *árbol de decisión (86.4%)* y las *máquinas de soporte (79.5%)*.

Tabla 5.16.- Resultados obtenidos con los tres algoritmos de clasificación

Métricas de evaluación	Árbol de decisión			Bosques aleatorios			Máquinas de soporte		
	Casa	*Trabajo*	***Prom.***	*Casa*	*Trabajo*	***Prom.***	*Casa*	*Trabajo*	***Prom.***
Exactitud	0.86		**0.86**	0.866		**0.866**	0.79		**0.79**
Error	0.14		**0.14**	0.134		**0.134**	0.21		**0.21**
Precisión	0.849	0.927	**0.888**	0.943	0.734	**0.838**	0.846	0.749	**0.797**
Exhaustividad	0.881	0.774	**0.823**	0.86	0.88	**0.87**	0.857	0.731	**0.794**
F-score	0.885	0.823	**0.854**	0.899	0.83	**0.864**	0.851	0.739	**0.795**

Los experimentos sobre los tres clasificadores también revelaron que las características de PUs son más precisas conforme aumenta el número de viajes para su detección ya que hay mayor cantidad de información en cada PU y consecuentemente el rendimiento del clasificador aumenta. La clasificación con los *bosques aleatorios* mejoró con los PUs obtenidos con más de tres viajes; mientras que con los *árboles de decisión* y *máquinas de vectores* los resultados mejoraron con los PUs obtenidos con más de dos viajes.

Conclusiones y trabajos futuros

El algoritmo basado en *velocidad* tiene un mejor rendimiento que los algoritmos basados en *densidad* y el de *distancia-tiempo* para la detección POIs casa y trabajo ya que permite detectar hasta el 90% de PUs que corresponden a dichos POIs en un radio de *100 metros* con al menos *tres viajes*; lo que significa que un sujeto deberá partir o arribar a un punto de ubicación al menos por tres ocasiones para que el PUs sea detectado.

El algoritmo de *densidad* presenta también buenos resultados para la detección de PUs con respecto al algoritmo de *distancia-tiempo* ya que las tasas de detección son superiores al 80%. Los resultados evidencian también que este es mejor para la detección de casas con al menos de dos viajes que trabajos con al menos cuatro viajes; sin embargo, este algoritmo también es más costoso computacionalmente que los otros dos algoritmos presentados.

Además, se encontró que las tasas de detección para PUs que corresponden a los POIs casa son más altas que los PUs que corresponden a los POIs trabajo y estas se equiparan a medida que el radio de la geometría circular construida alrededor del POI aumenta ya que generalmente el área de los lugares de trabajo es más grande que el área de las casas. Asimismo, se evidencio que las velocidades registradas por los sujetos al salir de casa son superiores a las registradas al salir de sus trabajos lo que se puede explicar por la premura que estos tendrían en para llegar a sus destinos tanto en la mañana a sus trabajos como en la tarde a sus oficinas.

Las características derivadas de información geoespacial, temporal y de viajes que fueron planteadas en este trabajo permiten entrenar un clasificador eficaz. Además, se determinó que el desempeño del clasificador entrenado con los *bosques aleatorios* es superior al del *árbol de decisión* y las *máquinas de soporte* dado que permite clasificar los PUs en casa y trabajo con 86.6% de *exactitud, 83.5% de precisión, 87% de exhaustividad y un F-score de 86.4%*. Se comprobó también que el clasificador funciona mejor con PUs obtenidos con más de tres viajes debido a que a medida que aumentan los viajes existe más información y consecuentemente las características son más precisas.

Una de las limitaciones encontradas en este estudio fue que los experimentos fueron realizados sobre un *dataset* de 423 viajes reportados de forma voluntaria por 22 sujetos a través de una aplicación móvil ya que los usuarios reportaron al menos una

vez su POI y un viaje de partida/llegada. Sin embargo, como lo evidencia los resultados de este trabajo se requieren de al menos tres viajes para que el PUs en casa o trabajo sea detectado. Por tanto, se recomienda para trabajos futuros que al momento de realizar la recolección de datos solicitar a los voluntarios que reporten más de tres viajes de partida y llegada a los distintos POIs, probar este método con un *dataset* más grande, y evaluar los enfoques de detección y clasificación con PUs con otros POIs distintos a casa y trabajo; esto con la finalidad de obtener resultados más precisos.

ANEXOS

Anexo I: *Detalle de data sets de sujetos más activos luego de validación de viajes*

No.	Sujetos	Dataset I: Viajes						Dataset II: POIs		
		Viajes			Días	Modos de Transporte	Dispositivos	POIs		
		Cantidad	Casa	Trabajo				Cantidad	Casa	Trabajo
1	4648980	64	51	24	35	3	1	2	1	1
2	4649569	62	57	22	44	5	2	2	1	1
3	169	40	37	11	19	5	1	2	1	1
4	4497987	31	31	0	14	4	1	1	1	0
5	182	28	21	8	21	5	1	2	1	1
6	23	26	16	18	16	3	1	2	1	1
7	22	25	24	14	19	1	1	2	1	1
8	82	22	20	11	14	3	1	2	1	1
9	430	19	13	9	9	1	1	2	1	1
10	152	16	15	8	12	3	2	2	1	1
11	4648776	11	9	6	10	4	1	2	1	1
12	227	10	7	9	8	3	1	2	1	1
13	154	9	4	8	8	2	1	2	1	1
14	428	9	7	4	4	2	1	2	1	1
15	2922733	8	8	3	5	2	1	2	1	1
16	4649148	8	8	1	3	1	1	2	1	1
17	4649878	7	6	4	5	2	2	2	1	1
18	4649217	7	7	0	3	2	1	1	1	0
19	186	6	5	4	5	2	1	2	1	1
20	117	5	1	4	5	1	1	2	1	1
21	1737031	5	5	0	4	1	1	1	1	0
22	4651908	5	5	0	4	4	1	1	1	0
Total:		**359**	**306**	**144**	**-**	**-**	**-**	**40**	**22**	**18**

BIBLIOGRAFÍA

[1] Andrienko, G., Andrienko, N., Fuchs, G., Raimond, A.M.O., Symanzik, J. and Ziemlicki, C. 2013. Extracting Semantics of Individual Places from Movement Data by Analyzing Temporal Patterns of Visits. *Proceedings of The First ACM SIGSPATIAL International Workshop on Computational Models of Place - COMP '13*. (2013), 5–8.

[2] Arnow, B.J. 1994. On laplace's extension of the buffon needle problem. *The College Mathematics Journal*. 25, 1 (1994), 40–43.

[3] Ashbrook, D. and Starner, T. 2003. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*. 7, 5 (2003), 275–286.

[4] Belgium Country Commercial Guide: 2016. *https://www.export.gov/article?id=Belgium-Transportation*. Accessed: 2017-01-01.

[5] Bertram, J.E. and Ruina, A. 2001. Multiplewalking speed–frequency relations are predicted by constrained optimization. *Journal of theoretical Biology*. 209, 4 (2001), 445–453.

[6] Bhattacharya, T., Kulik, L. and Bailey, J. 2014. Automatically recognizing places of interest from unreliable GPS data using spatio-temporal density estimation and line intersections. *Pervasive and Mobile Computing*. 19, (2014), 86–107.

[7] Bhattacharya, T., Kulik, L. and Bailey, J. 2012. Extracting significant places from mobile user GPS trajectories. *Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12*. c (2012), 398.

[8] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.G. 1984. *Classification and Regression Trees*. Belmont.

[9] Chen, C., Gong, H., Lawson, C. and Bialostozky, E. 2010. Evaluating the feasibility of a passive travel survey collection in a complex urban environment: Lessons learned from the New York City case study. *Transportation Research Part A: Policy and Practice*. 44, 10 (2010), 830–840.

[10] Cortes, C. and Vapnik, V. 2001. Support-vector network. *Machine Learning*. 20, (2001), 1–25.

[11] Demissie, M.G., Correia, G.H. de A. and Bento, C. 2013. Exploring cellular

network handover information for urban mobility analysis. *Journal of Transport Geography*. 31, (2013), 164–170.

[12] Do, T.M.T. and Gatica-Perez, D. 2014. The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data. *IEEE Transactions on Mobile Computing*. 13, 3 (2014), 638–648.

[13] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. 1996. A Density-Based Clustering Algorithm for Discovering Clusters. *KDD-96 Proceedings*. (1996), 226–231.

[14] Feng, T. and Timmermans, H.J.P. 2013. Transportation mode recognition using GPS and accelerometer data. *Transportation Research Part C: Emerging Technologies*. 37, (2013), 118–130.

[15] Fu, Z., Tian, Z., Xu, Y. and Qiao, C. 2016. A Two-Step Clustering Approach to Extract Locations from Individual GPS Trajectory Data. *ISPRS International Journal of Geo-Information*. 5, 10 (2016), 166.

[16] Gong, L., Sato, H., Yamamoto, T., Miwa, T. and Morikawa, T. 2015. Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines. *Journal of Modern Transportation*. 23, 3 (2015), 202–213.

[17] Guidotti, R., Monreale, A., Rinzivillo, S., Pedreschi, D. and Giannotti, F. 2015. Retrieving points of interest from human systematic movements. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 8938, (2015), 294–308.

[18] Han, J., Kamber, M. and Jian, P. 2012. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management System Series.

[19] Isaacman, S., Becker, R., Cáceres, R. and Kobourov, S. 2011. Identifying Important Places in People's Lives from Cellular Network Data. *9th International Conference, Pervasive 2011* (San Francisco, USA, 2011), 133–151.

[20] James, G., DanielaWitten, Hastie, T. and Tibshirani, R. 2013. *An Introduction to Statistical Learning with Applications in R*. Springer.

[21] Kang, J.H., Welbourne, W., Stewart, B. and Borriello, G. 2005. Extracting places from traces of locations. *ACM SIGMOBILE Mobile Computing and Communications Review*. 9, 3 (2005), 58.

[22] Lawson, C.T., Chen, C. and Gong, H. 2010. *Advanced Applications of Person-based GPS in an Urban Enviroment*. University at Albany.

[23] Lesmeister, C. 2015. *Mastering Machine Learning with R*. Packt Publishing Ltd.

[24] Li, D. and Du, Y. 2007. *Artificial Intelligence with Uncertainty*. CRC Press: Boca Raton.

[25] Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. and Ma, W.Y. 2008. Mining user similarity based on location history. *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. c (2008), 34.

[26] Lopez, A.J., Semanjski, I. and Gautama, S. 2016. Forecasting Travel Behaviour from Crowdsourced Data with Machine Learning Based Model. *Fifth International Conference on Data Analytics*. 5, (2016), 93–99.

[27] Luo, T., Zheng, X., Xu, G., Fu, K. and Ren, W. 2017. An Improved DBSCAN Algorithm to Detect Stops in Individual Trajectories. *ISPRS International Journal of Geo-Information*. 6, 3 (2017), 63.

[28] Lv, M., Chen, L., Xu, Z., Li, Y. and Chen, G. 2015. The discovery of personally semantic places based on trajectory data mining. *Neurocomputing*. 173, (2015), 1142–1153.

[29] Macqueen, J. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1, 233 (1967), 281–297.

[30] Maimon, O.Z. and Rokach, L. 2005. *Data Mining and Knowledge Discovery Handbook*. Springer.

[31] Marmasse, N. and Schmandt, C. 2000. Location-aware information delivery with comMotion. *Handheld and Ubiquitous Computing*. 1927, 2 (2000), 157–171.

[32] McNeill Alexander, R. 2002. Energetics and optimization of human walking and running: the 2000 raymond pearl memorial lecture. *American Journal of Human Biology*. 14, 5 (2002), 641–648.

[33] Mousavi, S.M., Harwood, A., Karunasekera, S. and Maghrebi, M. 2016. Geometry of interest (GOI): spatio-temporal destination extraction and partitioning in GPS trajectory data. *Journal of Ambient Intelligence and Humanized Computing*. October (2016).

[34] Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O., Tietbohl, A., Bogorny, V., Kuijpers, B. and Alvares, L.O. 2008. A clustering-based approach for discovering interesting places in trajectories. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*. (2008), 863.

[35] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y. and Yan, Z. 2013. Semantic trajectories modeling and analysis. *ACM Computing Surveys*. 45, 4 (2013), 42:1-42:32.

[36] Rodrigue, J.-P., Comtois, C. and Slack, B. 2013. *The Geography of Transport Systems*. Routledge.

[37] Saar-tsechansky, M. and Provost, F. 2007. Handling Missing Values when Applying Classification Models. 8, (2007), 1625–1657.

[38] Schuessler, N. and Axhausen, K. 2008. Identifying trips and activities and their characteristics from GPS raw data without further information. *Transportation Research Record: Journal of the Transportation Research Board.* 2105, (2008), 1–28.

[39] Schuessler, N. and Axhausen, K.W. 2009. Processing Raw Data from Global Positioning Systems Without Additional Information. *Transportation Research Record: Journal of the Transportation Research Board*. 2105, 1 (2009), 28–36.

[40] Shen, L. and Stopher, P.R. 2014. Review of GPS Travel Survey and GPS Data-Processing Methods. *Transport Reviews*. 34, 3 (2014), 316–334.

[41] Stopher, P., Jiang, Q. and FitzGerald, C. 2005. Processing GPS data from travel surveys. *2nd International Colloqium on …*. March (2005), 1–21.

[42] Suh, W., Henclewood, D., Guin, A., Guensler, R., Hunter, M. and Fujimoto, R. 2017. Dynamic data driven transportation systems. *Multimedia Tools and Applications*. (2017), 1–17.

[43] Tran, K. a, Barbeau, S. J., & Labrador, M. a. (2013). Bacchelors: Automatic Identification of Points of Interest in Global Navigation Satellite System Data : A Spatial Temporal Approach Categories and Subject Descriptors, (January), 33–42.Tran, K. a, Barbeau, S.J. and Labrador, M. a 2013. Bacchelors: Automatic Identification of Points of Interest in Global Navigation Satellite System Data : A Spatial Temporal Approach Categories and Subject Descriptors. January (2013), 33–42.

[44] TranSafety, I. 1996. Study compares older and younger pedestrians walking speeds. *Transportation Research Board's Transportation Research Record*. 1538, (1996).

[45] Transport Road Safety: 2017. *http://ec.europa.eu/transport/road_safety/going_abroad/belgium/speed_limits_en.htm*.

Accessed: 2017-01-11.

[46] Witten, I.H., Frank, E. and Hall, M.A. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

[47] Wolf, J., Bachman, W., Oliveira, M.S., Auld, J., Mohammadian, A. (Kouros), Vovsha, P. and Zmud, J. 2014. *Applying GPS Data to Understand Travel Behavior, Volume II: Guidelines*. National Academies Press.

[48] Wolf, J., Bachman, W., Oliveira, M.S., Auld, J., Mohammadian, A. and Vovsha, P. 2014. *Applying GPS Data to Understand Travel Behavior, Volume I: Background, Methods, and Tests*. National Cooperative Highway Research Program.

[49] Zhang, J., Pechenizkiy, M., Pei, Y. and Efremova, J. 2016. A Robust Density-based Clustering Algorithm for Multi-Manifold Structure. (2016), 832–838.

[50] Zheng, V.W., Zheng, Y., Xie, X. and Yang, Q. 2010. Collaborative location and activity recommendations with GPS history data. *Proceedings of the 19th international conference on World wide web WWW 10*. (2010), 1029.

[51] Zheng, Y., Li, Q., Chen, Y., Xie, X. and Ma, W.-Y. 2008. Understanding mobility based on GPS data. *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*. 49 (2008), 312.

[52] Zheng, Y., Zhang, L., Xie, X. and Ma, W.-Y. 2009. Mining Interesting Locations and Travel Sequences fromGPS Trajectories. 49 (2009), 1–10.

[53] Zheng, Z., Rasouli, S. and Timmermans, H. 2014. Evaluating the Accuracy of GPS-based Taxi Trajectory Records. *Procedia Environmental Sciences*. 22, (2014), 186–198.

[54] Zhou, C., Frankowski, D., Ludford, P., Shekhar, S. and Terveen, L. 2007. Discovering personally meaningful places. *ACM Transactions on Information Systems*. 25, 3 (2007).