

PRÁCTICA 1

IA en la Industria Automotriz

Aplicaciones Avanzadas de la IA

**Máster Universitario en Ingeniería Informática
(2023/24)**



Introducción

En el ámbito de la conducción, el desarrollo de Sistemas Avanzados de Asistencia al Conductor (ADAS) tienen como objetivo hacer que la conducción sea más fácil y segura. En particular, uno de los aspectos relevantes de los ADAS consiste en detectar lo que el conductor está haciendo o lo que éste va a hacer. Esta detección de acciones puede realizarse mediante la observación directa, utilizando cámaras de visión frontal, o la observación indirecta, con el uso de determinados sensores en el vehículo.

El modelado del comportamiento del conductor (Driver Behaviour Modelling - DBM) es un problema de investigación que ha sido ampliamente estudiado en muchos trabajos de investigación [1] [2] [3] [4] [5].

En [6], se emplea un enfoque de aprendizaje automático para detectar y clasificar una amplia gama de maniobras de conducción, realizar una selección de sensores automovilísticos a gran escala y detectar la falta de atención de los conductores mediante el uso de los sensores disponibles en el parque automovilístico actual.

En [7] se propone una estructura jerárquica de la tarea del usuario en la carretera. En esa estructura, el rendimiento se divide en tres niveles de habilidades y control:

- 1) estratégico (planificación),
- 2) de maniobra (táctico) y
- 3) de control (operativo).

Los resultados externos de esos tres niveles son, respectivamente:

- 1) planes generales,
- 2) pautas de acción controladas y
- 3) pautas de acción automáticas.

En [8] se define una adaptación de ese modelo jerárquico y se incluye un nivel más: nivel de comportamiento. Este nivel se refiere a las condiciones previas y las características personales, y tiene la máxima prioridad porque estos aspectos influyen en gran medida en las decisiones de conducción en los niveles inferiores. La Figura 1 muestra la estructura jerárquica de estos niveles de conducción.

Otro aspecto muy importante son las entradas utilizadas en la tarea de modelado del conductor. En este sentido, los datos de los vehículos utilizados pueden provenir de la Red de Área de Control (Controller Area Network - CAN), de diferentes sensores (tales como radares, sensores de posición de carril, GPS, acelerómetros o giróscopos), o de diversas cámaras. Esta práctica se centra sólo en los datos del bus CAN.

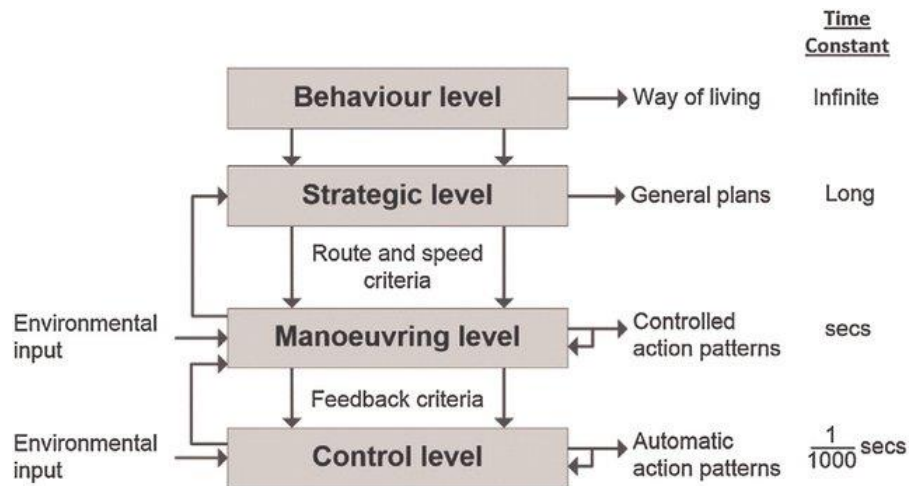


Figura 1. Diferentes niveles jerárquicos del automóvil [8]

Objetivo

A partir del *dataset* entregado, **seleccionar al menos una de las 5 maniobras de conducción y determinar qué secuencia de eventos (acciones atómicas) son características de dicha maniobra**. Tener en cuenta que se deberían considerar todos los ficheros (usuarios) de una misma maniobra para que los eventos sean lo más representativos posibles.

Por otra parte, es necesario que se realice un pre-proceso de las señales obtenidas con la finalidad de desechar información redundante y obtener únicamente aquella información relevante para el objetivo que se quiere alcanzar. En este sentido, algunas de las señales de las que se dispone, podrían ser discretizada y el uso de ventanas de tiempo debe ser considerado.

Para obtener este objetivo, es necesario tener en cuenta los siguientes aspectos:

- 1) Estudiar los ficheros de maniobra, teniendo en cuenta que en los ficheros hay momentos de “no-maniobra”.
- 2) Tener en cuenta que un evento está distribuido a lo largo del tiempo, por lo que deberá variar el tamaño (duración) de la ventana deslizante.

Además, aunque no sea obligatorio, se valorará positivamente el tratamiento de los datos como flujos de datos. Para ello, se podrá utilizar la librería *River* (<https://riverml.xyz/0.10.0/>), diseñada para tal fin. Tal y como se indica en [9], *River* es una nueva biblioteca de Python creada para entrenar modelos de aprendizaje automático de forma incremental, en el entorno de *streaming*. Proporciona algoritmos de aprendizaje de última generación, métodos de transformación de datos y métricas de rendimiento para diferentes tareas de aprendizaje *online*. Es el producto de la fusión de las mejores partes de las librerías *creme* y *scikit-multiflow*, ambas creadas con el mismo objetivo: “proporcionar a la comunidad las herramientas para avanzar en el estado del aprendizaje automático en streaming y promover su uso en aplicaciones del mundo real”.

Cómo se han obtenido los datos a utilizar

Los datos con los que se realizará esta práctica se han obtenido utilizando un entorno de simulación de conducción. El entorno utilizado está basado en los sistemas avanzados de STISIM Drive® [9], e incorpora dispositivos de conducción que simulan el comportamiento de los que son instalados en un vehículo real. Teniendo en cuenta los controles del vehículo, el sistema de simulación incorpora un volante que simula el sistema de dirección asistida y unos pedales como los de un vehículo real, tal y como se puede ver en la Figura 2.



Figura 2. Entorno de Simulación

Este entorno proporciona una experiencia de simulación muy realista durante la recogida de datos de conducción. La metodología que se ha utilizado en el proceso de recogida de datos sobre las acciones y maniobras atómicas consiste en las siguientes fases:

1. **Diseño de un escenario de conducción simulado:** El escenario debe ser diseñado para utilizar elementos en la conducción tales como intersecciones, semáforos o vehículos lentos, que permitan la identificación visual de un conjunto de maniobras M . Además, no se podrán realizar diferentes tipos de maniobra de forma simultánea. La Figura 3 muestra una secuencia de imágenes (obtenidas del simulador de conducción) en las que el conductor ejecuta una maniobra de adelantamiento dentro de un escenario de conducción diseñado para tal fin.



Figura 3. Maniobra de adelantamiento

2. **Recopilación de datos:** Utilizando el escenario diseñado, los datos relacionados con los controles de conducción (es decir, las señales en bruto) son obtenidos y almacenados en tiempo real. También se almacenan los vídeos de la ejecución de la conducción. Como resultado de toda ejecución de un determinado escenario, se obtiene un archivo de datos en el que cada línea (es decir, instancia) se identifica con una marca de tiempo y contiene el conjunto de las señales generadas por los controles de conducción en ese instante de tiempo.
3. **Identificación visual de las maniobras:** Utilizando los vídeos grabados de cada simulación, se realiza un conjunto de ejemplos identificando los intervalos de tiempo en los que el conductor realiza una maniobra específica, $z = 0, 1 \mid m_i \in M$
4. **Fusión de datos en bruto y etiquetas de maniobras:** Las instancias, que pertenecen a los intervalos de tiempo del conjunto de maniobras identificadas, se seleccionan del conjunto de datos. Como resultado de este proceso, se obtiene un nuevo archivo en el que cada instancia está compuesta por el conjunto de señales obtenidas en la fase 2, y el conjunto de etiquetas M obtenidas en la fase 3.

Para el desarrollo de esta práctica, se dispone de cinco maniobras diferentes:

- Adelantar,
- Detenerse,
- Realizar giro,
- Realizar giro en tres pasos,
- Realizar giro en U.

A continuación, se dan más detalles sobre estas dos últimas maniobras:

- Realizar giro en tres pasos: Método común para girar un vehículo en la dirección opuesta en un espacio limitado o en una carretera estrecha. El conductor tiene que utilizar la marcha adelante y la marcha atrás para cruzar la carretera, así después de girar debe ir marcha atrás y de nuevo hacia adelante para completar el giro. Obsérvese la Figura 4 a) que muestra gráficamente la maniobra.

- Realizar giro en U: diferencia del giro en 3 puntos, esta maniobra también se realiza en una carretera recta, pero con espacio suficiente para realizar toda la maniobra de una sola vez. Figura 4 a) que muestra gráficamente esta maniobra.

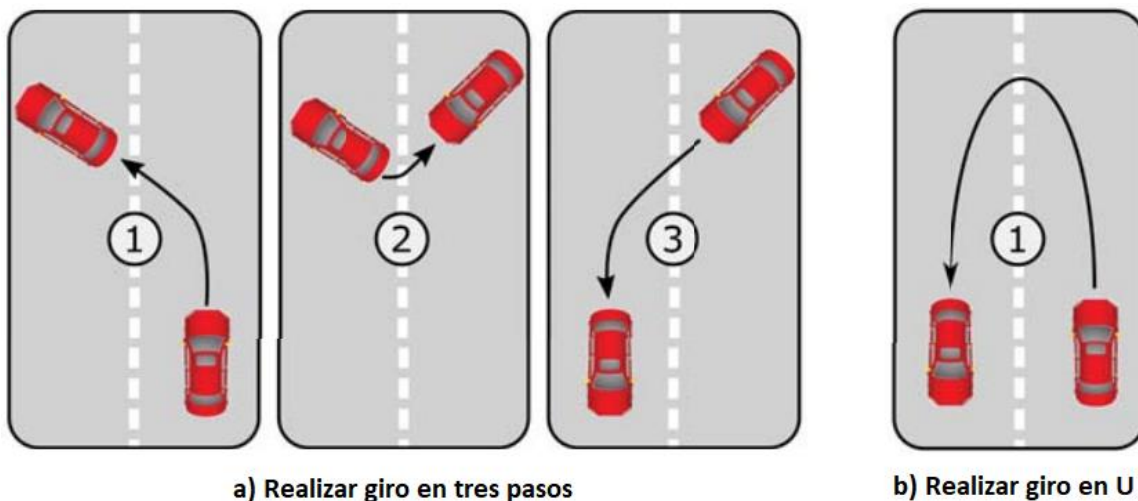


Figura 4. Maniobras: a) Giro en tres pasos y b) Giro en U.

Descripción de los datos a utilizar

Para esta práctica, se dispondrá de los datos recogidos del simulador STISIM Drive de 5 conductores diferentes de la forma indicada en el apartado anterior. Cada uno de los 5 conductores ha realizado cada una de las 5 maniobras explicadas anteriormente (Adelantar, Detenerse, Realizar giro, Realizar giro en tres pasos y Realizar giro en U) durante un periodo de tiempo. Este conjunto de datos se encuentra en AG con el nombre **ManiobrasSimulador.zip**

Así, por cada uno de los conductores se ha generado un fichero por maniobra en el que se almacenan todas las señales (“en crudo”) del simulador recibidas cada instante de tiempo y si el conductor está realizando la maniobra correspondiente. Para ello, se indicará con 1 en el campo *Maneuver marker flag* si el conductor realiza dicha maniobra y un 0 en caso contrario. La cantidad de datos obtenidos por conductor y maniobra es muy alta, ya que estos datos se almacenan 20 veces por segundo.

Las señales almacenadas y que podrán utilizarse en esta práctica son:

- *Speed* - Velocidad,
- *RPM* - Revoluciones por minuto
- *Steering Wheel angle* - Ángulo del volante.
- *Gas Pedal* - Pedal del acelerador.
- *Brake Pedal* - Pedal del freno (b),
- *Clutch Pedal* - Pedal del embrague
- *Gear* – Marcha.

Además, en los conjuntos de datos se indican otros datos (*Long Dist*, *Lat Pos*, *Throttle input*, etc...) que no es necesario que sean utilizados en la práctica. También hay dos datos (*Left turn* y *Right turn*) que no pueden utilizarse para reconocer la maniobra de giros.

Así, por ejemplo, el conductor *driver1* almacena en el directorio de su mismo nombre los ficheros:

- *STISIMData_3step-Turnings.xlsx*
- *STISIMData_Overtaking.xlsx*
- *STISIMData_Stopping.xlsx*
- *STISIMData_Turnings.xlsx*
- *STISIMData_U-Turnings.xlsx*

Cada uno de estos ficheros almacena todos los datos del simulador obtenidos cada 1/20 segundos junto con un valor en el data "*Maneuver marker flag*" que indica si la correspondiente maniobra se está realizando o no. Así, si el campo "**Maneuver marker flag**" es 0 indica que la acción no está realizándose. Sin embargo, el valor es 1 indica que se está ejecutando la maniobra correspondiente.

Por otra parte, el atributo "*Brake pedal force*" está relacionado con la fuerza que ejerce el conductor en el pedal, mientras que "*Brake pedal*" es la "fuerza" que está haciendo el propio coche para frenar. Como ejemplo, podéis considerar que, si se estuviera subiendo una pendiente, "*Brake Pedal Force*" debe ser mayor para obtener el mismo valor en "*Brake Pedal*". Sin embargo, todas las simulaciones en las que se han obtenido los datos se han realizado sobre entornos sin pendientes, por lo que este valor estará totalmente correlado.

En relación con la normalización de datos, se puede considerar lo que consideréis más adecuado. Sin embargo, algo que debéis tener en cuenta es que hay instancias en las que el valor de "*gear*" es 0 durante el instante de tiempo en el que se está realizando un cambio en la marcha. Por ejemplo, en el paso de 3ª a 2ª, entre las instancias con valor 3 y las instancias con valor 2, hay una serie de instancias (normalmente son un número reducido) con valor 0. En este caso, debéis tener en cuenta que está en "punto muerto" por la transición entre una marcha y otra. Por lo tanto, independientemente del valor 0 "entre medias", la normalización en dicho caso se debe considerar como que se está decrementando de marcha.

Desarrollo de la práctica

Para alcanzar el objetivo de esta práctica, el alumno podrá utilizar tanto la técnica de IA como la herramienta que considere más adecuada.

Como se ha comentado, se debe considerar si va a realizarse la práctica considerando los datos como flujos de datos o no. Además, es necesario realizar un pre-proceso de las señales obtenidas, pudiendo discretizar alguna de ellas y, sobre todo, utilizando ventanas de tiempo.

Por último, destacamos dos investigaciones científicas que han utilizado estos datos (o muy similares) con una finalidad más compleja a la planteada en esta práctica, pero aplicando un tipo de sistemas denominados *sistemas autoadaptativo*¹. A pesar de que las técnicas a aplicar en esta práctica son diferentes, pueden servir de referencias estas investigaciones para el tratamiento de los datos, análisis de resultados, etc...

Estas dos investigaciones se detallan a continuación:

1. Goran Andonovski, Oscar Sipele, José Antonio Iglesias, Araceli Sanchis, Edwin Lughofer, Igor Skrjanc: *Detection of driver maneuvers using evolving fuzzy cloud-based system*. 2020 IEEE Symposium Series on Computational Intelligence (Canberra, Australia). Pages: 700-706.

Link: <https://ieeexplore.ieee.org/document/9308520>

2. Igor Skrjanc, Goran Andonovski, Agapito Ledezma, Oscar Sipele, José Antonio Iglesias, Araceli Sanchis. *Evolving cloud-based system for the recognition of drivers' actions*. Expert Systems with Applications 99: 231-238 (2018)

Link: http://msc.fe.uni-lj.si/Papers/2016_%C5%A0krjanc_Driver_detection.pdf

Entrega de la práctica

- La práctica deberá realizarse en grupos de **3-4 personas** (y entregarse únicamente por uno de los integrantes del grupo).
- Dicha entrega constará de:
 - Documento explicativo que detalle la propuesta desarrollada, las herramientas utilizadas, los resultados y su análisis, conclusiones, etc...
 - No hay un formato de entrega establecido, y se si quiere, el documento podría ser el notebook de Python. Sin embargo, es importante que toda la información se muestre de forma clara y su presentación también es considerada para la nota de la práctica.
 - Aquellos ficheros que sean necesarios para “replicar” el sistema.
- La entrega de esta Práctica deberá realizarse por Aula Global antes del día **23 de febrero 2022 a las 15:00h**.

¹ https://es.wikipedia.org/wiki/Sistema_autoadaptativo

Evaluación de la práctica

Aspectos que evaluar en la corrección de práctica:

- Planteamiento y desarrollo del problema: 25%
- Resultados del problema: 25%
- Análisis de resultados y conclusiones: 25%
- Presentación: 25%

Bibliografía

- [1] K. Torkkola, S. Venkatesan y H. Liu, «Sensor sequence modeling for driving sequence modeling for driving,» de *Flairs conference*, 2005.
- [2] A. Sathyanarayana, P. Boyraz y J. H. Hansen, «Driver behavior analysis and route recognition by Hidden Markov Models.,» de *2008 IEEE international conference on vehicular electronics and safety*, Columbus, OH, USA, 2008.
- [3] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda y F. Itakura, «Driver modeling based on driving behavior and its evaluation in driver identification,» *Proceedings of the IEEE*, vol. 95, nº 2, pp. 427-437, 2007.
- [4] B. Shi, L. Xu, J. Hu, Y. Tang, H. Jiang, W. Meng y H. Liu, «Evaluating driving styles by normalizing driving behavior based on personalized driver modeling.,» *IEEE Transactions on Systems, Man, and Cybernetics: SystemS*, vol. 45, nº 12, p. 1502–1508, 2015.
- [5] A. Nakano, H. Okuda, T. Suzuki, S. Inagaki y S. Hayakawa, «Symbolic modeling of driving behavior based on hierarchical mode segmentation and formal grammar,» de *2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, USA, 2009.
- [6] K. Torkkola, M. Gardner, C. Schreiner, K. Zhang, B. Leivian y H. Zhang, «Computational intelligence in automotive applications,» *Studies in computational intelligence*, vol. 132, pp. 59-77, 2008.
- [7] J. A. Michon, «A critical view of driver behavior models: What do we know, what should we do?,» *Human behavior and traffic safety*, p. 485–520, 1985.
- [8] M. Panou, E. Bekiaris y V. Papakostopoulos, «Modelling driver behaviour in european and international projects,» *Modelling driver behaviour in automotive environments: Critical issues in driver interactions with intelligent transport systems*, pp. 3-25, 2007.
- [9] A. Amidon. [En línea]. Available: <https://towardsdatascience.com/river-the-best-python-library-for-online-machine-learning-56bf6f71a403>.
- [10] Systems Technology, «Car driving simulator & simulation software,» 2016. [En línea]. Available: [http:// www.stisimdrive.com/products/simulation-systems/m100-series ..](http://www.stisimdrive.com/products/simulation-systems/m100-series..)
- [11] J. A. Iglesias, P. Angelov y A. a. S. A. Ledezma, «Creating Evolving User Behavior Profiles Automatically,» *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, nº 2, pp. 131-149, 2012.

- [12] J. A. Iglesias, F. J. Ordoñez, A. Ledezma, P. de Toledo y A. Sanchis, «Evolving Activity Recognition from Sensor Streams,» de *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS 12)*, Madrid, Spain., 2012.
- [13] F. J. Ordóñez, J. A. Iglesias, P. d. Toledo, A. Ledezma y A. Sanchis, «Online activity recognition using evolving classifiers.,» *Expert Systems with Applications*, vol. 40, n° 4, pp. 1248-1255, 2013.
- [14] N. Abuali y H. Abou-zeid, «Driver behavior modeling : Developments and future directions,» *International Journal of Vehicular Technology*, pp. 1-12, 2016.
- [15] A. Doshi y M. M. Trivedi, «Tactical driver behavior prediction and intent inference: A review,» de *IEEE conference on intelligent transportation systems*, Washington, DC, USA, 2011.
- [16] W. Wang, J. Xi y H. Chen, «Modeling and recognizing driver behavior based on driving data: A survey,» *Mathematical Problems in Engineering*, pp. 1-20, 2014.
- [17] A. A. Chaaraoui, P. Climent-Perez y F. Florez-Revuelta, «A review on vision techniques applied to human behaviour analysis for ambient-assisted living.,» *Expert Systems with Applications*, , vol. 39, n° 12, pp. 10873-10888, 2012.
- [18] T. Levermore, A. Ordys y J. Deng, «A review of driver modelling,» de *2014 UKACC international conference on control*, Loughborough, U.K, 2014.