

Predicción de destinos geográficos mediante Modelos de Markov Ocultos

J. A. Álvarez-García^a, L. González-Abril^b y J. A. Ortega^a

^a Dpto. Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Avda Reina Mercedes s/n.

^b Dpto. de Economía Aplicada I. Universidad de Sevilla. Avda. Ramón y Cajal, s/n.

E-mail: {jaalvarez,luisgon}@us.es, ortega@lsi.us.es

Abstract

En este artículo se propone una metodología basada en los modelos de Markov Ocultos para realizar predicciones del futuro lugar geográfico hacia el que nos dirigimos cuando comenzamos un nuevo recorrido. En primer lugar recogemos datos suficientes como para poder generar una estructura de rutas basadas en similitudes entre recorridos. Una vez que conseguimos las rutas, detectamos automáticamente los cruces entre éstas y utilizamos los datos para entrenar al modelo. Una vez entrenado, a partir de una serie de observaciones predecimos el destino más probable hacia el que nos dirigimos basándonos en el recorrido real del usuario utilizando el algoritmo de Viterbi.

1. Resumen

Desde el año 2002, se juega con la idea de predecir con suficiente antelación el destino y la ruta que seguirá una persona cuando comienza un nuevo recorrido. Este tipo de predicción, permitiría múltiples aplicaciones potenciales que no sólo tuvieran información de la localización actual, sino que manejasen la localización futura que van a encontrarse en un instante temporal futuro de minutos o de horas. La utilidad de estas aplicaciones es muy elevada, ya que conociendo con antelación el lugar al que nos dirigimos en un instante futuro, se pueden preparar tareas complejas o costosas en tiempo como predecir encuentros [Ashbrook and Starmer, 2003] con otras personas en el camino que se está realizando o en el lugar destino, buscar recorridos más eficientes para llegar antes [Krumm and Horvitz, 2006], utilizar la información de los puntos de interés que atravesaremos para avisar de las tareas pendientes [Marmasse, 2004], ahorrar energía en vehículos híbridos [Johannesson *et al.*, 2007] o mostrar información relevante para la navegación hacia esos lugares sin necesidad de interaccionar con ningún dispositivo [Terada *et al.*, 2006; Torckkola *et al.*, 2007].

Normalmente las predicciones se apoyan en el histórico de desplazamientos de la persona dado que se ha podido comprobar que la periodicidad y repetición de las rutas y destinos típicos es elevado [Jensen *et al.*, 2004; Krumm and Horvitz, 2005], dándose en algunos casos que más del 60% de los desplazamientos corresponden a rutas ya realizadas. Aunque

estos datos dependen del lugar, la época del año y del tipo de persona, esta información es muy valiosa tanto para el correcto funcionamiento de las aplicaciones predictivas. Para conseguirla es necesario hacer una recogida de datos costosa tanto en tiempo como en procesado. En el primer caso se necesita aproximadamente un mes de recorridos del usuario para alimentar la base de datos inicial que permite realizar una clasificación supervisada. El coste en procesado es importante ya que es necesario realizar filtrados de puntos, regeneración de rutas en las que se pierde información por el efecto cañón o la tardía recepción de los satélites validados y agrupamiento de recorridos muy semejantes que permitan distinguir recorridos canónicos de los no canónicos. Por último, el coste en almacenamiento no es trivial ya se recoge información de lo que podría nombrarse como “sistema de información geográfica personal”, en cuanto se almacenan todos los puntos por los que el usuario pasa y su información histórica.

En este trabajo se utilizará como lógica, un modelo de predicción novedoso basado en los Modelos de Markov Ocultos [Rabiner and Juang, 1986].

2. Modelos de predicción de destinos

Tal y como hemos comentado, nuestro problema es abordar la predicción de rutas y lugares destino con la mayor antelación posible cuando una persona que utiliza un sistema de geoposicionamiento comienza un recorrido.

Ante este problema, se han desarrollado diferentes propuestas que podemos clasificar en aquellas que se basan en similitudes y las que se basan en modelos estadísticos.

- Las que se basan en similitudes intentan encontrar una medida que indique si la secuencia de puntos obtenidos por el sistema de localización durante un trayecto, se parece lo suficiente a una de las rutas etiquetadas con su correspondiente destino que se encuentra en la base de datos inicial. Si se encuentra alguno, el recorrido en curso se puede clasificar como la ruta de la base de conocimiento y por tanto conocer su destino final. Entre estas propuestas destacan [Froehlich and Krumm, 2008; Torckkola *et al.*, 2007; Alvarez *et al.*, 2007].
- Las que se basan en modelos estadísticos, indican una probabilidad con la que nos aproximamos a un recorrido u otro. En [Ashbrook and Starmer, 2003] se propone un modelo de Markov de segundo orden que obvia la

ruta y se centra únicamente en el destino que se pretende alcanzar. En los trabajos del grupo de Dieter Fox en la Universidad de Washington [Liao *et al.*, 2004; 2005; Patterson *et al.*, 2004] se desarrolla un modelo complejo basado en una red jerárquica Bayesiana dinámica que mejora los resultados del modelo de Markov. Por otra parte, otros modelos estadísticos han sido aplicados a problemas parecidos como la localización del siguiente tramo de carretera que escogerá un conductor. En estos casos en el que la predicción se aplica a medidas para transportes inteligentes, los modelos más importantes son [Simmons *et al.*, 2006] con un Modelo de Markov Oculto y [Krumm, 2008] con un modelo de Markov de décimo orden.

3. Nuestro modelo

Tras haber estudiado modelos basados en similitudes en [Alvarez *et al.*, 2007], pasamos a describir una nueva propuesta basada en un modelo de Markov oculto (Hidden Markov Model o HMM). Los HMM, son modelos doblemente estocásticos ya que uno de los procesos participantes no puede ser observado directamente (está oculto), sino que sólo puede hacerse a través de otro conjunto de procesos estocásticos, los cuales producen la secuencia de observaciones.

Se especifican mediante 5 componentes: (Q, V, π, A, B) que describimos a continuación adaptándolos a nuestra propuesta:

- Q : Es el conjunto de estados del modelo. Los estados se etiquetan como $1, 2, K, N$, y el estado actual en el instante de tiempo t es q_t . Nuestros estados corresponden con los lugares hacia los que nos dirigimos en un desplazamiento. Para tener una predicción basada en el histórico de desplazamientos, consideraremos sólo aquellos lugares que visitamos al menos una vez al mes. N será el número de estados.
- V : Es el conjunto de sucesos que pueden observarse en los distintos estados del modelo, también llamados observaciones y corresponderán con la localización (latitud, longitud, altura) particular que ocupa un usuario en un instante determinado del recorrido. Estos puntos serán una variación de los puntos de cruce entre rutas. Concretamente, será aquel área previa o posterior a un cruce entre diferentes rutas. M será el número de observaciones.
- $\pi = \{\pi_i\}$: Contiene la probabilidad de cada estado de ser el estado inicial. Normalmente el estado con mayor probabilidad es el hogar de la persona estudiada.
- $A = \{a_{ij}\}$: Contiene las probabilidades de transición entre estados. Esta matriz, indicará la probabilidad de que estando en un estado, cambiemos a otro. Es decir, que dirigiéndonos a un destino, cambiemos nuestra ruta para ir hacia otro lugar destino.
- $B = \{b_j(v_k)\}$: Contiene las probabilidades de aparición de los distintos sucesos para cada estado del modelo de Markov oculto. También se conoce como conjunto de probabilidades de emisión. Esta probabilidad nos indica

la probabilidad de que estando en un estado, pasemos por un punto de cruce.

Una vez explicados los componentes del modelo, es necesario explicar cómo se genera cada uno de ellos.

4. Recuperación de recorridos y selección de observaciones

Para el estudio, se utilizó tecnología de localización GPS para obtener trazas de los recorridos habituales de diferentes usuarios. El dispositivo era un Wintec 100 que permitía almacenar hasta 14.000 puntos GPS. La información que provenía de éste debía ser segmentada, filtrada y limpiada, para así conseguir recorridos en vez de datos planos. Finalmente se obtenían rutas como representaciones sin la componente temporal de los recorridos representando un conjunto de éstos mediante un algoritmo de elección de recorrido canónico.

Una vez conseguido el conjunto de rutas representantes, se obtuvieron los puntos que se considerarían observaciones o sucesos. Como hemos comentado anteriormente, serían los puntos previos y posteriores a los de corte entre rutas. Dado que hasta que no estemos en el instante posterior a atravesar ese cruce, no conoceremos cuál de las diferentes alternativas hemos seguido para solucionar la encrucijada, consideraremos los puntos de cruce, como aquellos que están después de atravesar un cruce. Además, dado que en muchos casos las rutas servían de referencia tanto en el recorrido de ida como en el de vuelta, éstas se supusieron como aristas no dirigidas por lo que cada ruta puede tomarse en un sentido o en otro. De ese modo, nuestros puntos de cruce serán aquellos puntos anteriores y posteriores a un cruce (dado que no conocemos a priori el sentido del itinerario).

Analizando los posibles cruces y sus puntos de corte anteriores y posteriores, detectamos 9 tipos representados en la figura 1 para dos rutas. Podemos observar como se trata de una matriz simétrica en la que sólo varían cuál es la ruta 1 y cuál la 2.

R1\R2	Origen	Punto medio	Destino
Origen			
Punto medio			
Destino			

Figura 1: Tipos de cruces entre dos recorridos y sus puntos previos y posteriores.

Cuando existe más de un cruce en puntos muy cercanos, debemos agrupar los cruces para evitar la redundancia y la complejidad del modelo. Hemos llamado a esta operación cluster de cruces y se representa en la figura 2.

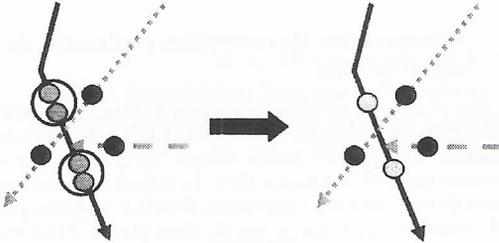


Figura 2: Operación de agrupamiento de puntos previos y posteriores de un cruce cuando están muy próximos.

Para generar esos puntos de cruce, hay que realizar una serie de consultas SQL sobre la base de datos que nos permita determinar cuándo estamos ante un cruce o cuando estamos ante recorridos solapados que pueden parecerse a cruces. En la figura 3 podemos ver un ejemplo de los puntos de cruce localizados.

5. Generación del modelo

Una vez determinados los estados y los puntos de cruce, es necesario generar las matrices de transición y emisión o lo que es lo mismo, crear el modelo. Una vez decididos los estados (destinos finales) y las observaciones (puntos previos y posteriores a un cruce de rutas), estimaremos los valores de las matrices para posteriormente aplicar el algoritmo de Baum-Welch [Baum, 1972] que permitirá estimar los parámetros del modelo que hacen máxima la probabilidad de una secuencia de observaciones usando programación dinámica para mejorar la eficiencia.

Estimación inicial Para realizar la estimación inicial de las matrices, utilizamos un subconjunto de 30 días de recorridos. La frecuencia de las rutas utilizadas y la aparición de puntos GPS que distan menos de un umbral a los puntos de corte de rutas (usando la distancia Haverseno, adecuada para la forma esférica de la tierra) permiten rellenar las matrices. Además, la matriz de estado inicial también se completa utilizando las frecuencias de aparición de los lugares considerados como estados. En la figura 4 podemos ver un ejemplo de la generación inicial del modelo a partir de un conjunto de entrenamiento. En ella podemos observar cómo partiendo del estado inicial S_i X y obteniendo como observación O el símbolo 'b', pasaríamos a dirigirnos al estado final S_f Z . Tras haber pasado el punto previo a cruce 'b', nos encontramos el punto posterior a cruce 'd' lo que hace que pasemos de nuevo al estado Z .

Algoritmo de Baum-Welch Este algoritmo, permite reestimar las matrices de modo que el modelo mejore de una manera iterativa hasta conseguir un máximo local para

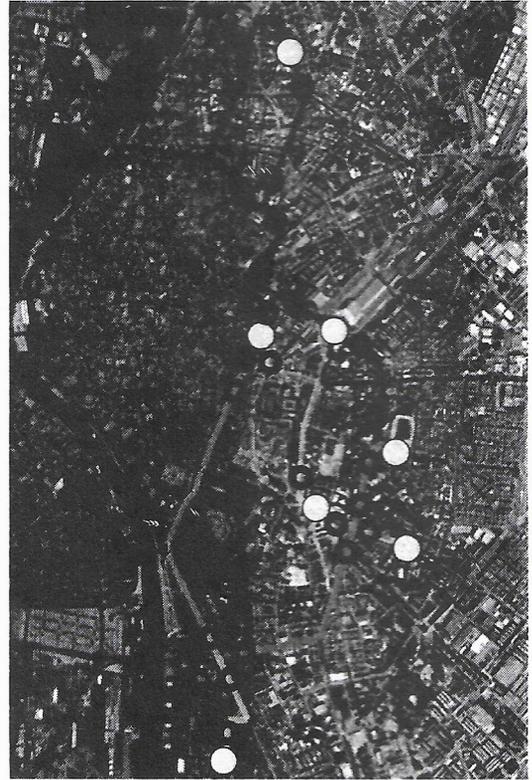


Figura 3: Ejemplo de recorridos modelado como grafo. Cada recorrido tiene un color diferente. Los círculos corresponden con lugares o estados no visibles. Los aros corresponden con las observaciones. En este caso tenemos 7 estados y 18 posibles observaciones.

la probabilidad de una secuencia de observaciones dada. Para aplicarlo, usamos métodos definidos por MATLAB para los modelos de Markov ocultos.

6. Predicción de destinos

Completo ya el modelo de Markov Oculto, pasamos a la predicción del lugar destino al que se dirige un usuario cuando comienza un recorrido nuevo. Para ello, vamos recogiendo aquellas localizaciones GPS derivadas de la ruta seguida por el mismo. Basándonos en el conjunto de los recorridos no usados para el entrenamiento, simulamos su ejecución tal y como lo haría un conductor o transeúnte, avanzando uno a uno por sus puntos GPS. Cuando se recoge un punto suficientemente cercano a una de las observaciones del modelo aplicamos el algoritmo de Viterbi para ver el destino más probable. Dado que la política de elección de observaciones ha hecho que éstas se reduzcan de media a menos de 10 por recorrido, el algoritmo se aplicará menos de 10 veces de media,

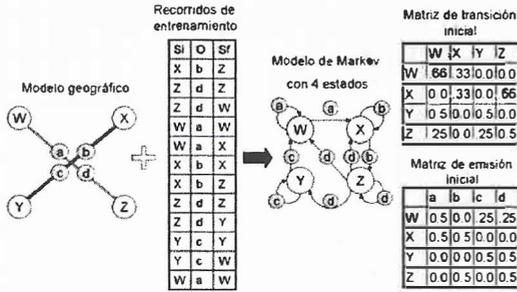


Figura 4: Ejemplo de estimación inicial de matrices a partir de un modelo de recorridos geográfico y un conjunto de entrenamiento.

reduciendo así el número de predicciones repetidas y mejorando el rendimiento.

Para hallar la secuencia de estados más probable, el criterio más ampliamente utilizado consiste en encontrar la mejor secuencia de estados posible considerando globalmente todos los instantes de tiempo, es decir, la secuencia de estados $S = (q_1, q_2, K, q_T)$ que maximiza $P(S|O, \mu)$, lo cual equivale a maximizar $P(S, O|\mu)$. Este es el procedimiento que sigue el algoritmo de Viterbi que se explica a continuación, basado en técnicas de programación dinámica.

Para encontrar la secuencia de estados más probable, $S = (q_1, q_2, K, q_T)$, dada la observación $O = (o_1, o_2, K, o_T)$, consideramos la variable $\delta_t(i)$ definida como

$$\delta_t(i) = \max_{q_1, q_2, K, q_{t-1}} P(q_1, q_2, q_{t-1}, q_t = i, o_1, o_2, K, o_t | \mu) \quad (1)$$

, es decir, $\delta_t(i)$ almacena la probabilidad del mejor camino que termina en el estado i , teniendo en cuenta las t primeras observaciones. Se demuestra que

$$\delta_{t+1}(j) = \left[\max_{1 \leq i \leq N} \delta_t(i) a_{ij} \right] b_j(o_{t+1}). \quad (2)$$

Una vez calculadas las $\delta_t(i)$ para todos los estados y para todos los instantes de tiempo, la secuencia de estados se construye hacia atrás, mediante una traza que recuerda el argumento que maximizó la ecuación 2 para cada instante t y para cada estado j . Esta traza se almacena en las correspondientes variables $\psi_t(j)$. La descripción completa del algoritmo es la siguiente:

1. Inicialización:

$$\delta_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N \quad (3)$$

Se produce cuando nos encontramos el primer punto de cruce, nos indicará el o los caminos más probables (dado que puede haber ramificaciones, lo normal es que al encontrar este primer punto aparezcan diferentes estados equiprobables)

2. Recurrencia:

$$\delta_{t+1}(j) = \left[\max_{1 \leq i \leq N} \delta_t(i) a_{ij} \right] b_j(o_{t+1}) \quad (4)$$

$$\psi_{t+1}(j) = \arg \max_{1 \leq i \leq N} \delta_t(i) a_{ij} \quad (5)$$

donde $t = 1, 2, K, T - 1$ y $1 \leq j \leq N$. Vamos aplicando la técnica por cada punto, reduciendo los posibles destinos más probables.

3. Terminación:

$$q_t^* = \arg \max_{1 \leq i \leq N} \delta_T(i) \quad (6)$$

Para nuestro algoritmo, por cada nuevo punto de cruce, tendremos una predicción posible

4. Reconstrucción de la secuencia hacia atrás:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, K, 1 \quad (7)$$

7. Conclusiones

La elección de los puntos previos y posteriores a los cruces como observaciones proporciona una reducción importante de puntos en el modelo, sin embargo en aquellos recorridos que no se cruzan con ninguno no tendrían observaciones. En esos casos es necesario introducir puntos intermedios que permitan conocer cuándo se está realizando ese recorrido.

Otra consideración importante es que los modelos de Markov no tienen en cuenta el instante temporal en el que se produce la observación por lo que podemos mejorar el modelo duplicando los destinos frecuentes mediante varias etiquetas como trabajo-mañana y trabajo-tarde. El problema que supone es que son necesarios más datos para poder validar el modelo.

8. Agradecimientos

Este trabajo ha sido parcialmente subvencionado por los proyectos FAMENET-InCare (TS12006-13390-C02-02) del Ministerio de Educación y Cultura Español y CUBICO (P06-TIC-02141) de la Junta de Andalucía.

Referencias

[Alvarez et al., 2007] Juan A. Alvarez, Luis González, Juan A. Ortega, and Francisco Velasco. Medidas de similitud entre caminos: Una propuesta. *XI Jornadas JARCA*, jun 2007.

[Ashbrook and Starmer, 2003] Daniel Ashbrook and Thad Starmer. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7:275–286, Oct 2003. doi:10.1007/s00779-003-0240-0.

[Baum, 1972] L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a markov process. *Inequalities*, 3:1–8, 1972.

[Froehlich and Krumm, 2008] Jon Froehlich and Jhon Krumm. Route prediction from trip observations. In *Society of Automotive Engineers (SAE) 2008 World Congress*, 2008.

[Jensen et al., 2004] C.S. Jensen, H. Lahrman, S. Pakalmis, and J. Runge. The infant data. Technical Report TR-79, Aalborg University, TimeCenter, <http://www.cs.aau.dk/TimeCenter>, 2004.

- [Johannesson *et al.*, 2007] Lars Johannesson, Mattias Asborgard, and Bo Egardt. Assessing the potential of predictive control for hybrid vehicle powertrains using stochastic dynamic programming. volume 8, pages 71–83, March 2007.
- [Krumm and Horvitz, 2005] John Krumm and Eric Horvitz. The microsoft multiperson location survey. Technical report, 2005.
- [Krumm and Horvitz, 2006] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In *UbiComp*, pages 243–260, 2006.
- [Krumm, 2008] John Krumm. A markov model for driver turn prediction. In *Society of Automotive Engineers (SAE) 2008 World Congress, 2008.*, 2008.
- [Liao *et al.*, 2004] Lin Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *19th National Conference on Artificial Intelligence (AAAI 2004)*. San Jose, CA, USA., 2004.
- [Liao *et al.*, 2005] Lin Liao, D. Fox, and H. Kautz. Location-based activity recognition. In *Advances in Neural Information Processing Systems 19*, 2005.
- [Marmasse, 2004] Natalia Marmasse. *Providing Lightweight Telepresence in mobile communication to enhance collaborative living*. PhD thesis, 2004.
- [Patterson *et al.*, 2004] Donald J. Patterson, Lin Liao, Krzysztof Gajos, Michael Collier, Nik Livic, Katherine Olson, Shiaokai Wang, Dieter Fox, and Henry Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In *UbiComp 2004: Ubiquitous Computing*, 2004.
- [Rabiner and Juang, 1986] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, Jan 1986.
- [Simmons *et al.*, 2006] Reid Simmons, Brett Browning, Yilu Zhang, and Varsh Sadekar. Learning to predict driver route and destination intent. In *IEEE Intelligent Transportation Systems Conference*, sep 2006.
- [Terada *et al.*, 2006] Tsutomu Terada, Masakazu Miyamae, Yasue Kishino, Kohei Tanaka, Shojiro Nishio, Takashi Nakagawa, and Yoshihisa Yamaguchi. Design of a car navigation system that predicts user destination. In *IEEE 7th International Conference on Mobile Data Management (MDM'06)*, 2006.
- [Torkkola *et al.*, 2007] Kari Torkkola, Keshu Zhang, Haifeng Li, Harry Zhang, Christopher Schreiner, and Mike Gardner. Traffic advisories based on route prediction. In *Workshop on Mobile Interaction with the Real World (MIRW 2007)*, 2007.