Evaluation of Clustering Techniques to Estimate the Effective Bandwidth of a Markovian Fluid from Traffic Traces

Carina N. Fernández, José M. Bavio, and Beatriz S. Marrón

***Abstract*—Integrated services digital networks, designed to transport data in real time, are modeled by a multiplexer system, where several data fluids share a single output. At the time of admission of a new connection, in order to maintain the quality of service (QoS), it is important to know the amount of available resources required by the connections sharing the channel. It is therefore important to have models for the sources and techniques that provide accurate estimates of the resources required for each of them. It is assumed that the network sources are modeled using the Generalized Markovian Fluid Model (GMFM) because of its versatility in describing traffic fluctuations. This is a Markovian fluid where the transfer rate is a random variable whose range and probability distribution are determined by the state of the modulating chain. To measure resource allocation, the concept of Effective Bandwidth (EB) is used, since it allows expressing this magnitude as a function of the model parameters, which will be estimated from traffic traces. As the size of the data to describe the behavior of a source is clearly enormous, they are studied using clustering techniques. In this work, different methods, supervised and unsupervised, are presented to estimate the parameters involved in the calculation of the EB. Finally, the performance of the estimators is analyzed by calculating partition comparison indices, corresponding to the dispatch intervals, which are based on the confusion matrix and the notion of mutual information entropy between partitions, on simulated data.**

***Index Terms*—Markovian fluid, Effective bandwidth, Supervised and unsupervised estimation methods, Clustering.**

I. Introducción

Cuando se trabaja con varios servicios agregados en una red de telecomunicaciones, se debe recurrir a una red digital de servicios integrados. La integración significa que la red puede transportar muchos tipos de información, como voz, vídeo y datos, en forma digital, utilizando una única infraestructura, por lo que los recursos son compartidos por un conjunto de fuentes heterogéneas. El problema radica en estimar las necesidades de recursos de cada fuente ya que, si muchas despachan la tasa máxima, habría probabilidad no nula de desborde del buffer. Desde la noción de Ancho de Banda Efectivo (EB) introducida por Kelly en 1996 [1], ha surgido con fuerza el desarrollo de herramientas estadísticas que permiten encontrar expresiones para estimar la probabilidad de pérdida en un enlace. En necesario entonces, de contar con Criterios de Control de Admisión (CAC) para aceptar una nueva conexión que minimice los efectos de pérdida de datos y mantenga la Calidad de Servicio (QoS), posibilitando el acceso más universal y asequible a Internet [2]. En los últimos años, especialmente en pandemia, el consumo de datos por contenidos audiovisuales y gaming, streeming, entre otros, ha aumentado de manera significativa, manteniendo el tema vigente [3], [4]. Surge así la necesidad de disponer de modelos matemáticos que describan el comportamiento de cada fuente, para poder dimensionar, entre otras, las componentes de la red, evaluar su rendimiento y encontrar los descriptores adecuados que caractericen el servicio [5]. Es importante disponer de buenas estimaciones para estas magnitudes, utilizando trazas de tráfico que reflejen el comportamiento de la fuente, y como la cantidad de datos de las mismas es realmente grande, se los suele agrupar en intervalos de velocidades de despacho. Se desea estimar los parámetros del modelo involucrados en el cálculo del EB, tomando como punto de partida la estimación de los estados de la cadena aplicando técnicas de clustering, actualmente objeto de investigación activa en diversas áreas científicas. Dichas técnicas se utilizan para comprender los datos masivos utilizando métodos de agrupación y dependen en gran medida de la elección de la distancia adecuada que refleje la proximidad de los objetos que, en muchos casos, obedece a la naturaleza de los datos [6].

Este trabajo extiende los resultados obtenidos en [7] proporcionando como novedad un nuevo método de estimación no supervisado, cuya implementación no requiere de ninguna información previa acerca del modelado de la fuente, propiciando una herramienta útil en vistas del trabajo futuro: aplicar estas técnicas a trazas de tráfico reales. Además, aporta un estudio comparativo de las estimaciones mediante el cálculo de sendos índices de comparación basados en la matriz de confusión y la noción de entropía de la información: El primero, relevante porque se conoce su distribución permitiendo realizar estadística inferencial [8], mientras que el segundo requiere de menos supuestos para su aplicación y da buen resultado cuando los clusters son muy desbalanceados como suele ocurrir en este tipo de datos [9].

Se estructura de la siguiente manera: En la Sección II presentamos el Modelo de Flujo Markoviano Generalizado (GMFM) y proporcionamos una expresión para determinar su EB. En la Sección III describimos tres métodos que utilizamos para estimar el EB: Estimación de la Densidad por Núcleos (KDE), Modelo de Mezcla Gaussianas (GMM) y Affinity Propagation (AP); y describimos dos métodos que utilizamos para determinar el número óptimo clusters, para nosotros la cantidad de intervalos de despacho, siendo estos el método de

Carina N. Fernández, José M. Bavio and Beatriz S. Marrón are with Mathematics Department, Universidad Nacional del Sur. e-mail: carina.fernandez@uns.edu.ar.

Silhouette y el Método del Codo. En la Sección IV mostramos los parámetros utilizados en la simulación de las trazas y los algoritmos implementados para las estimaciones, cuyos resultados se presentan en la Sección V. En la Sección VI evaluamos la performance de los estimadores obtenidos, mediante el cálculo de dos índices: el Mínimo Error de Clasificación (CEM) y el de Información Mutua (MI). Finalmente, en la Sección VII presentamos conclusiones, junto con algunas consideraciones para trabajos futuros.

II. Descripción del tráfico

A. El Modelo

Los modelos de flujos markovianos se han desarrollado durante más de dos décadas y han sido especialmente útiles para modelar, con relativa precisión, muchas fuentes de datos reales porque permiten captar la correlación temporal de los datos. El más sencillo es el modelo On/Off, especialmente utilizado en el tráfico de voz, extendido para modelar todo tipo de tráfico en Internet mediante el multiplexado de varias fuentes On/Off, que, si bien genera una mejora sustancial, puede aumentar mucho su dimensión. Asumimos que las fuentes en la red se modelan mediante el GMFM [10], dado que es versátil para describir las fluctuaciones del tráfico. El mismo puede ser interpretado como una forma de reducir el modelo de flujo Markoviano cuando su dimensión es muy grande, agrupando tasas de despacho similares como un ruido alrededor de un valor central, por eso es adecuada modelarlas mediante una distribución gaussiana. Se trata de un modelo modulado por una cadena de Markov continua, homogénea e irreducible en el tiempo, donde la tasas de transferencia son variables aleatorias cuyos rangos y distribuciones de probabilidad están determinados por los estados de la cadena modulante, los cuales pueden interpretarse como un tipo de actividad realizada por un usuario, como el chat o la videollamada. Un cambio repentino en la tasa de transferencia, puede identificarse como un cambio de estado en la cadena.

B. Ancho de Banda Efectivo

Dada una QoS esperada, interpretada como la probabilidad de desbordamiento del buffer, el EB de una fuente de tráfico es una medida realista de la ocupación del canal que cuenta con propiedades que la hacen interesante en la aplicación al estudio de redes de datos [1], entre ellas, por ser una magnitud que efectivamente se encuentra entre la tasa máxima y la tasa media de despacho. Otro aspecto importante es que puede expresarse en función de los parámetros del modelo, que para el GMFM son: Q el generador infinitesimal de la cadena, π la distribución invariante y H una matriz diagonal que contiene las tasas medias de despacho de cada estado de la cadena. En [10] se da la fórmula explícita del EB para el GMFM:

$$\alpha(s,t) = \frac{1}{st} \log \left\{ \pi \exp \left[(Q + sH)\, t \right] \mathbf{1} \right\}, \qquad (1)$$

donde $\mathbf{1}$ es un vector columna con todas las entradas iguales a 1.

III. Métodos de estimación

A continuación describiremos brevemente tres métodos de estimación, supervisados y no supervisados, que utilizamos para hallar los estimadores $\hat{Q}$, $\hat{\pi}$ y $\hat{H}$, involucrados en (1). El método de KDE, seleccionado porque su estimador cuenta con propiedades importantes como consistencia y es asintóticamente insesgado [11], el GMM elegido principalmente por su relación con el GMFM, cuya relación puede verse en la Sección A, y el método de AP porque no requiere información previa acerca del modelo, como la cantidad de intervalos de despacho.

A. Estimador de la Densidad por Núcleos

Sea una muestra aleatoria simple $X_1, \ldots, X_n$ de la variable X, con función de densidad f. El *Estimador de la Densidad por Núcleos* fue propuesto por Rosenblatt [12] como:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h} \right),$$

donde a K se lo denomina núcleo y es cualquier función integrable tal que $\int_{\mathbf{R}} K(x)\mathrm{d}x = 1$ y a h se lo suele denominar parámetro de suavizado o ancho de ventana, que verifica que $h \to 0$ y $nh \to \infty$ para asegurar que $\hat{f}$ tienda a la verdadera densidad f. Este parámetro suele ser un punto crucial en la estimación, dado que se encuentra altamente relacionado con el parámetro de suavizado.

Existe una relación entre las propiedades del estimador y la elección tanto del ancho de la ventana, como del núcleo. Por un lado, ventanas demasiado pequeñas conducirán a estimadores variables, ya que en cada punto los entornos tendrán insuficientes observaciones en las cuales basar la estimación, mientras que ventanas demasiado grandes conducirán a estimadores suaves, impidiendo captar la estructura local de la densidad, dando lugar posiblemente a estimadores sesgados. En [11] se discuten los valores apropiados para h.

B. Modelo de Mezcla de Gaussianas

En el caso de la presencia de subpoblaciones dentro de una misma población, no sería recomendable explicar la distribución de los datos mediante una única distribución estadística. Es necesario utilizar una composición de distribuciones, que suele describirse mediante modelos de mezcla, los cuales quedan determinados por los parámetros de cada componente de la mezcla y las proporciones en las que cada una de ellas contribuye a la distribución general. El conjunto de parámetros que definen a estos modelos puede ser estimado mediante Estimadores de Máxima Verosimilitud (MLE).

El modelo multimodal *Modelo de Mezclas Gaussianas*, introducido por Duda y Hart [13], considera una función de densidad f como una mezcla de k componentes Gaussianas simples. El método consiste en particionar un conjunto de datos en k grupos o clusters, de manera que cada dato pertenezca sólo a uno de ellos. La siguiente ecuación define una mezcla Gaussiana para una variable observada $\mathbf{x} = (x_1, \ldots, x_n)$, donde x_i pertenece al conjunto de n datos observados:

$$f(\mathbf{x}) = \sum_{i=1}^{k} \pi_i f_{\theta_i}(\mathbf{x}),$$

donde $f_{\theta_i}(\mathbf{x}) \sim N(\mathbf{x}|\mu_i, \Sigma_i)$. Cada Gaussiana explica los datos contenidos en cada grupo de manera individual, mientras que los coeficientes de mezcla π_i, que representan la probabilidad a priori de que una observación pertenezca al grupo i, deben verificar que $\sum_{i=1}^{k} \pi_i = 1$.

Para estimar f debemos hallar los valores de los parámetros $\theta = (\pi_i, \theta_i)_{i=1}^{k}$ que pertenecen al espacio de parámetros Θ definido por:

$$\Theta = \{\theta = (\pi_i, \theta_i)_{i=1}^{k} : \sum_{i=1}^{k} \pi_i = 1, 0 \leq \pi_i \leq 1, \theta_i = (\mu_i, \Sigma_i),$$
$$\mu_i \in \mathbf{R}^n, \Sigma_i \in \mathbf{R}^{n \times n} \text{ definida positiva, } i = 1, \ldots, k\}.$$

Resolver este problema, implica encontrar el θ que maximice esta función, que será el MLE, dado por:

$$\hat{\theta}_{MLE} = \arg\max_{\theta \in \Theta} \sum_{j=1}^{n} \log \left(\sum_{i=1}^{k} \pi_i f_{\theta_i}(x_j) \right).$$

Como no hay solución analítica a este problema por desconocer los parámetros, utilizamos para estimarlos el algoritmo iterativo de Esperanza-Máxima, introducido en [14], que consta de dos pasos: en el primero se encuentra una expresión para el valor esperado de la log-verosimilitud, dados los valores iniciales o la estimación previa de los parámetros, y en el segundo paso se maximiza esa esperanza sobre el espacio de parámetros. Utilizamos este algoritmo porque es numéricamente estable, de fácil implementación y bajo costo de cada iteración y por tener una convergencia global fiable [11].

C. *Affinity Propagation*

Dado un conjunto de datos $X = \{x_1, x_2, \ldots, x_n\}$ y una medida de similitud $s(x_i, x_j)$ entre el punto x_i y el punto x_j, pretendemos agrupar los datos en m $(m < n)$ clusters, cada uno representado por un ejemplar de X. Estos ejemplares forman un subconjunto del conjunto de datos original, llamémoslo $X_e = \{e(x_1), \ldots, e(x_m)\} \subset X$, donde $e(x_i)$ es el ejemplar de cada punto x_i en X. El objetivo es maximizar la suma de similitudes entre cada dato y su ejemplar dada por:

$$S(X, X_e) = \sum_{i=1}^{n} s(x_i, e(x_i)).$$

Frey y Dueck [15] propusieron el método de AP para encontrar ejemplares a partir de un conjunto de datos, de modo que la suma de las distancias entre los puntos y sus ejemplares sea mínima. El procedimiento consiste en el intercambio de mensajes entre puntos, potenciales soluciones, hasta que una condición de convergencia sea alcanzada. Los mensajes son de dos tipos, de responsabilidad y de disponibilidad, y son intercambiados entre los puntos indicando la afinidad que cada uno de ellos tiene respecto de otro para actuar como su ejemplar. El mensaje de responsabilidad, que notaremos $r(i, k)$, se envía desde x_i a x_k y refleja lo bien que x_k sirve como ejemplar de x_i considerando otros ejemplares potenciales para x_i. Los mensajes de disponibilidad, que notaremos $a(i, k)$, se emiten desde x_k a x_i, y refleja lo apropiado que es que x_i elija a x_k como su ejemplar, considerando otros puntos potenciales que pueden elegir a x_k como su ejemplar. La información se actualiza de forma iterativa como sigue:

$$r(i, k) \leftarrow s(x_i, x_k) - \max_{k' : k' \neq k} \{a(i, k') + s(x_i, x_{k'})\},$$

$$a(i, k) \leftarrow \min \left\{ 0, r(k, k) + \sum_{i' : i' \notin \{i, k\}} \max\{0, r(i', k)\} \right\},$$

mientras que la autodisponibilidad se actualiza como:

$$a(k, k) \leftarrow \sum_{i' : i' \neq k} \max\{0, r(i', k)\}.$$

Tras la convergencia, el ejemplar para cada punto x_i se elige como $e(x_i) = x_k$ donde k maximiza el siguiente criterio:

$$arg \max_{k} a(i, k) + r(i, k).$$

El algoritmo AP utiliza la matriz de similitud completa para realizar la propagación, por lo que en cada etapa hay n^2 pares de datos cuyos valores de responsabilidad y disponibilidad deben ser calculados, resultando en un costo computacional del $O(n^2 T)$, donde T el número de iteraciones. Esto afecta en gran medida a la velocidad del algoritmo, especialmente cuando la cantidad de datos es grande.

La principal ventaja del algoritmo AP es que no necesita de antemano el número de clusters. Esto se debe a que considera cada punto como un ejemplar potencial y la probabilidad de ser un ejemplar depende del valor compartido de la preferencia.

D. *El Problema del Número Óptimo de Clusters: Método de Silhoette & Método de Codo*

El Ancho de Silueta (SW) se utiliza para medir cuán similar es un objeto a su propio cluster en comparación con otros clusters [16], y es independiente del número de agrupaciones, k. Para una observación x_i, el SW se define como:

$$s_{x_i} = \begin{cases} 1 - \frac{a_{x_i}}{b_{x_i}} & , \text{ si } a_{x_i} < b_{x_i} \\ 0 & , \text{ si } a_{x_i} = b_{x_i} \\ \frac{b_{x_i}}{a_{x_i}} - 1 & , \text{ si } a_{x_i} > b_{x_i} \end{cases}, \tag{2}$$

donde a_{x_i} representa la distancia media entre x_i y todos los demás puntos en el mismo cluster. Si para x_i y cualquier grupo C_m que no lo contenga, calculamos la distancia promedio de x_i a todos los elementos pertenecientes al grupo dado, b_{x_i} será al mínimo de dichos valores para todos los grupos.

De (2) se deduce que s_{x_i} toma valores entre -1 y 1: un valor cercano a 1 significa que los datos están apropiadamente agrupados y si está próximo a -1 sería más conveniente que x_i se agrupara en su cluster vecino. Un valor cercano a 0 significa que el dato está en la frontera de dos clusters.

Es posible obtener una medida general de la bondad de la agrupación calculando el Ancho de Silueta Promedio (ASW), esto es,

$$\bar{s}_k = \frac{1}{n} \sum_{i=1}^{n} s_{x_i},$$

donde n es el número total de datos observados. El valor de k óptimo es el *Coeficiente de Silueta* (SC), introducido por Kaufman y Rousseeuw [17] como:

$$SC = \max_{k} \bar{s}_k,$$

donde el máximo se toma sobre todos los k para los que se pueden construir las siluetas, esto es, $k = 2, 3, \ldots, n-1$.

Para tomar una mejor decisión, el ASW se suele estudiar en conjunto con los Gráficos de Silueta, los cuales se realizan individualmente para cada valor de k. Para cada cluster C_i, $i = 1, \ldots, k$, se grafica una línea horizontal de longitud igual a s_{x_j}, calculado con (2), en orden decreciente según s_{x_j}, siendo x_j, con $j = 1, \ldots, n$ los elementos que se encuentran dentro del cluster C_i. La otra dimensión de una silueta es su altura, que simplemente es igual al número de objetos en C_i. Una silueta ancha indica valores grandes de s_{x_j} y, por lo tanto, un grupo pronunciado. De esta forma, todo el agrupamiento se puede mostrar por medio de un gráfico único, lo que nos permite distinguir los conglomerados "claros" de los "débiles". Se busca que los gráficos de silueta tengan las puntuaciones de los grupos que superen el ASW, con un grosor mayormente uniforme y sin grandes fluctuaciones en el tamaño.

En conjunto con el ASW y sus gráficos, se estudia el *Método de Codo*. Se trata de un método visual para probar la consistencia del número óptimo de clusters. Consiste en un gráfico de líneas donde en el eje horizontal se representan los posibles valores de k y en el eje vertical su correspondiente Suma de Cuadrados de los Errores (*SSE*), definido por:

$$SSE = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - c_i\|^2 ,$$

donde x_j es el j-ésimo elemento en C_i y c_i su centroide.

Al igual que en el método Silhouette, se elige previamente un rango de valores candidatos de k. Una SSE baja indica una mejor calidad de agrupación para particiones con el mismo k. El valor de k elegido será aquel donde cae repentinamente la distancia promedio. El gráfico de líneas parece un brazo, por lo que el "codo" en el brazo es el valor óptimo de k, donde la disminución de la SSE comienza a parecer lineal.

IV. Simulación a partir de trazas

A. Parámetros del Modelo

Se realizaron simulaciones de tráfico, generadas por los algoritmos de Cadenas de Markov Monte Carlo, según el modelo presentado en la Sección A. La cadena de Markov modulante tiene 9 estados, cada uno asociado a un intervalo de velocidad de transferencia de datos, tal como se muestra en el Cuadro I.

CUADRO I
RANGOS TEÓRICOS DE DESPACHO.

Estado	Velocidad de transferencia (Mbps)
1	$(0, 1024]$
2	$(1024, 2048]$
3	$(2048, 3072]$
4	$(3072, 4096]$
5	$(4096, 5120]$
6	$(5120, 6144]$
7	$(6144, 7168]$
8	$(7168, 8192]$
9	$(8192, 10240]$

Para diseñar el generador infinitesimal Q de la cadena, consideramos que ésta puede pasar de un estado a otro con la misma probabilidad, para no introducir una dificultad adicional en la interpretación de los resultados obtenidos de las estimaciones, según los distintos métodos, facilitando la comparación entre ellos.

$$Q = \begin{pmatrix} -8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -8 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -8 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -8 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -8 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -8 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & -8 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -8 \end{pmatrix}.$$

Por lo mencionado en la Sección A, dentro de cada intervalo, asumimos que la cantidad efectivamente despachada se sortea mediante una distribución normal con media igual al punto medio y desvío igual a un sexto de su longitud. La matriz diagonal H contiene en su diagonal principal los valores medios de estas distribuciones.

La traza simulada es una sucesión de pares (v_i, t_i), donde v_i es la velocidad de transferencia, t_i es el momento en que la cadena salta a otro estado, por lo que el enlace se transfiere a velocidad v_i mientras $t_{i-1} < t < t_i$, para $i = 1, \ldots, 14000$, número de saltos de la cadena. En todos los algoritmos estimamos Q mediante MLE como en [10].

B. Algoritmos Implementados para la Estimación

Para $i = 1, \ldots, 14000$, para $j = 1, \ldots, 9$, utilizamos los siguientes algoritmos para estimar el EB según cada método:

Algoritmo 1: Estimación del EB por KDE.
Input: (v_i, t_i). $h = 200$ {Ancho de ventana}. Núcleo Gaussiano.
1: Aplicar KDE a los v_i, para obtener $\hat{f}$.
2: Estimar los intervalos de despacho con los mínimos de la densidad estimada, considerando el valor 0 como el límite inferior del primer intervalo, y 10240 como el máximo.
3: Estimar las tasas medias de despacho h_j, con los puntos medios de los intervalos de despacho, por la simetría de la distribución, y construir la matriz $\hat{H}$.
4: Estimar las π_j, con el área bajo la densidad $\hat{f}$ para cada intervalo de despacho.
5: Recorrer la traza comparando cada v_i con el rango estimado para asignar el estado correspondiente, para obtener la cadena estimada $(\hat{v}_i, t_i)$.
6: Hallar el estimador del generador infinitesimal, $\hat{Q}$.
7: Calcular el EB estimado con $\hat{H}$, $\hat{\pi}$, y $\hat{Q}$, como en (1).

Algoritmo 2: Estimación del EB por GMM.
Input: (v_i, t_i). $k = 9$ {Cantidad de clusters}
1: Aplicar GMM a los v_i, para obtener las medias $\hat{\mu}_j$, las varianzas $\hat{\sigma}_j^2$, y los pesos $\hat{\pi}_j$ de la mezcla.
2: Construir la matriz $\hat{H}$ con $\hat{h}_j = \hat{\mu}_j$.
3: Reconstruir los intervalos de velocidades de despacho utilizando el 99 % del área de cada distribución Gaussiana, centrada en $\hat{h}_j$ y con varianza $\hat{\sigma}_j^2$. Considerar el valor 0 como el límite inferior del primer intervalo.
4: Estimar π con los pesos $\hat{\pi}_j$.
5: Paso 5, 6, 7 del Algoritmo 1.

Algoritmo 3: Estimación del EB por AP.
Input: (v_i, t_i). Preferencia$= -210000000$.

1: Aplicar AP a los v_i, para obtener los ejemplares e_j y etiquetas l_j para cada v_i.
2: Reconstruir el intervalo j de velocidades de despacho, utilizando el mínimo y máximo de los valores de v_i correspondientes a cada etiqueta l_j, para cada j.
3: Estimar π, considerando $\hat{\pi}_j = \frac{\#\text{elementos en el intervalo } j}{14000}$, por la ergodicidad del modelo.
4: Paso 5, 6, 7 del Algoritmo 1.

V. Resultados numéricos de las estimaciones

Estimamos el EB aplicando los métodos y algoritmos descriptos en las Secciones III y IV con Python 3.7, haciendo uso de la biblioteca sklearn.neighborslibrary [18]. Los códigos se pueden proporcionar contactando a los autores.

Como el GMM requiere conocer previamente el número de cluster k, realizamos el análisis de agrupación de las velocidades de despacho para k entre 7 y 11, aplicando el método de Silhouette.

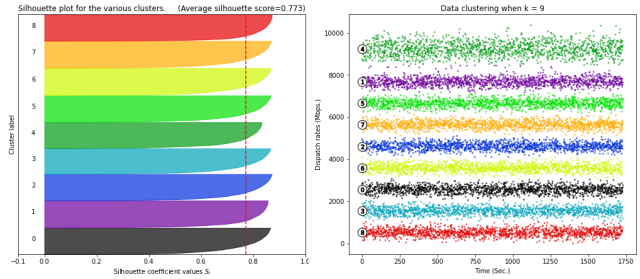

Fig. 1. Análisis de Silhouette para $k = 9$.

En la Fig. 1 el gráfico de la izquierda se corresponde con el gráfico de silueta para $k = 9$, construido según la Sección D: hay 9 siluetas, representando la agrupación de las velocidades en 9 rangos de despacho. La línea vertical roja indica el valor del ASW. El gráfico de la derecha muestra la agrupación de las velocidades de despacho en 9 intervalos. Omitimos los casos para k igual a 7, 8, 10 y 11 por la presencia, tanto de grupos con puntuaciones de silueta inferiores a la media, como de grandes fluctuaciones en los SW. El valor de $k = 9$ resulta ser el más apropiado, por tener una puntuación de la silueta de cada grupo por encima de la media y por presentar SW similares para cada grupo.

La Fig. 2 contiene las gráficas de los ASW (en rojo) y los valores de la SSE (en azul), para cada número posible de clusters. En el gráfico azul, el codo marca el punto en el que el gráfico rojo exhibe su máxima curvatura, por lo que tendríamos el mayor ASW. Antes de llegar a ese punto, un aumento del número de clusters ayuda a reducir la SSE. Es de esperar que, después del codo encontremos rendimientos decrecientes: las reducciones incrementales de la SSE, añadiendo más clusters, se harían más insignificantes cuanto más nos alejemos del codo y lo harían relativamente más rápido tras haber pasado el punto de inflexión de la curva, es decir, su codo. Este valor se alcanza cuando los datos se agrupan en 9 clusters, valor del mayor ASW, por lo que $k_{opt} = 9$.

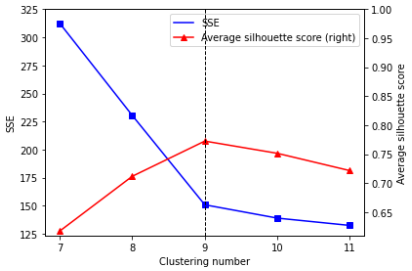

Fig. 2. SSE y ASW vs Número de clusters.

El Cuadro II contiene las estimaciones de los intervalos de despacho y el Cuadro III las tasas medias de despacho estimadas para cada método.

CUADRO II
RANGOS TEÓRICO Y ESTIMADOS.

Rango teórico	Rango estimado		
	KDE	GMM	AP
(0,1024]	(0,1026.109]	(0,1024.106]	(0,1023.916]
(1024,2048]	(1026.109,2048.361]	(1024.106,2044.548]	(1023.916,2038.102]
(2048,3072]	(2048.361,3072.296]	(2044.548,3078.494]	(2038.102,3068.786]
(3072,4096]	(3072.296,4102.291]	(3078.494,4098.483]	(3068.786,4092.832]
(4096,5120]	(4102.291,5123.870]	(4098.483,5124.649]	(4092.832,5105.880]
(5120,6144]	(5123.8701,6142.421]	(5124.649,6140.820]	(5105.88,6122.922]
(6144,7168]	(6142.421,7162.654]	(6140.820,7156.276]	(6122.922,7159.159]
(7168,8192]	(7162.654,8287.233]	(7156.276,8189.639]	(7159.159,8453.649]
(8192,10240]	(8287.233,10240.000]	(8189.639,10240.000]	(8453.649,10240.000]

CUADRO III
MEDIAS TEÓRICAS Y ESTIMADAS.

Medias teóricas	Medias estimadas		
	KDE	GMM	AP
512	510.774	513.239	512.852
1536	1539.086	1535.495	1540.726
2560	2572.783	2559.703	2556.341
3584	3584.601	3585.396	3583.004
4608	4605.507	4611.103	4612.448
5632	5631.799	5634.654	5635.215
6656	6654.725	6653.186	6654.240
7680	7674.285	7676.488	7686.066
9216	9160.708	9223.279	9234.707

Para evaluar la performance en la estimación de Q, calculamos el Mapa de Calor de los errores de estimación cometidos en cada método, mostrados en la Fig. 4. Cada elemento representa el porcentaje del error de estimación. Valores negativos indican una subestimación y los positivos una sobreestimación de las componentes del generador infinitesimal.

Hallamos la Matriz de Confusión para cada método a fin de evaluar la performance en la estimación de los estados de la cadena, mostrados en la Fig. 5. Las filas representan los estados reales y las columnas los estimados, lo que nos permite visualizar los aciertos y errores de estimación.

Utilizando las estimaciones de Q, π, y H y (1), estimamos el EB para cada método. En la Fig. 3 se muestra una comparación del EB teórico vs los estimados.

VI. Comparación de los resultados obtenidos

Para cuantificar el error de clasificación cometido en cada método utilizamos dos medidas: el índice de validación Míni-

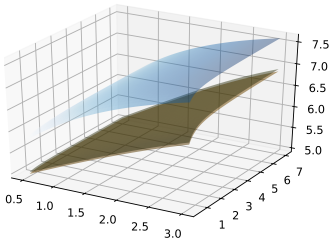

Fig. 3. EB teórico (azul), EB estimado por KDE (naranja) por GMM (rojo) y por AP (verde).

mo Error de Clasificación y la Información Mutua entre dos particiones.

A. *Mínimo Error de Clasificación*

El Mínimo Error de Clasificación, introducido por Meila [8], mide la "distancia" entre dos particiones según el error de clasificación de una partición respecto de otra. Dado que las etiquetas de las particiones son arbitrarias, la medida considera el mínimo error permutando las etiquetas de las observaciones de todas las maneras posibles.

Consideremos D un conjunto de n datos. Sean $\mathscr{C} = \{C_1, \ldots, C_K\}$ y $\mathscr{C}' = \{C'_1, \ldots, C'_{K'}\}$ dos particiones de D, $K \leq K'$, y supongamos que en C_k hay n_k puntos y en $C'_{k'}$ hay $n'_{k'}$ puntos.

Este criterio de comparación de clusters puede describirse mediante la matriz de confusión entre $\mathscr{C}$ y $\mathscr{C}'$. El elemento k, k'-ésimo representa la cantidad de datos que hay en la intersección de los clusters C_k de $\mathscr{C}$ y $C'_{k'}$ de $\mathscr{C}'$, esto es,

$$n_{k,k'} = |C_k \cap C'_{k'}|\,.$$

A cada cluster de $\mathscr{C}$ se le da una "mejor coincidencia" en $\mathscr{C}'$. Luego, el CEM se calcula como la masa de probabilidad total "no coincidente" en la matriz de confusión:

$$CEM = 1 - \frac{1}{n} \max_{\sigma} \sum_{k=1}^{K} n_{k,\sigma(k)}, \tag{3}$$

donde σ es un mapeo inyectivo de $\{1, \ldots, K\}$ en $\{1, \ldots, K'\}$.

Para cada σ tenemos una correspondencia entre las etiquetas de los clusters en $\mathscr{C}$ y $\mathscr{C}'$. Si consideramos la agrupación como una tarea de clasificación con la correspondencia de etiquetas fija, calculamos el error de clasificación de $\mathscr{C}'$ con respecto a $\mathscr{C}$. El mínimo error de clasificación posible bajo todas las correspondencias es precisamente CEM.

B. *Índice de Información Mutua*

La Información Mutua, fue introducida por Shannon [19] y mide la dependencia entre dos particiones. Se basa en la teoría de información y en la noción de entropía como medida de incertidumbre. La entropía puede definirse como la probabilidad de que un elemento esté en el cluster $C_k \in \mathscr{C}$:

$$P(k) = \frac{|C_k|}{n}.$$

Luego, la entropía asociada a la partición $\mathscr{C}$ con K clases está dada por:

$$H(\mathscr{C}) = -\sum_{k=1}^{K} P(k) \log_2 P(k).$$

Es posible extender la noción de entropía de una partición a la información mutua entre dos de ellas, cuando el objetivo es medir la dependencia entre dos particiones, interpretándola como la reducción de incertidumbre de una partición que se produce, debido al conocimiento de otra partición del mismo conjunto de datos. La MI entre dos particiones $\mathscr{C}$ y $\mathscr{C}'$ se define como:

$$MI(\mathscr{C}, \mathscr{C}') = \sum_{k=1}^{K} \sum_{k'=1}^{K'} P(k, k') \log_2 \frac{P(k, k')}{P(k)P(k')}, \tag{4}$$

donde $P(k, k')$ es la probabilidad de que un elemento esté en el cluster C_k de $\mathscr{C}$ y $C'_{k'}$ de $\mathscr{C}'$. MI toma el valor 0 cuando las particiones son independientes, dado que no se aportan información entre ambas particiones y el valor 1 si las particiones son idénticas, ya que toda la información de la primera partición es compartida por la segunda.

C. *Resultados Obtenidos*

Calculamos el CEM con (3) y el índice de MI con (4), comparando la partición teórica de los datos simulados, con las particiones obtenidas por cada método.

CUADRO IV
RESULTADOS DE LOS CEM Y MI PARA CADA MÉTODO.

Medidas	Métodos		
	KDE	GMM	AP
CEM	0.00165	0.00170	0.00270
MI	0.99214	0.99194	0.98880

VII. CONCLUSIONES Y TRABAJO FUTURO

Observando las estimaciones de los parámetros podemos apreciar que todos los métodos proporcionaron buenos resultados, siendo KDE y GMM los más precisos. Esto se evidencia en principio, en la Fig. 3, mostrando el EB estimado para estos métodos más próximos al teórico, en particular para KDE esta diferencia es visualmente imperceptible. Los errores de clasificación de estados, calculados en base a la Fig. 5, resultaron ser $2{,}36 * 10^{-3}$ para KDE, $2{,}43 * 10^{-3}$ para GMM y $3{,}86 * 10^{-3}$ para AP, en concordancia con lo mencionado previamente. Los valores obtenidos de los índices de comparación de particiones apoyan estas conclusiones, mostrando el CEM más bajo y el de MI más alto para KDE y GMM, siendo el primero mínimamente superador con una diferencia absoluta de $5 * 10^{-5}$, mientras que el segundo la misma es de $2 * 10^{-4}$.

Como trabajo futuro inmediato, consideraremos modificar el generador infinitesimal para mostrar supuestos más realistas en el modelo, donde por ejemplo, sea más habitual saltar de un estado a los estados adyacentes o bien al de tasa de transferencia máxima o al de transferencia mínima o nula. Como estudio a largo plazo, trabajaremos con datos de tráfico reales, y de acuerdo a la naturaleza de los mismos, estimaremos el EB aplicando distintos métodos.

AGRADECIMENTOS

Los autores agradecen al Proyecto de Grupo de Investigación PGI 24/L112 de la Universidad Nacional del Sur por financiar parcialmente este trabajo.

Fig. 4. Mapas de calor para el error de estimación de Q.

(a) KDE (b) GMM (c) AP

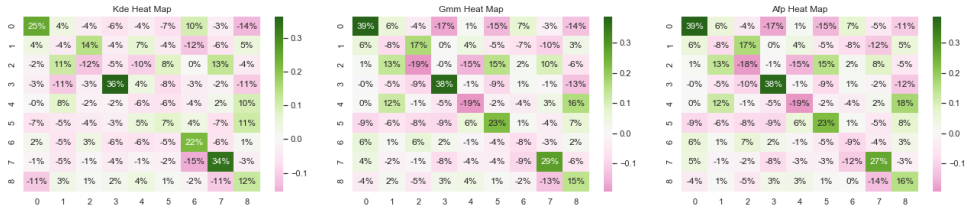

Fig. 5. Matrices de confusión de los estados estimados.

(a) KDE

$$\begin{pmatrix} 1535 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1494 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1540 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1602 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 1540 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1574 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1554 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1573 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 1555 \end{pmatrix}$$

(b) GMM

$$\begin{pmatrix} 1535 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1494 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1540 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1600 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1541 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1574 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1552 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1573 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1557 \end{pmatrix}$$

(c) AP

$$\begin{pmatrix} 1535 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1494 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1540 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1601 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1541 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1574 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1554 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1573 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 24 & 1534 \end{pmatrix}$$

REFERENCIAS

[1] F. P. Kelly, S. Zachary, and I. B. Ziedins, eds., *Notes on Effective Bandwidth*, pp. 141–168. Oxford University Press, 1996.

[2] C. A. Courcoubetis and A. Dimakis, "The price of queueing," *ArXiv*, vol. abs/2007.08318, 2020.

[3] M. Zehri, A. Haastrup, D. Rincon, J. R. Piney, S. Sallent, and A. Bazzi, "A qos-aware dynamic bandwidth allocation algorithm for passive optical networks with non-zero laser tuning time," *Photonics*, 03 2021.

[4] J. Zhao, J. Du, Y. Yue, and J. Liu, "Special issue on advanced technique and future perspective for next generation optical fiber communications," *Photonics*, vol. 9, p. 280, 04 2022.

[5] A. Parra León, E. M. Piedrahita, and O. Salcedo, "Aplicaciones del modelo on/of al tráfico agregado en las redes de comunicaciones," *Tecnura*, vol. 15, p. 129–147, jun. 2011.

[6] A. Seal, A. Karlekar, O. Krejcar, and E. Herrera-Viedma, "Performance and convergence analysis of modified c-means using jeffreys-divergence for clustering," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. In Press, p. 1, 12 2021.

[7] J. Bavio, C. Fernández, and B. Marrón, "Comparison of effective bandwidth estimation methods for data networks," *Global Journal of Computer Science and Technology: ENetwork, Web & Security*, vol. 22, no. 2, pp. 12–20, 2022.

[8] M. Meila and D. Heckerman, "An experimental comparison of model-based clustering methods," *Machine Learning*, vol. 42, pp. 9–29, 2001.

[9] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, p. 2837–2854, dec 2010.

[10] J. Bavio and B. Marrón, "Properties of the estimators for the effective bandwidth in a generalized markov fluid model," *Open Journal of Statistics*, vol. 8, no. 1, pp. 69–84, 2018.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[12] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832 – 837, 1956.

[13] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*. Wiley, New York, 1973.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1—22, 1977.

[15] E. H. Koroishi, F. A. L. Molina, A. W. Faria, and V. Steffen Junior, "Clustering by passing messages between data points," *Journal of Aerospace Technology and Management*, vol. 315, pp. 972–6, 2007.

[16] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Addison Wesley, 2005.

[17] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction To Cluster Analysis*. Wiley, New York, 1990.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[19] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

Carina N. Fernández holds a degree in Mathematics and is studying for a PhD in Mathematics at the Universidad Nacional del Sur. She is currently developing her teaching work in the area of Probability and Statistics in the Math Department of Universidad Nacional del Sur. Her research has focused on the study of the modeling of data networks using stochastic processes, the current topic of her doctoral thesis. She is a member of accredited research projects.

José M. Bavio holds a degree in Mathematics, and the Ph.D. in Mathematics from the Universidad Nacional del Sur. He is currently developing his teaching work in the Math Department of Universidad Nacional del Sur and at Universidad Salesiana, in Bahía Blanca. His research has been concerned with simulation and estimation of stochastic processes. He had also work in collaboration in different fields making statistic analysis using python tools.

Beatriz S. Marrón holds a degree in Mathematics and the Ph.D. in Mathematics from the Universidad Nacional del Sur, Bahía Blanca, Argentina. She is a full time professor at the Department of Mathematics at the Universidad Nacional del Sur, Argentina. She is currently developing her teaching and research work in the area of Probability and Statistics and is director of accredited research projects.