



Estimación del tiempo de respuesta de *proxies* web en redes comunitarias utilizando algoritmos de factorización matricial

Diego Bores Quijano*, Miguel L. Bote Lorenzo*, Eduardo Gómez Sánchez*, Roc Meseguer Pallarés**

*Departamento de Teoría de la Señal, Comunicaciones e Ingeniería Telemática, Universidad de Valladolid
ETSI de Telecomunicación, Paseo de Belén 15, 47011 Valladolid

**Departamento de Arquitectura de Computadores, Universitat Politècnica de Catalunya
Campus Nord - Edificios D6/C6, C. Jordi Girona 1-3, 08034 Barcelona
diego@gsic.uva.es, migbot@tel.uva.es, edugom@tel.uva.es, meseguer@ac.upc.edu

Resumen—En este artículo se evalúa la precisión de 3 algoritmos de factorización matricial para estimar el tiempo de respuesta de los *proxies* web en redes comunitarias a partir medidas de dicho indicador de calidad de servicio que son tomadas por los clientes y compartidas entre ellos. Para ello se ha llevado a cabo una serie de experimentos utilizando los datos obtenidos de la emulación de una red comunitaria formada por 8 clientes y 5 *proxies*. Los resultados muestran que 2 de los algoritmos consiguen hacer las estimaciones con una elevada precisión, siendo el mejor el algoritmo de factorización matricial adaptativa, incluso cuando el número de *proxies* sondeados por cada cliente para obtener las medidas de tiempo de respuesta es bajo.

Palabras Clave—redes comunitarias, *proxy* web, calidad de servicio, algoritmos de factorización matricial

I. INTRODUCCIÓN

Las redes comunitarias [1] son redes distribuidas, descentralizadas y a gran escala. Estas redes permiten que las comunidades locales creen una infraestructura de red propia, y ofrecen a sus usuarios conexión con Internet de forma asequible, incluso en áreas rurales. En Europa hay varias redes comunitarias de gran tamaño, como es el caso de la Linux [2] y Funkfeuer [3], desplegadas en Italia y Austria respectivamente. En España, la red comunitaria Guifi.net [4] cuenta con más de 30.000 nodos situados principalmente en Cataluña.

El acceso a la Web en las redes comunitarias se hace típicamente a través de *proxies*, siendo los usuarios los que eligen su *proxy* preferido de entre todos los disponibles en la red [5]. Esto implica que, para acceder a un servidor web externo, suele ser necesario recorrer una ruta dentro de la red comunitaria que atraviesa varios enlaces y nodos hasta llegar al *proxy* seleccionado. Dicho *proxy* será el encargado de realizar la petición al servidor externo en nombre del usuario.

Del hecho de que sean los usuarios quienes eligen el *proxy* que desean utilizar surge la posibilidad de congestión en estos nodos de salida [5]. Por lo tanto, podría ocurrir que mientras unos *proxies* estuviesen subocupados, otros acabarían saturados por la demanda, dando lugar a una degradación del rendimiento del acceso a la Web. Así, la elección del *proxy* por parte de los usuarios puede afectar a la calidad de servicio que éstos perciben.

Los mecanismos de selección automática de *proxies* web en redes comunitarias que es posible encontrar en la literatura como [5], [6] requieren que, cada cierto tiempo, los clientes lleven a cabo sondeos de un número determinado de *proxies*. Esto permite a los clientes obtener medidas indicativas de la calidad de servicio que cabe esperar de cada uno de los *proxies*. Algunos ejemplos de estas medidas son la latencia y el tiempo de respuesta [6].

Cuanto mayor es el número de *proxies* web de los cuales se dispone de medidas indicativas de calidad de servicio, mayores son las posibilidades del cliente de seleccionar uno adecuado. Sin embargo, el sondeo de un elevado número de *proxies* por parte de los clientes puede suponer una elevada carga tanto para unos como para otros, además de un incremento notable de tráfico en la red comunitaria [6]. Para evitar este problema, cabe la posibilidad de que cada cliente sondee un número reducido de *proxies* y comparta con otros clientes las medidas indicativas de calidad de servicio que obtenga. Los clientes que reciben dichas medidas pueden utilizarlas para estimar las medidas que recabarían si ellos mismos sondearan los *proxies* correspondientes. Con las medidas obtenidas de sus propios sondeos y las estimaciones realizadas a partir de los sondeos hechos por otros clientes, cada cliente tiene la posibilidad de elegir entre un número mayor de *proxies* sin necesidad de realizar sondeos adicionales.

Para que esta aproximación para la selección de *proxies*

web pueda ser efectiva, es imprescindible que las estimaciones que hagan los clientes sean precisas. En trabajos de la literatura como [7], [8], [9] se ha comprobado que los algoritmos de factorización matricial pueden obtener estimaciones precisas de medidas de calidad de servicio en problemas semejantes al planteado en este artículo. Por ejemplo, algunos de estos algoritmos se estudiaron en [9] para la estimación tanto del tiempo de respuesta como de la tasa de transferencia ofrecidos por servicios web a partir de las medidas obtenidas mediante sondeos de los clientes y que después eran compartidas entre ellos. Hasta donde saben los autores, estos algoritmos no han sido estudiados para la estimación de medidas de calidad de servicio en el contexto de las redes comunitarias.

El objetivo de este artículo es estudiar la posibilidad de utilizar algoritmos de factorización matricial para estimar el tiempo de respuesta de *proxies* web en redes comunitarias. Para ello se evalúa la precisión de 3 algoritmos de factorización matricial bien conocidos cuando estos se utilizan para llevar a cabo la estimación de los tiempos de respuesta de un conjunto de 5 *proxies* web a partir de los sondeos realizados cada 10 segundos durante 2 días por 8 clientes.

La estructura del resto del artículo es la siguiente. En la sección II se discute el trabajo relacionado que es posible encontrar en la literatura. A continuación, la sección III presenta el marco experimental que ha sido utilizado en nuestro estudio con el objetivo de facilitar su reproducibilidad. La sección IV presenta y analiza los resultados de los experimentos. Finalmente, en la sección V se recogen las principales conclusiones que es posible obtener de este estudio y se avanzan las principales líneas de trabajo futuro.

II. TRABAJO RELACIONADO

La monitorización siempre ha sido un elemento clave para asegurar el rendimiento de un sistema distribuido complejo, como una red comunitaria. Es un primer paso para controlar la calidad del servicio, detectar anomalías o tomar decisiones sobre la asignación de recursos. En el caso de los *proxies* web de una red comunitaria esta monitorización es especialmente útil para la selección de un determinado *proxy* por parte de un cliente. Los clientes pueden seleccionar el *proxy* adecuado de acuerdo con alguna métrica de calidad de servicio. Además, esto está relacionado con el problema que supone que un gran número de clientes puedan navegar por la Web aprovechando la capacidad agregada de un número reducido de *proxies*.

La visibilidad completa entre clientes y *proxies* es una primera aproximación a esta monitorización de soporte para la selección. Conceptualmente es una matriz de todos los *proxies* medidos por todos los clientes con una métrica de calidad de servicio dada, como puede ser el tiempo de respuesta. Un ejemplo típico es la basada en la infraestructura de una red de distribución de contenidos como la de Google [10]. Aunque estas medidas no son sencillas de obtener ni precisas sin implicar a *proxies*, infraestructura y especialmente a los clientes [11]. Esta

visibilidad completa también se puede conseguir de forma activa por parte de los clientes. Una primera aproximación es la fuerza bruta, donde el cliente en solitario sondea activamente todos los *proxies* [12]. Aunque esta solución puede ser útil para una red pequeña no es escalable a una red comunitaria y tiene un gran coste en sondeos a los *proxies*.

Para conseguir escalabilidad y reducir el coste de la monitorización, una aproximación típica es reducir la cantidad de *proxies*: probando solo los más cercanos [6] o seleccionando algunos aleatoriamente [13]. Aunque con estas aproximaciones tenemos una visión muy reducida, perdiendo la visibilidad completa.

La visibilidad completa también se puede conseguir cooperando entre los diferentes clientes y compartiendo las visiones parciales de cada uno. Una estrategia típica son las coordenadas virtuales. Un ejemplo típico es Vivaldi [14], que es un sistema que permite la estimación de latencias entre todos los nodos que participan en él. En [15] se presenta un sistema basado en Vivaldi que permite estimar latencias entre un nodo del sistema, por ejemplo un cliente, y otro externo, por ejemplo un *proxy*. Esta aproximación a la visibilidad completa tiene dos problemas. El primero, las medidas que Vivaldi necesita son algo costosas por las continuas pruebas activas por parte de los clientes y no son precisas [11], [16]. El segundo problema es la métrica usada; Vivaldi usa la latencia de red entre nodos. En el caso concreto de la selección de *proxy* la latencia no refleja algunos aspectos importantes, como la carga del propio *proxy*, y la carga de la conexión a Internet del *proxy* [6]. En [5] se propone un sistema basado en Vivaldi donde los clientes comparten tanto los nuevos valores de latencia entre los clientes, como la latencia de los *proxies* probados.

Hay otras estrategias de cooperación no basadas en coordenadas virtuales como la basada en reutilizar los paquetes de sondeo. En [17], los clientes intermedios capturan los sondeos de otro cliente a un *proxy* para calcular su propia información sin hacer un sondeo por su cuenta. Esta aproximación presenta un importante problema: solo se pueden reutilizar los sondeos de otros clientes que pasan por el propio cliente. En las redes comunitarias eso no es lo habitual; los clientes están en los extremos de la topología y no actúan como encaminadores, por lo que no pasan peticiones por ellos [4].

Otra estrategia de cooperación se basa en el sondeo aleatorio de algunos *proxies*. En [18], los clientes sondean el último *proxy* usado y aleatoriamente dos más. Los clientes comparten estos valores solo con los clientes más cercanos topológicamente. El problema de esta estrategia es que no garantiza la obtención de los valores de todas las celdas de la matriz de visibilidad correspondientes a las combinaciones de clientes y *proxies*. Este caso usa la idea de la semejanza entre los valores de sondeo del mismo *proxy* de clientes topológicamente cercanos para rellenar las celdas vacías. Concretamente, se rellenan directamente con los valores medidos por los clientes más cercanos.

Para evitar estas celdas vacías en la matriz de visibilidad se pueden usar técnicas de estimación de estos valores.

En [19] se usan técnicas de aprendizaje profundo (*deep learning*) para realizar obtenerlas. El problema de esta aproximación es el coste computacional de los algoritmos de aprendizaje profundo, por lo que tienen que ejecutarse en equipos dedicados con el problema añadido de la recolección y transferencia a este equipo de la información necesaria para la predicción.

El filtrado colaborativo es una aproximación para la estimación habitualmente mucho menos costosa computacionalmente, lo cual comporta la ventaja de poder ejecutarse en los propios clientes y encaminadores de una red comunitaria. En [20] se propone el uso de un algoritmo de filtrado colaborativo para la recomendación o la selección de servicios. Se basa en el coeficiente de correlación de Spearman entre los vectores de los valores de los sondeos de los clientes. Adicionalmente se reduce el conjunto de clientes usando solo el subconjunto de clientes más cercanos en distancia. Por otra parte, en [21] se trata de obtener los valores desconocidos de la matriz mediante un mecanismo de filtrado colaborativo basado el algoritmo de *Kernel Least Mean Square (KLMS)*. El problema de esta propuesta es también el elevado coste computación. Otro ejemplo es el sistema de recomendación basado en filtrado colaborativo propuesto en [22]. En este caso los valores de la matriz no son solo medias de calidad de servicio, también utiliza las preferencias de los usuarios, como por ejemplo el precio.

La factorización matricial es también una técnica de filtrado colaborativo que ha sido aplicada con éxito para la estimación de los valores desconocidos de la matriz de visibilidad. En [7] se propone un algoritmo de factorización extendida matricial, en el que usan diferentes técnicas de cálculo de similitud tanto entre clientes como entre servicios para obtener un conjunto de clientes y servicios cercanos. La estimación se basa en la idea de una percepción similar de servicios cercanos por parte de clientes cercanos. En [8] se hace una propuesta similar pero en este caso fusionan información global con información cercana. El valor a estimar depende de el conjunto de una información del usuario, la información de los vecinos del usuario, y alguna información global de la matriz. En [9] se propone un algoritmo de factorización matricial adaptativa y en línea, que además tiene en cuenta la evolución temporal de la métrica de rendimiento del servicio. Esta propuesta aporta dos contribuciones interesantes: en línea, con lo que puede ejecutarse en tiempo real en los clientes, y la evolución temporal, con lo que estima siguiendo la tendencia temporal. Hasta donde saben los autores, en la literatura no se ha estudiado la estimación de los valores de la matriz de visibilidad con algoritmos de factorización matricial para llevar a cabo la monitorización y selección de *proxies* en el contexto de una red comunitaria con las restricciones de recursos de este tipo de redes.

III. MARCO EXPERIMENTAL

A. Conjunto de datos

Para la obtención de datos con los que realizar los experimentos, se emuló el funcionamiento de una red

Tabla I
DISTRIBUCIÓN DE CLIENTES Y PROXIES EN DISTINTOS NODOS DE LA PLATAFORMA PLANETLAB.

	Nodo
Cliente 1	cse-yellow.cse.chalmers.se
Cliente 2	mars.planetlab.haw-hamburg.de
Cliente 3	pl2.uni-rostock.de
Cliente 4	planet3.cs.huji.ac.il
Cliente 5	planetlab3.di.unito.it
Cliente 6	planetlab-1.ing.unimo.it
Cliente 7	planetlab-2.cs.ucy.ac.cy
Cliente 8	ple43.planet-lab.eu
Proxy 1	ple3.planet-lab.eu
Proxy 2	planet4.cs.huji.ac.il
Proxy 3	cse-white.cse.chalmers.se
Proxy 4	planetlab1.informatik.uni-kl.de
Proxy 5	planetlab1.cs.aueb.gr

comunitaria en la plataforma PlanetLab [23]. Esto se hizo desplegando 8 clientes y 5 *proxies* [24] web en diferentes nodos de PlanetLab distribuidos por Europa y Oriente Próximo tal y como se muestra en la tabla I. Si bien es cierto que algunas redes comunitarias pueden llegar a tener un número de clientes y *proxies* mucho mayor, este conjunto de datos puede considerarse suficiente para estudiar el potencial de los algoritmos de factorización para la estimación del tiempo de respuesta.

Cada cliente sondeó a todos los *proxies* cada 10 segundos durante 2 días. En todos los sondeos se solicitó al *proxy* el mismo fichero¹ y, siempre que se consiguió hacer la descarga, se registró el tiempo de respuesta, entendido como el tiempo que transcurre desde el cliente realiza la petición hasta que éste recibe el último *byte* de la respuesta. De acuerdo con [6], esta medida tiene una correlación alta con el rendimiento real de los *proxies*.

La tabla II recoge el número de medidas del tiempo de respuesta obtenidas por cada cliente para cada *proxy*. En todos los casos se observa que la cantidad de medidas registradas es inferior a los 17.280 sondeos que se realizaron por cada pareja de cliente y *proxy*. En algunos casos las medidas no pudieron realizarse debido a problemas de comunicación entre el cliente y el *proxy*, lo cual puede deducirse de la elevada diferencia en el número de medidas recogidas del *proxy* 5 por los clientes 4 a 6 respecto al resto de clientes. En otros casos el *proxy* no pudo acceder al recurso solicitado por el cliente, con lo que tampoco fue posible medir el tiempo de respuesta. Estos problemas de comunicación tanto entre cliente y *proxy* como entre *proxy* y servidor del recurso solicitado se dan también en las redes comunitarias.

En la figura 1 se muestra la distribución de las medidas recogidas para cada pareja de cliente y *proxy*. En ella se puede ver que las distribuciones de las medidas no son normales en la mayoría de los casos. También se observa que algunas distribuciones presentan una variabilidad significativa. Ambos hechos suponen *a priori* una mayor dificultad para la estimación de los tiempos de respuesta.

¹<http://ovh.net/files/1Mb.dat>

Tabla II
NÚMERO DE MEDIDAS DEL TIEMPO DE RESPUESTA RECOGIDAS POR CADA CLIENTE Y PROXY.

	Proxy 1	Proxy 2	Proxy 3	Proxy 4	Proxy 5	Total
Cliente 1	15.476	15.474	15.474	15.476	11.664	73.564
Cliente 2	15.486	15.484	15.485	15.486	11.666	73.607
Cliente 3	15.474	15.474	15.473	15.474	11.665	73.560
Cliente 4	14.697	14.697	14.697	14.697	14.697	73.485
Cliente 5	14.742	14.742	14.742	14.742	14.742	73.710
Cliente 6	14.678	14.678	14.678	14.678	14.678	73.390
Cliente 7	15.472	15.470	15.470	15.471	11.658	73.541
Cliente 8	15.502	15.500	15.500	15.501	11.667	73.670
Total	121.519	121.519	121.525	121.525	102.437	588.527

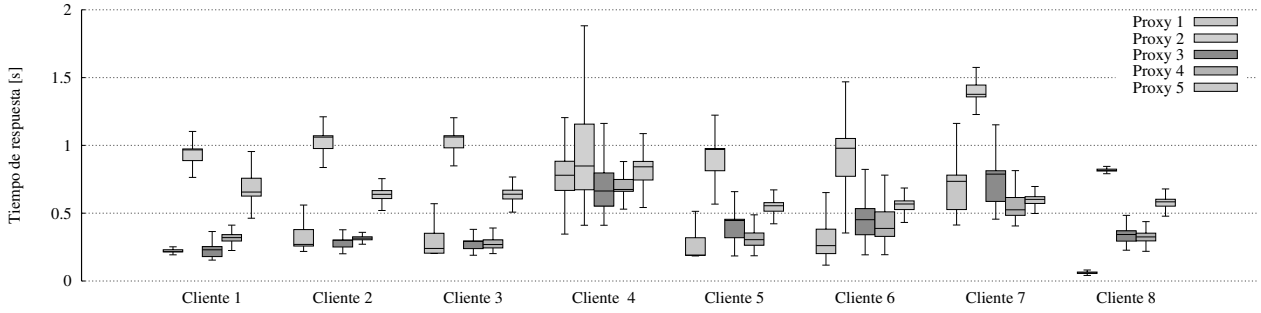


Figura 1. Diagramas de cajas del tiempo de respuesta medido por cada cliente para cada proxy.

B. Algoritmos de factorización matricial para la estimación del tiempo de respuesta

Considérese un conjunto de c clientes y p proxies web en una red comunitaria. Si, de manera periódica, cada cliente sondea un número de proxies inferior a p y compartiera con el resto de clientes las medidas indicativas de calidad de servicio obtenidas, cada cliente podrá crear una matriz de medidas $M \in \mathbb{R}^{c \times p}$. Cada elemento de esta matriz $m_{i,j}$ se corresponde con la medida de indicativa de calidad de servicio entre el cliente c_i y el proxy p_j . Dicho elemento tomará un valor conocido en caso de que el cliente c_i haya sondeado p_j exitosamente y un valor desconocido en caso contrario. Se asume que los valores de la matriz presentarán un alto nivel de correlación dado que se espera que dos clientes diferentes que tengan condiciones de acceso de red similares a un mismo proxy obtengan medidas similares al sondearlo.

Los valores desconocidos pueden ser estimados mediante la factorización de la matriz M aprovechando dicha correlación. Así, por ejemplo, el algoritmo básico de factorización matricial [25] intenta encontrar las matrices $C \in \mathbb{R}^{d \times c}$ y $P \in \mathbb{R}^{d \times p}$ de forma que la matriz $\hat{M} = C^T P$ de rango d minimice la distancia euclídea a la matriz objetivo M calculada teniendo en cuenta sólo los elementos con valores conocidos. De este modo, a cada elemento $m_{i,j}$ con valor desconocido le corresponde la estimación $\hat{m}_{i,j}$.

En este artículo se estudia la posibilidad de estimar el tiempo de respuesta de los proxies web de una red comunitaria utilizando 3 algoritmos de factorización matricial bien conocidos. Dichos algoritmos son los siguientes:

- AMF (*Adaptive Matrix Factorization*, factorización matricial adaptativa) [9], que realiza una factoriza-

ción matricial capaz de tener en cuenta la evolución temporal de los valores en la matriz. Este algoritmo emplea técnicas de transformación de datos, aprendizaje en línea y pesos adaptativos.

- NTF (*Non-negative Tensor Factorization*, factorización tensorial no negativa) [26] también tiene en cuenta la evolución temporal de los valores en la matriz. Sin embargo, en lugar de construir un modelo que evolucione en el tiempo, se genera un modelo a partir de un tensor formado por las matrices correspondientes a diferentes instantes.
- PMF (*Probabilistic Matrix Factorization*, factorización matricial probabilística) [27], que no tiene en cuenta la evolución temporal de los valores de la matriz en la matriz.

La implementación de AMF empleada en los experimentos de este artículo se encuentra en [28], mientras que las de NTF y PMF están disponibles en [29].

C. Algoritmo de base y métrica de rendimiento

La precisión de los algoritmos de factorización matricial estudiados en este artículo se compara con la de un algoritmo de base, estableciéndose así un rendimiento mínimo que se espera que sea superado por los primeros. Este algoritmo de base toma el valor medio de todos los elementos con valor conocido de M como estimación para cualquiera de los elementos con valor desconocido de esa misma matriz.

Para evaluar la precisión de los distintos algoritmos se utiliza el error absoluto medio (MAE, *Mean Average Error*) de las estimaciones. Esta métrica es ampliamente utilizada en la literatura (ej. [7], [8]) para medir el rendimiento de algoritmos de estimación en problemas similares al abordado en este artículo.

D. Experimentos

Los experimentos descritos en este artículo estudian la precisión de los distintos algoritmos suponiendo que no varía el número de *proxies* que cada cliente sondea cada 10 segundos. Sin embargo, los *proxies* que en concreto son sondeados por un cliente dado pueden variar de una ronda de sondeos a otra, puesto que son elegidos aleatoriamente en cada una de ellas. Específicamente, se han llevado a cabo experimentos considerando 2, 3 y 4 *proxies* sondeados por cada cliente. Esto hace que las matrices de medidas que se pueden construir tras cada ronda de sondeos tengan una densidad de valores conocidos del 40 %, 60 % y 80 %, respectivamente. Las medidas que no son incluidas en la matriz de valores conocidos son después utilizadas para evaluar la precisión de las estimaciones.

Para cada densidad de matriz considerada en el artículo se han generado 5 conjuntos diferentes de entrenamiento y test a partir del conjunto de datos inicial. La diferencia entre conjuntos reside en que, para cada ronda de sondeos, los *proxies* que en concreto son sondeados por un cliente dado pueden variar porque son elegidos aleatoriamente.

De este modo, la precisión de todos los algoritmos ha sido evaluada para cada densidad de matriz mediante 5 experimentos de estimación con los conjuntos de datos correspondientes. Para evaluar dicha precisión se han obtenido tanto el valor medio del MAE de las estimaciones hechas en cada ronda como la desviación estándar del MAE entre rondas.

Cada algoritmo utilizado tiene, además, una serie de parámetros que determinan su comportamiento. A continuación se indican los valores que se utilizaron en los experimentos tras llevar a cabo una exploración no sistemática del espacio de posibles valores de los parámetros. Para el caso de AMF, los parámetros son la dimensionalidad de los factores latentes (en nuestro caso 10), la tasa de aprendizaje (0,8), el parámetro de regularización (0,0003), el número máximo de iteraciones (50), el umbral de convergencia (6e-3) y el peso de la media exponencial móvil (0,3). Por su parte, para el caso de NTF los parámetros son, de nuevo, la dimensionalidad de los factores latentes (10), el parámetro de regularización (40), y el número máximo de iteraciones (300). Por último, para PMF los parámetros son la dimensionalidad de los factores latentes (10), el parámetro de regularización (0), el valor inicial de la tasa de aprendizaje (0,01) y el número máximo de iteraciones (600).

IV. RESULTADOS

En la tabla III se muestran los resultados obtenidos al ejecutar los métodos presentados en las subsecciones III.B y III.C sobre el conjunto de datos descrito en la subsección III.A. Para cada método y densidad se muestra el valor medio y la desviación del MAE entre rondas. Como se puede observar, AMF obtiene un rendimiento muy elevado, con errores bajos y poco variables, lo que supone una prueba de que los algoritmos de factorización matricial tienen un gran potencial para estimar desde los clientes el tiempo de respuesta de distintos *proxies*, habiendo sondeado sólo a

algunos de ellos y empleando las medidas compartidas por sus vecinos. Además, en la tabla IV se puede apreciar la adecuación al problema de AMF, en gran medida gracias a que incorpora mecanismos de pre- y post-transformación de los datos que son muy adecuados cuando éstos siguen una distribución de probabilidad muy asimétrica, como es el caso. El resultado es considerablemente mejor que el del algoritmo de base, lo que justifica el coste computacional de su utilización (obsérvese que el coste de compartición entre los clientes de los tiempos de respuesta medidos se aplica a los cuatro algoritmos). Finalmente, cabe observar cómo el aumento de la densidad ayuda a mejorar el rendimiento en todos los métodos, aunque en una proporción pequeña, lo que probablemente es debido al escaso número de clientes y *proxies*. En la figura 2 se muestran las estimaciones del tiempo de respuesta que un cliente hace de un *proxy* durante un pequeño intervalo tomado como ejemplo, corroborando que AMF hace estimaciones muy precisas y detecta bien las variaciones más marcadas del tiempo de respuesta.

Para comprender mejor los resultados, las tablas IV y V muestran el desglose del rendimiento de los algoritmos AMF y base, respectivamente, para cada pareja de cliente y proxy, con densidad 40 % (en cada ronda cada cliente sondea a 2 de los 5 *proxies*), aunque las observaciones que se realizan a continuación se repiten para las otras densidades evaluadas. En primer lugar, en la tabla IV puede verse cómo los clientes 1, 2, 3 y 8 obtienen tiempos de respuesta más estables para sus sondeos. Además, estos tiempos son bajos para los *proxies* 1, 3 y 4. Observando el comportamiento de ambos métodos, el AMF obtiene unos resultados de gran precisión para estas parejas cliente y *proxy*. Hay que observar que los otros clientes tienen medidas más altas y con mayor variabilidad, pero AMF aprende para cada cliente cuáles son los clientes más parecidos. Por el contrario, el algoritmo de base trata la información de cualquier otro cliente por igual, cometiendo un alto error para estas parejas (de hecho, en ellas se da el mayor ratio de mejora de AMF sobre el algoritmo de base). Esto refuerza la idea de que este algoritmo de factorización matricial puede ser una buena solución cuando se disponga de muchos clientes, cada uno de los cuales sondee a un conjunto pequeño de *proxies*.

Un detalle particular que se aprecia en la tabla IV es que el cliente 8 obtiene tiempos de respuesta muy bajos del *proxy*. Tanto es así que el rango de estas medidas no se solapa con el de los tiempos de cualquier otra pareja cliente y *proxy*. Esto hace que el resultado sea peor para ambos métodos, ya que en este cliente y para este *proxy* se estima un valor próximo a los medidos en otros clientes. No obstante, aunque afecta a AMF, lo hace en mucha mayor medida con el algoritmo de base. Algo semejante sucede con las medidas que el cliente 7 toma del *proxy* 2, en las que se obtienen valores muy elevados, y con muy poco solape con los medidos en otros clientes. Esto hace que esta pareja de cliente y *proxy* tenga el peor MAE medio, siendo de nuevo una situación mucho más desfavorable para el algoritmo de base.

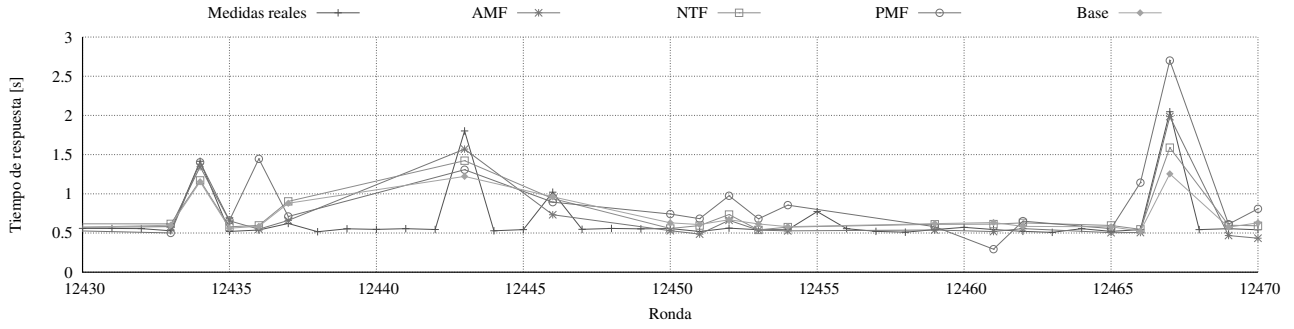


Figura 2. Extracto de la estimación del tiempo de respuesta que el cliente 6 hace del *proxy* 5, utilizando los algoritmos de factorización matricial y de base, frente al tiempo real, con densidad 60 %.

Tabla III

MEDIA Y DESVIACIÓN ESTÁNDAR DEL MAE OBTENIDO A LO LARGO DE LAS RONDAS DEL EXPERIMENTO, PARA AMF, NTF, PMF Y EL ALGORITMO DE BASE, CON DISTINTAS DENSIDADES DEL CONJUNTO DE DATOS

Densidad	AMF	NTF	PMF	Algoritmo de base
40 %	$0,104 \pm 3,8e-4$	$0,140 \pm 6,18e-3$	$0,570 \pm 9,70e-4$	$0,315 \pm 1,2e-4$
60 %	$0,100 \pm 3,6e-4$	$0,123 \pm 1,62e-3$	$0,438 \pm 1,05e-3$	$0,313 \pm 4,5e-4$
80 %	$0,096 \pm 5,6e-4$	$0,116 \pm 3,36e-3$	$0,359 \pm 1,51e-3$	$0,313 \pm 3,2e-4$

Tabla IV

MEDIA DEL MAE OBTENIDO POR AMF PARA CADA PAREJA DE CLIENTE Y PROXY CON DENSIDAD 40 %

	Cliente 1	Cliente 2	Cliente 3	Cliente 4	Cliente 5	Cliente 6	Cliente 7	Cliente 8	Media
<i>Proxy 1</i>	0,050	0,055	0,051	0,170	0,109	0,119	0,108	0,203	$0,108 \pm 0,00069$
<i>Proxy 2</i>	0,136	0,141	0,139	0,138	0,183	0,194	0,209	0,137	$0,159 \pm 0,00175$
<i>Proxy 3</i>	0,043	0,044	0,041	0,163	0,107	0,116	0,100	0,051	$0,082 \pm 0,00050$
<i>Proxy 4</i>	0,057	0,050	0,057	0,158	0,113	0,125	0,087	0,056	$0,087 \pm 0,00052$
<i>Proxy 5</i>	0,059	0,053	0,056	0,151	0,093	0,091	0,073	0,053	$0,080 \pm 0,00049$
Media	0,069	0,069	0,069	0,156	0,122	0,130	0,118	0,102	$0,104 \pm 0,00038$

Tabla V

MEDIA DEL MAE OBTENIDO POR EL ALGORITMO DE BASE PARA CADA PAREJA DE CLIENTE Y PROXY CON DENSIDAD 40 %

	Cliente 1	Cliente 2	Cliente 3	Cliente 4	Cliente 5	Cliente 6	Cliente 7	Cliente 8	Media
<i>Proxy 1</i>	0,383	0,306	0,343	0,257	0,367	0,343	0,180	0,536	$0,340 \pm 0,00076$
<i>Proxy 2</i>	0,454	0,528	0,532	0,366	0,457	0,447	0,927	0,322	$0,506 \pm 0,00209$
<i>Proxy 3</i>	0,388	0,329	0,338	0,180	0,219	0,208	0,226	0,277	$0,273 \pm 0,00082$
<i>Proxy 4</i>	0,291	0,297	0,342	0,205	0,313	0,269	0,152	0,286	$0,270 \pm 0,00071$
<i>Proxy 5</i>	0,153	0,122	0,124	0,321	0,164	0,129	0,112	0,109	$0,157 \pm 0,00079$
Media	0,343	0,327	0,347	0,267	0,307	0,282	0,330	0,316	$0,315 \pm 0,00012$

Por último, la última columna de las tablas IV y V muestra los agregados por *proxy*, incluyendo la desviación estándar. Cabe apreciar que la desviación del error es muy pequeña en todos los casos, aunque es sensiblemente mayor en el *proxy* 2, tanto para AMF como para el algoritmo de base. De hecho, individualmente para cada cliente también la desviación es mayor para este *proxy* (esto no se muestra en la tabla por problemas de espacio). Finalmente, los valores medios del MAE son, con ambos métodos, mayores con el *proxy* 2. La explicación de esto está en la enorme variabilidad de las medidas de este *proxy* tomadas desde los clientes 4, 5 y 6. Aquí se puede apreciar que el hecho de tener clientes muy ruidosos, que obtienen medidas en rangos que se solapan con los de otros clientes (y, que por lo tanto hacen que a veces sean parecidos) distorsiona las estimaciones que hacen esos otros clientes.

V. CONCLUSIONES Y TRABAJO FUTURO

El acceso a la Web en redes comunitarias se hace habitualmente a través de *proxies*. Por este motivo, es importante que los clientes seleccionen *proxies* web que puedan ofrecer una calidad de servicio adecuada a los usuarios. Esta selección puede ser llevada a cabo por cada cliente basándose tanto en las medidas indicativas de calidad de servicio recogidas por el propio cliente mediante el sondeo de algunos *proxies* como en las estimaciones de las medidas que el mismo cliente podría obtener de otros *proxies* si los sondeara. Estas estimaciones pueden hacerse empleando algoritmos de factorización matricial a partir de las medidas tomadas por un cliente dado junto con las medidas realizadas por otros clientes, los cuales las comparten con el primero.

En este artículo se ha evaluado la precisión de 3 algoritmos de factorización matricial para la estimación del tiempo de respuesta de los *proxies* web en redes comunitarias.

rias. Para ello se ha realizado una serie de experimentos de estimación empleando los datos obtenidos de la emulación de una red comunitaria formada por 8 clientes y 5 *proxies* web en la que los primeros sondeaban cada 10 segundos a los segundos. Los resultados muestran el potencial de los algoritmos de factorización matricial, y en concreto de AMF, para la estimación del tiempo de respuesta a partir unas pocas medidas tomadas desde los clientes y compartidas entre ellos.

No obstante, para generalizar estos resultados, en el trabajo futuro está previsto generar un conjunto de datos en el que tanto el número de clientes como el de *proxies* sea mayor, lo que permitiría comprender mejor el efecto de la densidad de medidas y validar las observaciones realizadas a partir de los experimentos de este artículo. Además, se pretende que este nuevo conjunto de datos sea obtenido de clientes y *proxies* web desplegados en una red comunitaria real.

En relación a los algoritmos empleados, se llevará a cabo una exploración sistemática del espacio de valores que pueden tomar los parámetros de entrenamiento. También se comprobará si la técnica de transformación de datos empleada por AMF puede ser usada también para mejorar los resultados de NTF y PMF.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Fondo Europeo de Desarrollo Regional y la Agencia Estatal de Investigación del Ministerio de Ciencia, Innovación y Universidades a través de los proyectos de investigación TIN2017-85179-C3-2-R y TIN2016-77836-C2-2-R.

REFERENCIAS

- [1] Bart Braem, Chris Blondia, Christoph Barz, Henning Rooze, Felix Freitag, Leandro Navarro, Joseph Bonicioli, Stavros Papathanasiou, Pau Escrib, Roger Baig Viñas, Aaron Kaplan, Axel Neumann, Ivan Vilata i Balaguer, Blain Tatum, Malcom Matson, "A case for research with and on community networks", ACM SIGCOMM Computer Communication Review, n. 3, pp. 68-73, 2013. doi: 10.1145/2500098.2500108
- [2] Leonardo Maccari, "An analysis of the Ninux wireless community network", Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 1-7, 2013. doi: 10.1109/WiMob.2013.6673332
- [3] Funkfeuer, "Initiative für freie Netze", <https://www.funkfeuer.at/>, Última visita: 9 de mayo de 2019.
- [4] Davide Vega, Roger Baig, Llorenç Cerdà-Alabern, Esunly Medina, Roc Meseguer, Leandro Navarro, "A technological overview of the guifi.net community network", Computer Networks, vol. 93, pp.260-278, 2015. doi: 10.1016/j.comnet.2015.09.023
- [5] Emmanouil Dimogerontakis, João Neto, Roc Meseguer, Leandro Navarro, Luís Vega, "Client-side routing-agnostic gateway selection for heterogeneous wireless mesh networks", Symposium on Integrated Network and Service Management (IM), pp. 377-385, 2017. doi: 10.23919/INM.2017.7987301
- [6] Khulan Batbayar, Emmanouil Dimogerontakis, Roc Meseguer, Leandro Navarro, Esunly Medina, Rodrigo M. Santos, "The RIMO Gateway Selection Approach for Mesh Networks: Towards a Global Internet Access for All", Conference on Ubiquitous Computing and Intelligence (UCAI), vol. 2, n. 1258, 2018.
- [7] Wei Lo, Jianwei Yin, Shuiguang Deng, Ying Li, Zhaohui Wu, "An extended matrix factorization approach for QoS prediction in service selection", Conference on Services Computing (SCC), pp. 162-169, 2012. doi: 10.1109/SCC.2012.36
- [8] Zibin Zheng, Hao Ma, Michael R. Lyu, Irwin King, "Collaborative web service QoS prediction via neighborhood integrated matrix factorization", IEEE Transactions on Services Computing, vol. 6, n. 3, pp. 289-299, 2013. doi: 10.1109/TSC.2011.59
- [9] Jieming Zhu, Pinjia He, Zibin Zheng, Michael R. Lyu, "Online QoS prediction for runtime service adaptation via adaptive matrix factorization", IEEE Transactions on Parallel and Distributed Systems, vol. 28, n. 10, pp. 2911-2924, 2017. doi: 10.1109/TPDS.2017.2700796
- [10] Rupa Krishnan, Harsha V. Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, Jie Gao "Moving beyond end-to-end path information to optimize CDN performance", SIGCOMM Internet Measurement Conference (IMC), pp. 190-201, 2009. doi: 10.1145/1644893.1644917
- [11] Qianwen Yin, Jasleen Kaur, F. Donelson Smith, "Can bandwidth estimation tackle noise at ultra-high speeds?", International Conference on Network Protocols (ICNP), pp. 107-118, 2014. doi: 10.1109/ICNP.2014.31
- [12] Stefano Salsano, Fabio Patriarca, Francesco Lo Presti, Pier Luigi Ventre, Valerio Maria Gentile, "Accurate and Efficient Measurements of IP Level Performance to Drive Interface Selection in Heterogeneous Wireless Networks", IEEE Transactions on Mobile Computing, vol. 17, n. 10, pp. 2223-2235, 2018. doi: 10.1109/TMC.2018.2807842
- [13] Hong Zhang, Junxue Zhang, Wei Bai, Kai Chen, Mosharaf Chowdhury, "Resilient Datacenter Load Balancing in the Wild", ACM Special Interest Group on Data Communication (SIGCOMM), pp. 253-266, 2017. doi: 10.1145/3098822.3098841
- [14] Frank Dabek, Russ Cox, Frans Kaashoek, Robert Morris, "Vivaldi: a decentralized network coordinate system.", ACM Special Interest Group on Data Communication (SIGCOMM), pp. 15-26, 2004. doi: 10.1145/1015467.1015471
- [15] Jonathan Ledlie, Margo Seltzer, Peter Pietzuch, "Proxy network coordinates", Target, vol. 22, pp. 25, 2008.
- [16] Cristian Lumezanu, Randolph Baden, Neil Spring, Bobby Bhattacharjee, "Triangle inequality variations in the internet", SIGCOMM Internet measurement conference (IMC) pp. 177-183, 2009. doi: 10.1145/1644893.1644914
- [17] Bong-Jun Ko, Sisi Liu, Murtaza Zafer, Ho Yin Starsky Wong, Kang-Won Lee, "Gateway selection in hybrid wireless networks through cooperative probing", Symposium on Integrated Network Management (IM), pp. 352-360, 2007.
- [18] Khulan Batbayar, Roc Meseguer, Emmanouil Dimogerontakis, Leandro Navarro, Ramin Sadre, "Collaborative informed gateway selection in large-scale and heterogeneous networks", Symposium on Integrated Network Management (IM), pp. 337-345, 2019.
- [19] Laisen Nie, Dingde Jiang, Lei Guo, Shui Yu, "Traffic matrix prediction and estimation based on deep learning in large-scale IP backbone networks", Journal of Network and Computer Applications, vol. 76, pp. 16-22, 2016. doi: 10.1016/j.jnca.2016.10.006
- [20] Xianrong Zhen, Li Da Xu, Sheng Chai, "QoS recommendation in cloud services", IEEE Access, vol. 5, pp. 5171-5177, 2017. doi: 10.1109/ACCESS.2017.2695657
- [21] Xiong Luo, Ji Liu, Dandan Zhang, Xiaohui Chang, "A large-scale web QoS prediction scheme for the Industrial Internet of Things based on a kernel machine learning algorithm", Computer Networks, vol. 101, pp. 81-89, 2016. doi: 10.1016/j.comnet.2016.01.004
- [22] G. Vadelou, "Collaborative filtering based web service recommender system using users' satisfaction on QoS", International Conference on Inventive Computation Technologies (ICICT), pp.1-5, 2016. doi: 10.1109/INVENTIVE.2016.7830110
- [23] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, Mic Bowman, "Planetlab: an overlay testbed for broad-coverage services", ACM SIGCOMM Computer Communication Review, vol. 33, n. 3, pp. 3-12, 2003. doi: 10.1145/956993.956995
- [24] "Lightweight HTTP, HTTPS, WebSockets Proxy Server in a single Python file", GitHub Repository, <https://github.com/abhinavsingh/proxy.py>, Última visita: 24 de mayo de 2019.
- [25] Robert Bell, Yehuda Koren, Chris Volinsky, "Matrix factorization techniques for recommender systems", Computer, vol. 42, pp. 30-37, 2009. doi: 10.1109/MC.2009.263
- [26] Wancai Zhang, Hailong Sun, Xudong Liu, Xiaohui Guo, "Temporal QoS-aware web service recommendation via Non-negative Tensor Factorization", Conference on World Wide Web (WWW), pp. 585-596, 2014. doi: 10.1145/2566486.2568001
- [27] Ruslan Salakhutdinov, Andriy Mnih, "Probabilistic Matrix Factorization", Conference on Neural Information Processing Systems (NIPS), pp. 1257-1264, 2007.

- [28] "AMF: Adaptive Matrix Factorization for Online QoS Prediction", GitHub Repository, <https://github.com/wsdream/AMF>, Última visita: 20 de mayo de 2019.
- [29] "Towards Open Datasets and Source Code for Web Service Recommendation", GitHub Repository, <https://github.com/wsdream/AMF>, Última visita: 20 de mayo de 2019.