ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Análisis de orientación de anuncios en páginas web utilizando Mahout para procesamiento masivo y escalable de datos”

INFORME DE

MATERIA DE GRADUACIÓN

Previo la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

Presentada por:

MARY VANESSA LLONGO GUAMÁN

RAQUEL MAGDALENA NAVAS ROSALES

GUAYAQUIL – ECUADOR

2012

DEDICATORIA

Este proyecto de graduación fruto de mi esfuerzo y dedicación se lo dedico a la mujer que ha estado conmigo apoyando siempre hasta el día de su muerte, quien a pesar que se fue de éste mundo la quiero mucho, para ti mi linda abuelita Luz, quien ha sido mi verdadera luz para seguir en éste caminar. Además se la dedico a todas las personas que creyeron en mí y que me dieron su apoyo desmedidamente, en especial a mi hermana mayor que en un momento de mi vida no me dejo caer y supo darme palabras de aliento para seguir en mi carrera.

Mary Llongo

DEDICATORIA

Les dedico esto a todos los que me aman.

Raquel Navas

AGRADECIMIENTO

Agradezco primeramente a Dios quien me ha iluminado en esta ardua trayectoria de mi vida y que a pesar de muchas pruebas ha sabido guiarme hasta lograr mi objetivo. A mi madre Carmita, quién me dio su apoyo incondicional para llegar hasta aquí y lograr con sus enseñanzas hacer la mujer que soy, a mi madre abuelita Luz quién día a día me enseño lo mejor de la vida para luchar por lo que quiero, a mi padre que a pesar de todo lo quiero y quién ha inculcado en mi principios hasta que vivió conmigo. A mis hermanos quienes han estado conmigo y han aportado con sus consejos para seguir adelante.

Mary Llongo

AGRADECIMIENTO

Agradezco la culminación de esta etapa de mi vida a mi familia y a las personas que estuvieron a mi lado y que en algún momento me ayudaron.

Raquel Navas

TRIBUNAL DE SUSTENTACIÓN

Ing. Vanessa Cedeño
PROFESORA DE LA MATERIA

Ing. Guido Calcedo
PROFESOR DELEGADO DEL DECANO

DECLARATORIA EXPRESA

"La responsabilidad del contenido de este Trabajo de Grado, nos corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

MARY VANESSA LLONGO GUAMÁN

RAQUEL MAGDALENA NAVAS ROSALES

RESUMEN

Los sistemas de recomendación son muy utilizados hoy en día, son sistemas que brindan a los usuarios sugerencias de diversos ítems relacionados con el tema que estén buscando, estos sistemas han mejorado la interacción entre el usuario y el sitio web, se basan en opiniones de usuarios, su historial e incluso en el contenido del sitio web que se está analizando, para así brindar una recomendación efectiva al usuario final.

El presente proyecto de Análisis de orientación de anuncios en páginas web, consta de 5 capítulos en el primer capítulo se analiza los antecedentes, descripción del problema, justificación, objetivo y alcance que con lleva a la realización del presente trabajo.

En el segundo capítulo se muestra los fundamentos teóricos acerca de los sistemas de recomendación que existen.

En el tercer capítulo se hace un análisis de la solución y en éste se describe que tipo de sistema de recomendación se utilizó en nuestro proyecto, el cual provee al usuario ítems relacionados al tema que esté buscando, este análisis se lo realiza teniendo en cuenta diversos parámetros como son sus preferencias pasadas, la evaluación del contenido buscado actualmente en

el sitio web y la relación del ítem buscado con otros ítems parecidos de otros usuarios. Para ello al momento de implementar el sistema se debe usar un sistema híbrido, el cual nos provee un mejor análisis de los ítems a recomendar debido a que explotan las características de los sistemas de recomendación de filtrado colaborativo basado en ítems y el sistema basado en contenido.

En el capítulo cuatro se detalla el diseño y la metodología utilizada para realizar la prueba de los algoritmos utilizados para brindar las recomendaciones a los usuarios, y para ello se trabajó con un dataset ya prestablecido.

En el capítulo cinco se presenta las pruebas y resultados de los datos a recomendar, el tiempo de ejecución de los datos.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

ABREVIATURAS

SR	Sistema de Recomendación
SSH	O SecureSHell es un protocolo para accesar de forma segura entre una computadora y otra.

INTRODUCCIÓN

Con la existencia de los sistemas de recomendación se ha mejorado mucho la interacción que tiene el usuario con los sitios web, ya que estos facilitan la búsqueda de información que se ajuste a las preferencias y necesidades del usuario.

Con la gran cantidad de información que un sitio web puede alcanzar a manejar se necesita tener un alto desempeño y precisión a la hora de procesar los datos. Una gran solución a esto es el uso de la plataforma Apache Mahout que esta implementado sobre Apache Hadoop usando el paradigma map/reduce, cuyo objetivo es construir librerías escalables de machine learning. Mahout contiene implementaciones para el agrupamiento, clasificación, recomendación y programación evolutiva.

Partiendo de este principio, este trabajo realiza un análisis de un sistema de recomendación que muestre anuncios en las páginas web que sean del interés para el usuario.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN.

1.1. Antecedentes y Descripción del Problema

Antecedentes

Existe un tipo especial de motor de recomendación cuyo objetivo es la publicidad. Este selecciona los anuncios que mejor se adapten a un visitante en particular. Además cada anunciante está dispuesto a pagar una cierta cantidad para que su anuncio sea visto, quieren que sus anuncios se muestren a las personas más proclives a comprar sus productos. Esto crea un reto complejo de optimización pues se tiene que procesar una gran cantidad de datos en tiempo real y luego presentar un anuncio que sea de interés para el usuario.

Descripción del Problema

La gran cantidad de información que existe en muchos sitios web puede resultar abrumadora para el usuario y conduce a que tome malas decisiones.

Justificación

Con la rápida evolución del comercio en línea existe la necesidad de mercados que se anticipen a las necesidades del consumidor.

1.2. Objetivo General

Experimentar un mecanismo que recomiende anuncios en las páginas web basándose en las preferencias de los usuarios sin que ellos los hayan solicitado explícitamente, usando un framework para el procesamiento masivo de los datos.

1.2.1. Objetivos Específicos

- Definir la metodología para recomendación de anuncios basándose en las preferencias del usuario prestablecidas en un peso.

- Realizar pruebas del uso del algoritmo para recomendación de anuncios, con un dataset de 111900 preferencias de usuarios el cual está estructurado por el identificador del usuario, el identificador del ítem y el peso.

1.3. Alcance

Se realizará un listado de los anuncios relevantes por cada usuario que haya estado en sesión de la página web, esa información la obtendremos a partir de un archivo de registros.

CAPÍTULO 2

2. FUNDAMENTOS TEÓRICOS DE LOS SISTEMAS DE RECOMENDACIÓN.

2.1. Sistemas de recomendación

Los sistemas de recomendación aplican técnicas para resolver el problema de realizar recomendaciones personalizadas para la información, productos o servicios durante las interacciones en vivo [8].Los sistemas de recomendación, captan opiniones de usuarios, acerca de productos, sitios web, personas, etc. (en general, ítems), clasifican tales opiniones, y las usan luego para sugerir nuevos ítems, o para predecir la utilidad de un ítem para un usuario particular.

2.1.1. Enfoques de recomendación

Existen los siguientes métodos de recomendación [10]:

- Filtrado colaborativo
- Filtrado basado en contenido
- Demográfico
- Basado en conocimiento
- Basado en comunidad

- Híbridos

Para comenzar este proyecto desde la fase de recolección de preferencias en tiempo real, se recomienda usar un sistema de recomendación híbrido la cual combina la recomendación basada en contenido y la recomendación por filtrado colaborativo, con el fin de aprovechar las ventajas del basado en contenido para suavizar las desventajas del filtrado colaborativo. En nuestro proyecto vamos a utilizar sólo los algoritmos de filtrado colaborativo, debido a que tenemos un dataset en el cual las preferencias son obtenidas por filtrado colaborativo.

2.1.1.1. Sistemas basados en contenido

Estos tipos de sistemas recomiendan ítems que son similares a los ítems que previamente fueron valorados por el usuario. Es decir las recomendaciones de ítems son hechas exclusivamente basadas en el historial de todos los ítems que le gustaron.

Los ítems en este tipo de sistema, se definen según sus características. (Ej.: palabras en el documento)

El perfil de usuario se basa en cómo el usuario valora esas características para obtener un resultado pertinente de lo que busca. [11]

2.1.1.1.1. Ventajas de los sistemas de Basados en Contenido

- La recomendación se basa por contenido que el usuario está analizando y no por opiniones subjetivas de otros usuarios.
- Brinda recomendaciones de acuerdo al historial del usuario.
- No hay dispersión, pues el modelado de la información está presente en las características del documento y no necesitan proveerlas otros usuarios.

2.2. Filtrado colaborativo basado en sistemas de recomendación

La idea principal de los algoritmos de filtrado colaborativo es proveer recomendaciones de ítems o predicciones basadas en las opiniones de otros usuarios.

2.2.1. Ventajas de los sistemas de Filtro Colaborativo

Los sistemas de filtro colaborativo tienen importantes ventajas con respecto a los no colaborativos a continuación se detalla varios puntos importantes que hay que considerar:

- Dan un mayor soporte para el filtrado de ítems cuyo contenido no es fácil de analizar por procesos automatizados.
- La posibilidad de filtrar ítems basándose en su calidad o preferencias.
- La posibilidad de realizar recomendaciones válidas, pero que no esperábamos, lo cual puede resultar de gran utilidad.

Sin embargo, también presentan inconvenientes, como por ejemplo que no trabajan bien a la hora de filtrar información para necesidades de contenido específico, o cuando el número de usuarios es bajo. Por ello, en muchas ocasiones la mejor opción es adoptar un enfoque híbrido entre colaborativo y no colaborativo y de esta forma disfrutar de las ventajas de ambos. [8]

2.2.2. Categorías de los algoritmos de filtrado colaborativo

2.2.2.1. Basados en usuario

El algoritmo colaborativo basado en usuario utiliza toda la base usuario-ítem para dar una predicción. Estos sistemas emplean técnicas estadísticas para encontrar un grupo de usuarios, conocidos como vecinos, que tienen una historia de acuerdos con el usuario objetivo. Una vez que una vecindad de usuarios es formada, estos sistemas usan diferentes algoritmos para combinar las preferencias de vecinos para producir una predicción o recomendaciones top-N para el usuario activo.

2.2.2.2. Basados en ítem

Los algoritmos de filtrado colaborativo basados en ítem proveen recomendaciones de ítems desarrollando un modelo de las puntuaciones de los usuarios sobre los ítems. En los algoritmos en esta categoría se utiliza una aproximación probabilística que calcula el valor esperado de una predicción del usuario a partir de sus puntuaciones sobre otros ítems.

2.2.3. Proceso del filtrado colaborativo

En esta sección describimos el proceso que debemos seguir en los Sistemas de Recomendación de filtrado colaborativo y los datos que se analizan para llegar a determinar los ítems que se desean recomendar a un usuario en particular. Como se mencionó antes en el capítulo 2.2 sobre los sistemas de filtrado colaborativo, podemos concluir que estos Sistemas de Recomendación basados en ítems en lugar de buscar similitud entre los usuarios buscan cercanías entre elementos. El procedimiento consiste en seleccionar los elementos que un usuario determinado ha votado y después comprobar como es de similar cada uno del resto de los elementos del sistema, para terminar recomendando los más parecidos. Existen distintas formas de evaluar la similitud entre elementos pero el procedimiento genérico consiste en tomar dos elementos x1, x2 y después calcular su similitud a partir de todos los usuarios que han votado ambos elementos. En teoría es la misma aproximación que la que se tenía con algoritmos basados en vecinos cercanos. [9]

Hay que recordar que los algoritmos que se suelen utilizar para implementar las técnicas de filtrado colaborativo se llaman métodos basados en vecindad.

2.2.4. Funcionamiento de los algoritmos de filtrado colaborativo

Funcionan seleccionando un conjunto apropiado de usuarios, según la similitud de los mismos con respecto al usuario activo, y usan las valoraciones de dichos usuarios para generar la valoración del usuario activo. Concretamente, los tres pasos a seguir para realizar esto son los siguientes:

1. Medir la similitud de todos los usuarios con respecto al usuario activo.

2. Seleccionar un subconjunto de usuarios cuyas valoraciones se van a usar y por tanto, tendrán influencia en la generación de la predicción para el usuario activo.

3. Normalizar las puntuaciones de los distintos usuarios y calcular una predicción a partir de algún tipo de combinación

basada en las puntuaciones asignadas al ítem por los usuarios seleccionados en el paso anterior. [8]

La ventaja es que en el caso de los elementos la similitud entre ellos es menos variable que la similitud entre usuarios, lo que permite pre-computar estas similitudes y hace el proceso mucho más rápido. [9]

2.2.5. Ventajas de los Sistemas de Recomendación de Filtrado Colaborativo basado en ítems

Los sistemas de recomendación basados en usuario no son tan rápidos y escalables a la medida que nos gustaría que fueran, sobre todo en el contexto de los sistemas reales que generan recomendaciones en tiempo real sobre datasets muy grandes. Para alcanzar estos objetivos, se utiliza los sistemas de recomendación basados en ítems, conocido también como basado en modelo.

Los sistemas de recomendación basados en ítems implican la construcción de un modelo basado en el conjunto de datos de las puntuaciones. En otras palabras, extraer alguna información del conjunto de datos, y usar eso como un "modelo" para hacer

recomendaciones sin tener que utilizar el conjunto completo de datos cada vez. Este enfoque ofrece potencialmente los beneficios de la velocidad y la escalabilidad [21].

Según los resultados de unas pruebas experimentales los algoritmos basados en ítems muestran un mejor rendimiento que los algoritmos basados en el usuario, mientras que al mismo tiempo proporciona una mejor calidad que los mejores algoritmos disponibles basados en el usuario [22].

A continuación listaremos las ventajas:

Escalabilidad: La mayoría de los modelos resultantes de los algoritmos basados en el modelo son mucho más pequeños que el conjunto de datos reales, por lo que, incluso para grandes datasets, el modelo termina siendo lo suficientemente pequeño como para ser utilizado de manera eficiente. Esto imparte escalabilidad al sistema global.

Velocidad de predicción: los sistemas basados en el modelo son también probables a que sean más rápido, al menos en comparación con los sistemas basados en usuario, conocidos

como basados en memoria también, ya que el tiempo necesario para consultar el modelo (al contrario de todo el dataset) es generalmente mucho menor que la requerida para consultar todo el dataset.

Evitan el sobreajuste: Si el dataset sobre los que construimos nuestro modelo es lo suficientemente representativo de datos del mundo real, es más fácil tratar de evitar el sobreajuste con sistemas basados en el modelo.

2.3. Sistemas Híbridos

Los sistemas híbridos explotan las características de dos o más tipos de sistemas de recomendación para obtener mejores recomendaciones.

Es importante observar que los enfoques de filtrado basados en contenidos y colaborativos no son mutuamente exclusivos, sino que pueden ser integrados en un mismo sistema para proporcionar sistemas híbridos más potentes.

En efecto, los sistemas de filtrado colaborativos son herramientas muy potentes para realizar el filtrado de información. Sin embargo, para

que sean completamente potentes es necesario combinarlos con técnicas de filtrado basadas en contenidos. Los sistemas colaborativos realizan buenas predicciones de ítems que casan con las preferencias e intereses de los usuarios, pero no trabajan tan bien a la hora de filtrar información para necesidades de contenido específicas.

Otro aspecto a tener en cuenta es que los sistemas colaborativos, como ya hemos visto, generan recomendaciones a partir de opiniones y preferencias de otros usuarios por lo que para un buen funcionamiento del sistema, se hace necesario contar con un cierto número de usuarios. Cuando un sistema ya se ha diseñado e implantado y empieza a funcionar, normalmente el número de usuarios con los que se cuenta es muy bajo, por lo que en este caso, se podría empezar trabajando con un filtrado basado en contenidos y cuando se llegue a un número de usuarios ya aceptable, pasar al filtrado colaborativo.

Es por esto que en muchas ocasiones la mejor opción es adoptar un enfoque híbrido entre colaborativo y basado en contenidos y de esta forma disfrutar de las ventajas de ambos [20].
Lo que se busca es sobrellevar las desventajas de los sistemas de recomendación [14].

2.4. Formación de vecindades

Una vecindad es un conjunto de usuarios que comparten gustos similares y que puede estar restringido por uno de los siguientes factores: la cantidad de individuos que lo forman o un valor mínimo de similaridad.

El primer factor determina que una vecindad está formada por los n vecinos más cercanos al usuario objetivo, es decir los n usuarios cuya similaridad con el usuario objetivo es la más alta.

El segundo factor determina que una vecindad está restringido por un umbral de similaridad, es decir que los usuarios que forman la vecindad son aquellos cuya similaridad con el usuario objetivo es mayor o igual a un valor.

Dado que el algoritmo de filtrado colaborativo basado en ítems, es el tipo de algoritmo sobre el que se centrará el presente proyecto, pasamos a continuación a describir distintas técnicas para obtener el modelo de recomendación.

- **Redes bayesianas:** Esta técnica consiste en obtener el modelo a partir de un conjunto de entrenamiento con un árbol de decisión, donde los nodos y ramas representan información de los usuarios. Suele resultar útil cuando el conocimiento sobre el perfil de usuario cambia de forma lenta.

- **Modelos basados en reglas:** El procedimiento es similar al de la técnica anterior, con la diferencia de que el modelo obtenido tiene la forma de un conjunto de reglas del tipo, Antecedente ->Consecuente'.

- **Técnicas de agrupamiento (clustering):** Hacen grupos de usuarios, denominados clusters, con gustos similares. Las predicciones para un individuo se realizan mediante la agregación de opiniones de otros usuarios del mismo grupo. La tarea de creación de los clusters conlleva un gran esfuerzo, pero una vez creados se obtiene un rendimiento bueno, ya que el grupo a considerar para realizar las recomendaciones queda considerablemente reducido.

Modelos basados en vecino más cercano: Estos algoritmos proporcionan recomendaciones desarrollando en primer lugar un

modelo, utilizando cualquiera de las tres técnicas anteriormente descritas, de las puntuaciones de los usuarios sobre los ítems.

Estos algoritmos miran en el conjunto de ítems evaluados por el usuario activo para calcular cómo de parecidas son estas puntuaciones al ítem activo, con el fin de realizar una predicción para el mismo. Para realizar las recomendaciones se realizan las siguientes tareas:

Paso 1: Exploración del conjunto de ítems que el usuario ha valorado.
Paso 2: Cálculo de la similitud de los ítems anteriores con respecto al ítem o producto del cual queremos predecir la puntuación que el usuario le daría.
Paso 3: Selección de los k ítems más cercanos (Knn).
Paso 4: Cálculo de la predicción del usuario objetivo sobre el ítem dado como la media ponderada de las valoraciones del usuario hacia los k ítems más similares.

Dentro de los Sistemas de Recomendación Colaborativos basados en ítem, emplearemos este último modelo para el algoritmo de filtrado de este proyecto. [23]

Si estudiamos a alto nivel este enfoque colaborativo y el anterior basado en contenidos, podemos observar que existe cierto grado de simetría entre ambos. En efecto, en ambos casos se hace uso de un vector de evaluaciones de los ítems ya experimentados por parte del usuario activo, que denotamos como A.

En el enfoque de Filtrado colaborativo, se cuenta además con un vector Aj para cada uno de los colaboradores indicando sus evaluaciones de los ítems correspondientes. Para cualquier ítem d que el usuario activo aún no haya experimentado, se dispone de un vector R cuyos componentes, rj son las evaluaciones de dicho ítem por parte de los colaboradores. El proceso de obtención del grado de recomendación, básicamente se puede dividir en los dos pasos siguientes ver figura 2.1:

Paso 1: Combinar A y Aj para obtener Sj , que mide el grado de similaridad entre nuestro usuario y cada uno de los colaboradores.

Paso 2: La predicción de la evaluación del ítem d se calcula mediante una agregación ponderada de las listas de los vectores Sj y rj .[20]

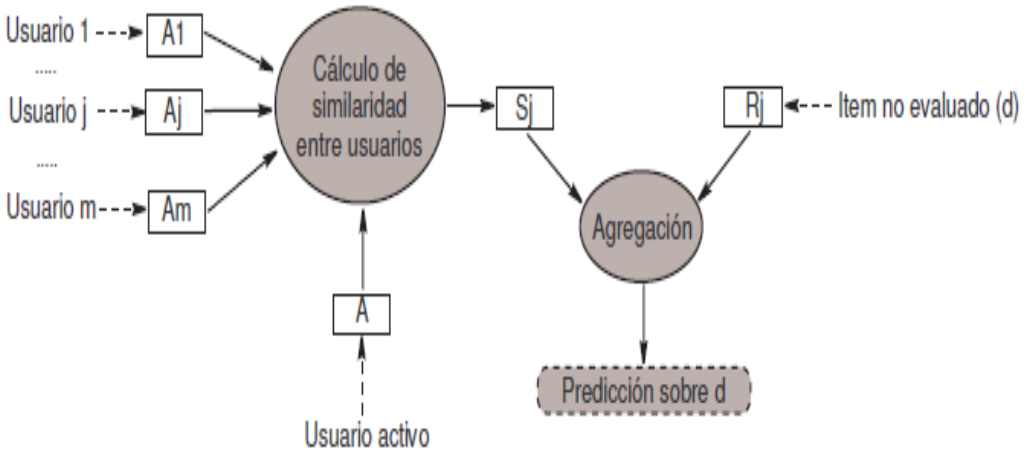

Figura 2. 1 : Esquema de Filtrado Colaborativo

2.5. Métrica Loglikelihood (logaritmo de verosimilitud)

En estadística, la estimación por máxima verosimilitud (conocida también como EMV y, en ocasiones, MLE por sus siglas en inglés) es un método habitual para ajustar un modelo y encontrar sus parámetros. [18]

El método de máxima verosimilitud fue introducido primero por R. A. Fisher, genetista y experto en estadística, en la década de 1920. La mayoría de los expertos en estadística recomiendan este método, al

menos cuando el tamaño muestral es grande, porque los estimadores resultantes tienen ciertas propiedades deseables de eficiencia.

- *Sea X una variable aleatoria con función de probabilidad f(x|θ), donde θ es un parámetro desconocido. Sean $X_1,\ldots, X_n$ los valores observados en una muestra aleatoria de tamaño n. La función de verosimilitud de la muestra es:*

$$L(\theta) = \prod_{i=1}^{n} f(x_i|\theta) \quad \text{(Ec. 1)}$$

Debemos considerar que la ecuación 1 es la función de densidad conjunta de la muestra aleatoria. Notemos que la función de verosimilitud es una función del parámetro desconocido *θ*.

- *El estimador de máxima verosimilitud de θ es el valor de θ que maximiza la función de verosimilitud L(θ).*

En ocasiones es más simple maximizar la función log-verosimilitud que la ecuación 1, dada por:

$$l(\theta) = \log\big(L(\theta)\big) = \sum_{i=1}^{n} \log f(x_i|\theta) \quad \text{(Ec. 2)}$$

El método de máxima verosimilitud puede emplearse en situaciones donde existen varios parámetros desconocidos, $\theta_1, \theta_2,\ldots, \theta_k$, que es necesario estimar. En tales casos, la función de verosimilitud es una función de los k parámetros desconocidos $\theta_1, \theta_2,\ldots, \theta_k$ y los

estimadores de máxima verosimilitud $\hat{\theta}_1$, $\hat{\theta}_2$,…, $\hat{\theta}_k$ se obtienen al igualar a cero las k derivadas parciales, dadas por:

$$\frac{\partial L(\theta_1, \theta_2, \dots, \theta_k)}{\partial \theta_i}, i = 1,2, \dots, k$$

y resolver el sistema resultante. [19]

CAPÍTULO 3

3. ANÁLISIS DE LA SOLUCIÓN.

En este capítulo se citará todo lo referente al diseño de la solución del proyecto.

3.1. Análisis del comportamiento de los usuarios en las páginas web

Una vez que el sitio web ha sido lanzado y es usado diariamente, tenemos a nuestra disposición una nueva fuente de información sobre el comportamiento del usuario: Los ficheros "log".

Estos, son extensos ficheros de texto plano que genera el servidor web, y en los que se registra cada una de las peticiones de páginas realizadas por los clientes al servidor.

Por cada petición del cliente al servidor se suele registrar la siguiente información:

- Dirección IP del cliente

- Identidad del usuario (para sitios con identificación)
- Password de acceso (para sitios con identificación)
- Fecha y hora de la petición
- Método
- Path o directorio de la página en el servidor.
- Código que indica si la petición ha sido resuelta correctamente o no.
- Número de bytes transferidos entre cliente y servidor.
- Página desde la que se pide el archivo al servidor (puede ser una URL interna, si a la página se llega por un enlace del mismo sitio web, o externa, en el caso de que sea a través de otro sitio web).
- Información sobre el agente software (navegador) del cliente.

A través del análisis de los ficheros logs se pueden responder preguntas como: ¿quién usa el sitio? ¿Cuándo lo usa? ¿Qué páginas suelen ser las más visitadas? ¿Desde qué páginas se llega? ¿Qué términos utiliza el usuario para interrogar al buscador interno?.

Se trata realmente de una información muy valiosa que correctamente analizada (normalmente ayudándonos de software específico), puede

servirnos para la toma de decisiones sobre el rediseño en sitios web implementados [13].

3.2. Selección de la categoría de los algoritmos de filtrado colaborativo y no colaborativo

Tomando en consideración que la solución propuesta se centra en los registros del usuario vamos a usar la recomendación basada en ítem y también la basada en contenido. Es decir se implementará un sistema híbrido, ya que éste tipo de sistema abarca todos los requerimientos que deseamos analizar para así realizar una recomendación más efectiva.

3.3. Selección del algoritmo de filtrado colaborativo

Mahout provee dos Map/reduce para apoyar al algoritmo filtrado colaborativo basado en Items, de los cuales se escogió el algoritmo ItemSimilarityJob porque éste algoritmo nos da como resultado los Items recomendados y el peso que tiene cada uno de ellos, éste peso se basa en cuanto se aproxima al gusto del usuario que se está analizando. Así se podrá recomendar ítems

que tengan mayor peso y a la vez será más óptimo para el usuario.

3.4. Requerimientos

Para el desarrollo de nuestro proyecto vamos a utilizar el paradigma Map/Reduce usando Mahout y Apache Solr.

3.4.1. Paradigma Map/Reduce

Paradigma Map/Reduce es un API de programación distribuida. Su implementación fue realizada para brindar soporte a la computación paralela sobre grandes colecciones de datos. Fue propuesto inicialmente por Google y fue implementado en el Apache Hadoop.

3.4.2. Apache Hadoop

Apache hadoop es una plataforma que es muy útil cuando se necesita escalabilidad al momento de realizar algún proyecto, ya que permite procesar y almacenar petabytes de información. Fue creado por Doug Cutting. Soporta el procesamiento de grandes cantidades de datos y brinda servicios de computación

distribuida, escalable y confiable. Hadoop posee licencia libre, este se trata de un subproyecto de Lucene y se basada en java.

3.4.3. Apache Mahout

Apache Mahout es un proyecto de código abierto implementado sobre Apache Hadoop usando el paradigma map/reduce, cuyo objetivo principal es construir librerías escalables de machine learning. Mahout contiene implementaciones para el agrupamiento, clasificación, recomendación y programación evolutiva. Además, Mahout se centra en el mundo real, proporciona diversas soluciones que pueden ser ampliamente aplicados en diversos campos, también proporciona documentación de calidad. Éste proyecto fue lanzado el 7 de abril de 2009 [12].

3.4.3.1. Requisitos

Los requisitos básicos del sistema que necesitamos instalar son los siguientes:

- Netbeans 7.1 o superior
- Java JDK 1.6 o superior

- Apache Mahout 0.4 o superior
- Apache Solr 3.6 o superior
- CPU, disco y requisitos de memoria, se basa en las múltiples opciones que necesitamos para la implementación de la aplicación que se realice con Mahout (tamaño del documento, número de documentos, y el número de accesos).

3.5. Apache Solr

Apache Solr es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas".

Sus principales características incluyen búsquedas de texto completo, resaltado de resultados, clustering dinámico, y manejo de documentos ricos (como Word y PDF).

Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices, y actualmente se está usando en muchos de los sitios más grandes de Internet [17].

3.5.1. Arquitectura de Solr

Solr se divide en dos partes:

- **Índice:** Es el sistema de ficheros que almacenan la información. Contiene la configuración de Solr y la definición de la estructura de datos.

- **Servidor:** Proporciona el acceso a los índices y las características adicionales. Admite plugins para añadir funcionalidades [17], ver figura 3.1.

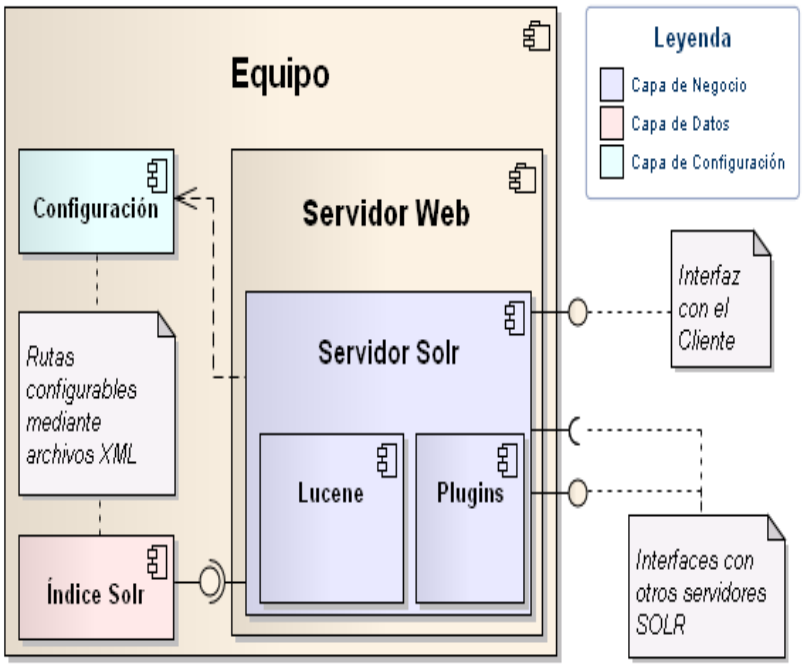

Figura 3. 1: Arquitectura de Solr

3.5.2. Arquitectura Distribuida

Solr permite búsquedas distribuidas, uno de los servidores actúa como maestro, consultando al resto y componiendo la respuesta [17] ver figura 3.2.

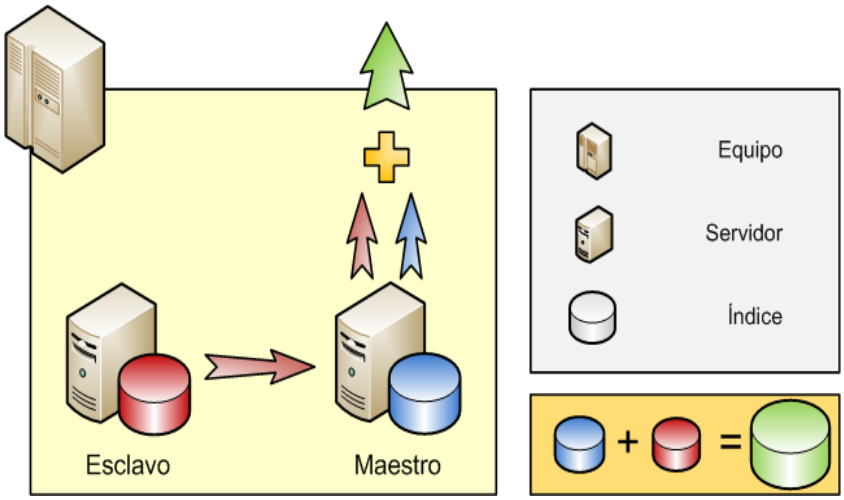

Figura 3. 2: Arquitectura Distribuida de Solr

CAPÍTULO 4

4. DISEÑO Y METODOLOGÍA UTILIZADA.

En este capítulo se describe la metodología usada para el análisis de un sistema de recomendación de anuncios para páginas web.

Primero explicaremos la estructura de los datos relevantes, luego para realizar la metodología hemos divido el proceso en varias fases: la primera consiste en la clasificación de usuarios, la segunda es la obtención de las preferencias de los usuarios sobre los ítems, la siguiente es la selección de los ítems recomendados y por último dependiendo de los ítems recomendados se presentarán los anuncios al usuario.

4.1. Estructura de los datos

4.1.1. Estructura del Dataset

Para el desarrollo de nuestro proyecto utilizamos un dataset, este debe estar compuesto de usuarios, ítems y preferencias. Estos datos fueron generados aleatoriamente en un principio

para los documentos de la Wikipedia, el cual nos proporcionaba información sobre libros y este lo adaptamos a nuestros requerimientos, modificándolo de tal manera que no solo tenga información de libros, si no de diversos temas.

El dataset obtenido tiene la información de 1000 usuarios, 2284 ítems de esos usuarios y un total de 111900 preferencias.

Está estructurado de la siguiente manera:

<id_usuario>,<id_item>,<peso>

El valor del peso ya vino prestablecido en el dataset, algunos de forma manual y otros de forma aleatoria, en la sección 4.3 se describirán los parámetros que se tomarán en cuenta para nuestro análisis y la ponderación que se le dará a este peso cuando este proyecto sea implementado en un sitio web.

4.1.2. Estructura de los anuncios

Los anuncios estarán almacenados de la siguiente manera:

<id del anuncio>, <titulo>, <contenido>,<keywords>

Estos keywords serán las palabras por las cuales el anunciante quiere que su anuncio sea encontrado.

4.2. Clasificación de usuarios y filtrado de la información

Teniendo como antecedentes que los usuarios al momento de hacer una búsqueda de un determinado artículo de interés, ya sea este para la compra o cotización del mismo, son usuarios que muchas veces no tienen idea las características de algún producto o a veces son usuarios que saben mucho del tema y todos ellos buscan recomendaciones para hacer una correcta elección de lo que andan buscando y también se parte de la hipótesis de que existen a veces usuarios anónimos que son aquellos que ingresan por primera vez en una página web y no desean registrarse o usuarios nuevos que son aquellos se registran por primera vez o usuarios que se encuentran registrados.

Por tanto clasificaremos a los usuarios que navegan las páginas web en tres tipos, como se mencionó anteriormente ver figura 4.1.

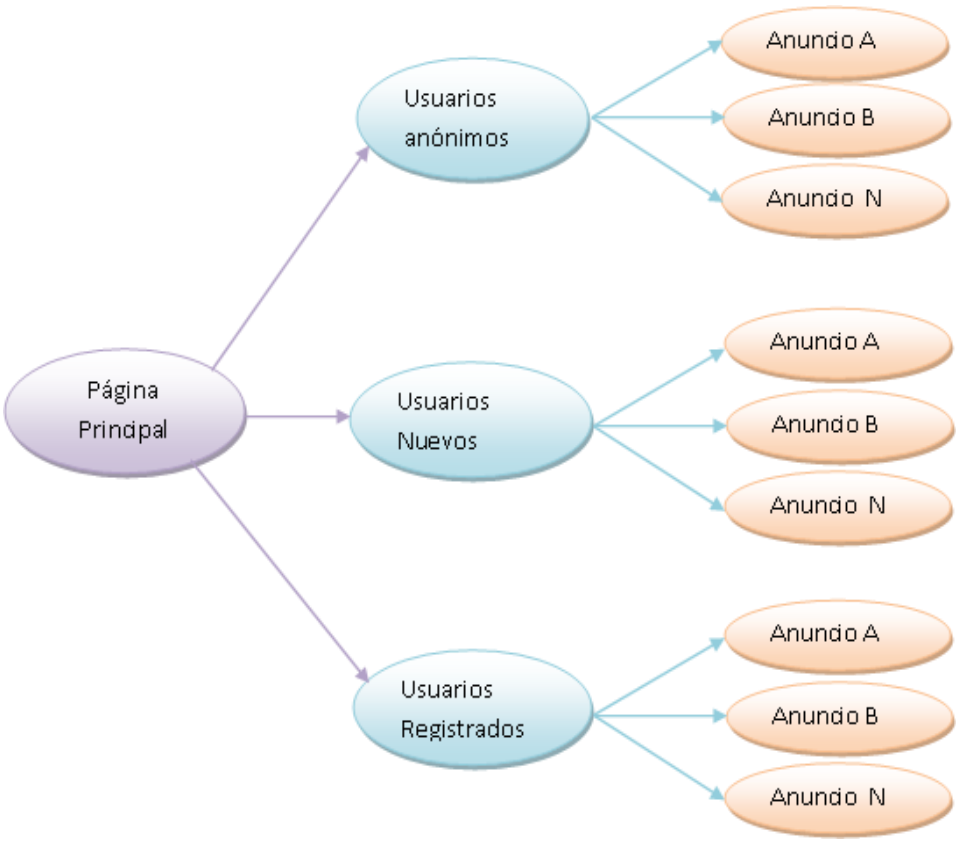

Figura 4. 1: Filtrado de la Información según el tipo de usuario

Para cada uno de estos tipos de usuarios se les evaluará diferentes métodos para realizar una recomendación más efectiva de acuerdo a los parámetros que se analizan de cada usuario.

4.3. Recolección de preferencias

Un aspecto fundamental que se plantea a la hora de diseñar un sistema que utiliza la técnica de filtrado colaborativo y un sistema

basado en contenido es la recolección de preferencias de las personas sobre elementos del sistema.

De esto depende su funcionamiento, dado que el sistema llega a ser útil luego de haber recogido una “masa crítica” de opiniones y del contenido adecuado que se analice al momento de hacer una búsqueda en la página web.

Las preferencias de las personas sobre los elementos se obtienen de forma explícita o implícita. La recolección explícita ocurre cuando el usuario manifiesta su preferencia sobre un ítem, esto será mediante un peso que realice el usuario en una escala numérica.

En cambio, la recolección implícita se basa en obtener las preferencias del usuario analizando las acciones que realiza al utilizar el sistema.

Para recoger las preferencias de forma implícita el sistema debe ser capaz de estudiar el comportamiento del usuario en su utilización, lo cual es claramente una tarea difícil. Además, es preciso considerar que las preferencias que se pueden obtener de esta forma son menos exactas, dado que son estimadas por el sistema y no proporcionadas directamente por el usuario. Sin embargo, esta forma de recolección

resulta más cómoda para las personas, dado que no deben puntuar cada uno de los elementos. [16]

Para el análisis de nuestro trabajo de recomendación de anuncios en páginas Web vamos a recoger información implícita sobre las preferencias, el usuario no debe añadir un peso a los ítems buscados, si no que la evaluación de sus acciones se los hace internamente dando un peso a los ítems (keywords) que se encuentran en una página web que el usuario visita.

Este valor numérico del peso lo calcularemos basado en una fórmula en la que se consideraran varios parámetros como son:

1. Número de veces que ha visitado el usuario esas páginas web que contienen ese ítem (keyword).
2. Tiempo que el usuario se demora en una página.
3. Valor 0 ó 1 que se le da al ítem, si se encuentra este dentro del perfil del usuario se le da un valor de 1, caso contrario su valor será 0.

4.4. Ponderación de los Parámetros del Valor Preferencial del ítem

Las ponderaciones de los parámetros se definen de la siguiente manera ver tabla 4.1.

Parámetro	Ponderación
1	35%
2	25%
3	40%
Total	100%

Tabla 4.1: Ponderación de Parámetros

4.5. Generación de ítems recomendados

Describiremos los procesos para la generación de recomendaciones de ítems por filtrado colaborativo basadas en ítems.

Proceso 1: Configuración del sistema

En la base de datos se tendrá que configurar el número de recomendaciones que se quieren obtener.

Proceso 2: Creación del modelo de datos

En el sistema se instancia un modelo de datos que contiene todas las recomendaciones realizadas implícitamente por el usuario a lo largo del uso del sitio web.

Proceso 3: Obtención de la similaridad del modelo de datos.

Los valores de similaridad son calculados con la métrica LogLikelihood. Este proceso identifica los ítems similares a los que el usuario ha ponderado.

Proceso 4: Obtención de la vecindad.

El resultado del proceso 3 es pasado a un Recomendador que se encarga de crear una lista ordenada por mayor puntuación de las recomendaciones.

```
//Creación del modelo de datos
JDBCDataModel model = new MySQLJDBCDataModel(dataSource,"preferencia","us_id","it_id","pr_peso");
 //Creación de un ItemSimilarity
ItemSimilarityitemSimilarity = new LogLikelihoodSimilarity(dataModel);
 //Crear un Item Based Recommender
ItemBasedRecommender recommender =
        new GenericItemBasedRecommender(dataModel, itemSimilarity);
 //Obtener las recomendaciones
 List<RecommendedItem> recommendations =
recommender.recommend(userId, numRecomendaciones);
```

Figura 4. 2: Fragmento de código para la Generación de Ítems Recomendados

4.6. Generación de ítems recomendados

Para valorar que anuncios son los mejores para la página web, primeramente utilizamos el algoritmo de filtrado colaborativo y el basado en contenido dependiendo del tipo de usuario que se analice, estos algoritmos se encargarán de darnos los ítems recomendados, que es el paso previo para obtener los anuncios.

Luego de saber cuales son los ítems recomendados, se tomaron estos datos para ser analizados con Apache Solr, para determinar los anuncios a recomendar.

4.7. Generación de anuncios recomendados

Luego de haber obtenido los ítems recomendados, estos serán usados como dato de entrada para el algoritmo de búsqueda de anuncios. Para ello se utilizó el algoritmo proveído por Solr.

4.8. Selección de anuncios a recomendar

Para la selección de anuncios a recomendar en un sitio web, se siguieron los siguientes procesos:

Proceso 1: Configuración del sistema

En la base de datos se tendrá que configurar el número máximo de anuncios que se quieren obtener.

Proceso 2: Almacenamiento de los anuncios

Se hace un levantamiento del servidor de Solr y se pasan los anuncios almacenados en la base de datos MySql al servidor de Solr.

```
   /* Esta es la URL donde se encuentra levantado el servidor de
Solr. */
   private static final String URL = "http://localhost:8983/solr";
   private CommonsHttpSolrServer server;
```

Figura 4. 3: Fragmento de código que muestra la url del servidor de Solr.

```
        // Se realiza la consulta. Los resultados se guardan en el
        // ResultSet rs
        ResultSet rs = s.executeQuery ("select an_id, an_titulo from
anuncio");
        // Se recorre el ResultSet, mostrando por pantalla los
resultados.
        while (rs.next())
        {
            //creamos un documento de Solr para insertar.
            //los campos están definidos en Solr, y no podemos
usar campos no definidos.
            SolrInputDocument doc1 = new SolrInputDocument();
            doc1.addField("id", ""+rs.getInt ("an_id"), 1.0f);
            doc1.addField("name", rs.getString ("an_keywords"),
1.0f);
```

```
            Collection<SolrInputDocument> docs = new
ArrayList<SolrInputDocument>();
            docs.add(doc1);
            server.add(docs);
            server.commit();
        }
```

Figura 4. 4: Fragmento de código que almacena los anuncios en el servidor de Solr.

Proceso 3: Obtención de los anuncios.

Se procede a realizar una búsqueda de los ítems recomendados en el servidor de Solr. Y como resultado obtendremos los anuncios cuyos keywords coincida con un ítem de entrada.

```
SolrQuery query = new SolrQuery();
query.setQuery("name:"+this.recomendaciones);
QueryResponse rsp = server.query(query);
SolrDocumentList docList = rsp.getResults();
```

Figura 4. 5: Fragmento de código que busca los anuncios en el servidor de Solr.

4.9. Algoritmo utilizado por tipo de usuario

Existen 3 tipos de usuarios que debemos considerar:

1. Los usuarios que no se registran en la página web.
2. Los usuarios nuevos.
3. Los usuarios registrados.

4.9.1. Los usuarios que no se registran en la página web

Para ellos usaremos el algoritmo que nos devuelve los ítems similares dando como parámetro de entrada el id del ítem. Este id del ítem lo obtendremos mediante el Algoritmo basado en contenido.

4.9.2. Los usuarios nuevos

Para estos usuarios también usaremos el algoritmo que se utiliza en la sección 4.9.1, porque no tienen un historial de actividades, pero sus datos sí serán almacenados en la base de datos.

4.9.3. Los usuarios registrados

Estos usuarios serán tratados con el algoritmo que dado el id del usuario nos devuelve los ítems preferenciales.

Para que el sistema recomendador de anuncios sea usado en el sitio web, este debe contar con el tab keywords en todas sus páginas web, estos keywords serán usados para alimentar la base de datos de recomendaciones de forma implícita, si el usuario está registrado. Y el peso de la recomendación será dado como ya se describió en la sección 4.2.

Los pasos son los siguientes:

1.- El sistema dependiendo de qué tipo de usuario es, ejecuta el algoritmo que dará las recomendaciones de los ítems (keywords).

2.- Una vez que se tiene el listado de ítems recomendados, entonces estos ítems sirven de entrada para otro algoritmo que utiliza Apache Solr, que se encargará de darnos los anuncios que coincidan con uno o varios ítems (keywords) recomendados.

4.10. Presentación de Resultados

Los resultados se almacenan en una base de datos MySql, el número de recomendaciones que se desee por usuario que sean almacenadas dependerá de la configuración del administrador del sitio web.

El diseño de la tabla que guarda las recomendaciones de los ítems cuenta con los siguientes campos: id del usuario, id del ítem, y el valor del peso.

El proceso de generar las recomendaciones de los anuncios es un proceso en línea que se ejecuta cuando el usuario ingresa a una página, por lo cual el hardware utilizado debe ser robusto. Los resultados de los anuncios serán presentados en el instante que el usuario este navegando en el sitio web.

CAPÍTULO 5

5. PRUEBAS Y RESULTADOS.

En este capítulo se hace una descripción de las pruebas efectuadas a los algoritmos utilizados para verificar su funcionamiento al generar las recomendaciones. Además se muestra un análisis de los resultados de las pruebas del sistema.

5.1 Ejecución de las pruebas

En las aplicaciones que utilizan técnicas de filtrado colaborativo y basado en contenido nos encontramos con la dificultad de la gran cantidad de datos confiables que necesita este tipo de sistema para hacer las pruebas. Por ello en nuestro proyecto utilizamos información proporcionada por la Wikipedia, el cual nos proveía 10000 datos de distintos usuarios.

5.1.1. Ejecución del Algoritmo para Recomendación de Ítems

Se necesitaba tener una gran cantidad de datos para obtener recomendaciones de ítems parecidos al que el usuario estaba buscando en el web site, por consiguiente se utilizó los datos proveídos por un sitio web, como se explicó en la sección 5.1.

Dado un id del usuario 995, ver tabla 5.1, el algoritmo de recomendación le provee 5 recomendaciones de ítems parecidos evaluando su preferencia, cada una de estas recomendaciones son proporcionadas con un peso que va en un rango de -5 a 5, el cual indica el nivel de similitud, siendo -5 menos similar y 5 el más similar.

Cabe recalcar que el número de recomendaciones dependerá de lo que requiera el administrador del web site, nosotros para nuestras pruebas vamos a mostrar los 5 ítems más parecidos.

A continuación se muestra la salida de las recomendaciones para ese usuario ver tabla 5.1.

Id Usuario	995		
Recomendaciones			
Nº	**Id ítem**	**Keyword**	**Peso**
1	2954818	internet	5.0
2	133164859	paginas-web	5.0
3	3366480	seguridad	5.0
4	133105710	redes	5.0
5	4083089	seguridad-web	5.0

Tabla 5.1: Resultado de Recomendaciones de ítems

5.1.2. Ejecución del Algoritmo para Recomendación de anuncios

Luego de tener la lista de ítems recomendados, se procede a que estos datos sean la información de entrada para el algoritmo de recomendación de

anuncios, para ello se utilizó Apache Solr, como se describió anteriormente en el capítulo 3.5.

5.1.3. Tiempo de ejecución del algoritmo de Mahout y Solr

Durante la ejecución del algoritmo se demoro el tiempo que se muestra en la tabla 5.2.

Tiempo de ejecución	**2 min**

Tabla 5. 2: Tiempo de ejecución de los algoritmo de recomendaciones

Este tiempo puede disminuir o aumentar dependiendo del hardware utilizado, mientras más robusto sea, más rápido se ejecutara.

5.2 Análisis de los resultados

En general no se realizan buenas recomendaciones, si no se posee una gran cantidad de información.

5.2.1. Análisis de los resultados de ítems recomendados

Como se puede observar en la tabla 5.1 de la sección 5.1.1, podemos concluir que los ítems recomendados para ese usuarios son 100% parecidos al ítem que el andaba buscando, debido a que el peso que arroja el algoritmo de Mahout a cada uno de los ítems recomendados tienen el máximo valor y esto hizo que sus recomendaciones se apeguen a sus preferencias.

Por ello se puede inferir que al hacer la valoración implícita el usuario tuvo una gran cantidad de información válida para ser utilizada al momento que el algoritmo realiza su ejecución.

5.2.2. Análisis de los resultados de anuncios recomendados

Dado la cantidad de ítems recomendados por el algoritmo de Mahout ver tabla 5.1, estos son analizados por el algoritmo de Solr y proveen una lista de anuncios relacionados con esos keywords, ver tabla 5.3.

Nº de anuncio	Anuncio
1	Curso práctico de diseño de paginas web
2	Publicidad: publicidad en internet, publicidad efectiva, publicidad web, publicidad para empresas.
3	Promotor internet
4	Cursos de 3Dmax, revit architecture, autocad, animación 3d, v-ray, mental ray, diseño web

Tabla 5. 3: Listado de anuncios a recomendar

CONCLUSIONES

1. Los sistemas de recomendación son herramientas que han mejorado la interacción de usuarios con los sitios web, ayudan a los usuarios a facilitar la búsqueda de información en una página web, debido a que brinda recomendaciones personalizadas, tomando en cuenta diversos parámetros que son evaluados por parte del usuario y también por datos referentes al comportamiento del usuario en la página web.

2. A medida que la cantidad de usuarios e ítems crece, también crece la cantidad de cómputo de vecinos más cercanos para la determinación de usuarios similares y como los cálculos se hacen en tiempo real el sistema tiende a colapsar, por lo cual se debe utilizar mejores recursos de hardware y software, que ayude a procesar gran cantidad de información y disminuya el tiempo de ejecución.

3. El tiempo que demora realizar una recomendación de anuncios a un usuario es aproximadamente de 2 minutos. Este tiempo se dio usando un hardware con 4GB de memoria RAM, un procesador de 2.40 Ghz y un sistema operativo de 64 bits. De manera general se concluye, que para que las recomendaciones logren alcanzar buenos tiempos de

respuesta durante las consultas en tiempo real, es necesaria una mejora en el hardware.

4. El tiempo de ejecución disminuirá si se utiliza Apache Hadoop, ya que al momento de hacer las recomendaciones internamente utilizaría más nodos y de esta manera distribuiría la carga al procesar la cantidad de información.

RECOMENDACIONES

1. Se recomienda que el usuario suministre al sistema, información suficientemente relevante sobre sus preferencias, y de esta manera se brindará una recomendación más efectiva con respecto a lo que el usuario anda buscando, porque de esta forma logrará tener un grupo bien definido de vecinos cercanos para realizar la recomendación y así disminuirá el problema de ColdStart.

2. El algoritmo de recomendación utilizado en este proyecto brinda una lista de recomendaciones acopladas a los requerimientos del usuario que hace la consulta, por ello se recomienda que al momento de presentar estos anuncios en el web site, el administrador coloque estos anuncios en un rotador, para así poder presentar todos los anuncios que fueron brindados por el recomendador, y no se limitaría a presentar un número específico de anuncios.

3. Los cálculos de las recomendaciones son en tiempo real, así que requieren de un mayor costo de hardware, dependiendo del número de usuarios que tenga la página web.

4. En un futuro se puede probar la efectividad de los anuncios recomendados a los usuarios, tomando como base este proyecto y realizando una aplicación web para visualizarla.

ANEXOS

ANEXO A

Instalación de Mahout

Mahout está escrito en java por eso necesitamos el jdk (java development kit) para compilar modificaciones que le hagamos al código.

Antes de instalar el Apache Maven se debe tener instalado del jdk, para saber como instalar consulte el anexo A.

Instalar apache maven

La librería Mahout tiene una estructura dada por el gestor de proyectos Apache Maven. Para instalar Maven sigue los pasos que están al final de esta página http://maven.apache.org/download.html o sigue los pasos que se muestran a continuación.

Descárgalo de la página oficial. Escoge la opción con extensión **tar.gz** (Para el ejemplo usaré la version 2.0.11)

Copia la descarga en el directorio

/opt

sudocp Descargas/apache-maven-2.0.11-bin.tar.gz /opt/

Ve hacia ese directorio y descomprímelo

cd /opt

sudo tar -xvzf apache-maven-2.0.11-bin.tar.gz

Crear la variable de ambiente M2_HOME

En consola:

cd --

nano .profile

Agregamos al final delarchivo

M2_HOME=/opt/apache-maven-2.0.11 export M2_HOME

PATH=$PATH:$M2_HOME/bin

Guardamos el archivo y escribimos

source .profile

para que los cambios sean leídos por el sistema

Comprobar que éste instalado con

mvn -version

vanessa@vanessa-desktop:~$ mvn -version

Warning: JAVA_HOME environment variable is not set.

Apache Maven 2.0.11 (r909250; 2010-02-12 00:55:50-0500)

Java version: 1.6.0_22

Java home: /usr/lib/jvm/java-6-sun-1.6.0.22/jre

Default locale: es_EC, platform encoding: UTF-8

OS name: "linux" version: "2.6.35-22-generic" arch: "i386" Family: "unix"

El warning que aparece es por no establecer la variable de entorno

JAVA_HOME

Establecer JAVA_HOME

Una vez instalado el jdk se debe establecer la variable de entorno JAVA_HOME

Agregamos la variable al archivo .profile y cambiamos el PATH

nano .profile

Añadimos antes de PATH

JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.22 export JAVA_HOME

Añadimos en el PATH la variable

PATH=$PATH:$JAVA_HOME/bin

Al final nosquedaráasí

M2_HOME=/opt/apache-maven-2.2.1 export M2_HOME

JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.22 export JAVA_HOME

PATH=$PATH:$JAVA_HOME/bin:$M2_HOME/bin

Guardamos el archivo y escribimos en consola

source .profile

Comprobamos que maven este instalado correctamente

mvn -version

Y el warning por java desapareció.

Mahout

Descarga Mahout .Escoge la que tiene extensión tar.gz *http://www.apache.org/dyn/closer.cgi/mahout/*

Copia la carpeta descargada a donde desees, nosotros lo haremos en /home/vanessa. Se utilizará la versión 0.4 de Mahout.
cp Descargas/mahout-0.4-src.tar.gz /home/vanessa/

Ve al directorio donde la copiaste y descomprime el archivo

```
cd /home/vanessa
sudo tar -xvzf mahout-0.4-src.tar.gz
```

Opcional: Elimina el archivo comprimido (ya lo tenemos la carpeta descomprimida)

rm -rf mahout-0.4-src.tar.gz

Dar permisos en las carpetas de mahout

```
sudo chmod -R 777 mahout-0.4/*
```

Ve al directorio principal de mahout

```
cd mahout-0.4
```

Escribe este comando que compilara los archivos y evitara correr los tests.
mvninstall -DskipTests=true

Si todo sale bien al final BUILD SUCCESSFUL entonces puedes trabajar tranquilamente con Mahout. -DskipTests=true es para evadir los test al momento de compilar, estos muchas veces dan problemas.

BIBLIOGRAFÍA

[1] **LAM CHUCK, VENNER JASON, WHITE TOM**, HADOOP IN ACTION,PRO HADOOP,HADOOP: THE DEFINITIVE GUIDE, http://es.wikipedia.org/wiki/Hadoop, 2010, 2009, 2009, 325, 440,524

[2] **TECHTERMS,** DEFINITION HADOOP**,** http://searchcloudcomputing.techtarget.com/definition/Hadoop, 2008

[3] **SACRISTÁN LUIS ,** HADOOP PLATAFORMA PARA TRABAJAR CON GRAN CANTIDAD DE DATOS**,** http://sentidoweb.com/2007/11/21/hadoop-plataforma-para-trabajar-con-gran-cantidad-de-datos.php, 2007

[4] **NAVARRETE SUÁREZ SERGIO**, HADOOP EN ACCIÓN CLUSTER DE BAJO PERFIL PARA EL ANÁLISIS DE GRANDES VOLÚMENES DE DATOS, http://www.slideshare.net/campuspartycolombia/hadoop-en-accion, 2011

[5] **SARWAR BADRUL, KARYPIS GEORGE, KONSTAN JOSEPH, AND RIEDl JOHN**, ITEMBASED COLLABORATIVE FILTERING RECOMMENDATION ALGORITHMS**,** http://www.www10.org/cdrom/papers/pdf/p519.pdf , 2001, 1-11

[6] **DHRUBA BORTHAKUR,** HDFS ARCHITECTURE GUIDE,http://hadoop.apache.org/common/docs/current/hdfs_design.html#Introduction, 2011

[7] **LAM CHUCK, VENNER JASON, WHITE TOM**,HADOOP IN ACTION, PRO HADOOP, HADOOP: THE DEFINITIVE GUIDE, http://es.wikipedia.org/wiki/Hadoop#Hadoop_Distributed_File_System, 2010, 2009, 2009, 325, 440, 524-526,

[8] **HERRERA-VIEDMA ENRIQUE, PORCEL CARLOS, HIDALGO LORENZO**, SISTEMAS DE RECOMENDACIONES: HERRAMIENTAS PARA EL FILTRADO DE INFORMACIÓN EN INTERNET, http://www.hipertext.net/web/pag227.htm#3.3.1, 2004

[9] **GALÁN NIETO SERGIO MANUEL**, FILTRADO COLABORATIVO Y SISTEMAS DE RECOMENDACIÓN,http://www.it.uc3m.es/jvillena/irc/practicas/06-07/31.pdf,2011

[10] **CHRISTENSEN INGRID,** PERSONALIZACIÓN Y RECOMENDACIÓN, http://www.exa.unicen.edu.ar/catedras/optia/public_html/Recomendacion.pdf, 2012 ,2-4

[12] **THE APACHE SOFTWARE FOUNDATION**, WHAT IS APACHE MAHOUT?, http://mahout.apache.org/, 2011

[13] **HASSAN YUSEF, MARTÍN FRANCISCO, IAZZA GHZALA,** DISEÑO WEB CENTRADO EN EL USUARIO: USABILIDAD Y ARQUITECTURA DE LA INFORMACIÓN, http://www.hipertext.net/web/pag206.htm#5.5.2, 2011

[14] **RODRÍGUEZ ANTONIO,** SISTEMA DE RECOMENDACIÓN COLABORATIVO BASADO EN ALGORITMOS DE FILTRADO MEJORADO, http://sinbad2.ujaen.es/sinbad2/files/pfc/pfc_albin.pdf, 2011, 10.

[15] **G. HUECAS, J.SALVACHÚA ANTONIO,** FILTROS COLABORATIVOS Y SISTEMAS DE RECOMENDACIÓN, http://www.slideshare.net/ghuecas/filtros-colaborativos-y-sistemas-de-recomendacin, 2011, 22-30.

[16] **BETARTE LETICIA, MACHADO RODRIGO, MOLINA VALERIA**, PGMÚSICA SISTEMA DE RECOMENDACIÓN DE MÚSICA, http://www.fing.edu.uy/inco/grupos/pln/prygrado/InformePGMusica.pdf, 2005-2006,11-40

[17] **SMILEY, D. Y PUGH, E.,** APACHE SOLR, http://www.google.com.ec/url?sa=t&rct=j&q=dando%20xml%20buscar%20con%20solr&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fseminario-apache-solr.googlecode.com%2Ffiles%2FSeminario%2520Apache%2520Solr.pptx&ei=wht3T9WpOoacgwfiyJX5Dg&usg=AFQjCNGZVOixygknIhq_yH65bsCs2T06ZQ&cad=rja, 2012, 4-8

[18] **ALDRICH JOHN,** MÁXIMA VEROSIMILITUD, http://es.wikipedia.org/wiki/M%C3%A1xima_verosimilitud, 2009, 162–176

[19] **OLIVARES-PACHECO JUAN,** ESTIMACIÓN DE MÁXIMA VEROSIMILITUD UTILIZANDO LA FUNCIÓN ÓPTIM EN R, http://www.mat.uda.cl/jolivares/probabilidades/EMV.pdf, 2006, 2

[20] **PORCEL GALLEGO CARLOS,** SISTEMAS DE ACCESO A LA INFORMACION BASADOS EN INFORMACION LINGÜÍSTICA DIFUSA Y TÉCNICAS DE FILTRADO, http://sinbad2.ujaen.es/cod/archivosPublicos/tesis/pdf/TesisCarlos.pdf, 2005

[21] **LEW DANIEL, BEN SOWELL, STEINBERG LEAH, TULADHAR AMRIT,** RECOMMENDER SYSTEMS,

http://www.cs.carleton.edu/cs_comps/0607/recommend/recommender/modelbased.html, 2012

[22] **SARWAR BADRUL, KARYPIS GEORGE, KONSTAN JOSEPH, RIEDL JOHN,** ITEM-BASED COLLABORATIVE FILTERING RECOMMENDATION ALGORITHMS, http://www10.org/cdrom/papers/519/sdm2.html, 2001

[23] **PALOMARES CARRASCOSA IVÁN,** RADIO ONLINE BASADA EN UN MOTOR DE FILTRADO COLABORATIVO, http://sinbad2.ujaen.es/cod/archivosPublicos/pfc/pfc_ivan_palomares.pdf, 2009