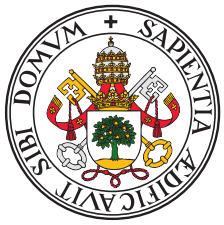


Universidad de Valladolid

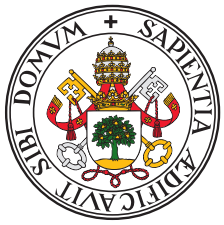
E.T.S Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática,
mención en Computación

Algoritmos para Big Data

Autor:
Sergio García Prado



Universidad de Valladolid

E.T.S Ingeniería Informática

Trabajo Fin de Grado

Grado en Ingeniería Informática,
mención en Computación

Algoritmos para Big Data

Autor:
Sergio García Prado

Tutor:
Manuel Barrio Solórzano

Prefacio

Para entender el contenido de este documento así como la metodología seguida para su elaboración, se han de tener en cuenta diversos factores, entre los que se encuentran el contexto académico en que ha sido redactado, así como el tecnológico y social. Es por ello que a continuación se expone una breve descripción acerca de los mismo, para tratar de facilitar la compresión sobre el alcance de este texto.

Lo primero que se debe tener en cuenta es el contexto académico en que se ha llevado a cabo. Este documento se ha redactado para la asignatura de **Trabajo de Fin de Grado (mención en Computación)** para el *Grado de Ingeniería Informática*, impartido en la *E.T.S de Ingeniería Informática* de la *Universidad de Valladolid*. Dicha asignatura se caracteriza por ser necesaria la superación del resto de las asignaturas que componen los estudios del grado para su evaluacion. Su carga de trabajo es de **12 créditos ECTS**, cuyo equivalente temporal es de *300 horas* de trabajo del alumno, que se han llevado a cabo en un periodo de 4 meses.

La temática escogida para realizar dicho trabajo es **Algoritmos para Big Data**. El Big Data es la disciplina que se encarga de "todas las actividades relacionadas con los sistemas que manipulan grandes conjuntos de datos. Las dificultades más habituales vinculadas a la gestión de estas cantidades de datos se centran en la recolección y el almacenamiento, búsqueda, compartición, análisis, y visualización. La tendencia a manipular enormes cantidades de datos se debe a la necesidad en muchos casos de incluir dicha información para la creación de informes estadísticos y modelos predictivos utilizados en diversas materias."[**?**]

Uno de los puntos más importantes para entender la motivación por la cual se ha escogido dicha temática es el contexto tecnológico en que nos encontramos. Debido a la importante evolución que están sufriendo otras disciplinas dentro del mundo de la informática y las nuevas tecnologías, cada vez es más sencillo y económico recoger gran cantidad de información de cualquier proceso que se dé en la vida real. Esto se debe a una gran cantidad de factores, entre los que se destacan los siguientes:

- **Reducción de costes derivados de la recolección de información**: Debido a la constante evolución tecnológica cada vez es más barato disponer de mecanismos (tanto a nivel de hardware como de software), a partir de los cuales se puede recabar datos sobre un determinado suceso.

- **Mayor capacidad de cómputo y almacenamiento**: La recolección y manipulación de grandes cantidades de datos que se recogen a partir de sensores u otros métodos requieren por tanto del apoyo de altas capacidades de cómputo y almacenamiento. Las tendencias actuales se están apoyando en técnicas de virtualización que permiten gestionar sistemas de gran tamaño ubicados en distintas zonas geográficas como una unidad, lo cual proporciona grandes ventajas en cuanto a reducción de complejidad algorítmica a nivel de aplicación.

- **Mejora de las telecomunicaciones**: Uno de los factores que ha permitido una gran disminución de la problemática relacionada con la virtualización y su capacidad de respuesta ha sido el gran avance a nivel global que han sufrido las telecomunicaciones en los últimos años, permitiendo disminuir las barreras geográficas entre sistemas tecnológicos dispersos.

Gracias a este conjunto de mejoras se ha llegado al punto en que existe la oporturnidad de poder utilizar una gran cantidad de conocimiento, que individualmente o sin un apropiado procesamiento, carece de valor a nivel de información.

El tercer factor que es necesario tener en cuenta es la tendencia social actual, que cada vez más, está concienciada con el valor que tiene la información. Esto se ve reflejado en un amplio abanico de aspectos relacionados con el comportamiento de la población:

- **Monitorización de procesos laborales**: Muchas empresas están teniendo en cuenta la mejora de la productividad de sus empleados y máquinas. Por tanto, buscan nuevas técnicas que les permitan llevar a cabo dicha tarea. En los últimos años se ha dedicado mucho esfuerzo en implementar sistemas de monitorización que permitan obtener información para después procesarla y obtener resultados valiosos para dichas organizaciones.
- **Crecimiento exponencial de las plataformas de redes sociales**: La inherente naturaleza social del ser humano hace necesaria la expresión pública de sus sentimientos y acciones, lo cual, en el mundo de la tecnología se ha visto reflejado en un gran crecimiento de las plataformas de compartición de información así como de las de comunicación.
- **Iniciativas de datos abiertos por parte de las administraciones públicas**: Muchas insitituciones públicas están dedicando grandes esfuerzos en hacer visible la información que poseen, lo que conlleva una mejora social aumentando el grado de transparencia de las mismas, así como el nivel de conocimiento colectivo, que puede ser beneficioso tampo para ciudadanos como para empresas.

Como consecuencia de este cambio social, posiblemente propiciado por el avance tecnológico anteriormente citado, la población tiene un mayor grado de curiosidad por aspectos que antes no tenia la capacidad de entender, debido al nivel de complejidad derivado del tamaño de los conjuntos de muestra necesarios para obtener resultados fiables.

En este documento no se pretenden abordar temas relacionados con las técnicas utilizadas para recabar nuevos datos a partir de los ya existentes. A pesar de ello se realizará una breve introducción sobre dicho conjunto de estrategias, entre las que se encuentran: *Heurísticas*, *Regresión Lineal*, *Árboles de decisión*, *Máquinas de Vector Soporte (SVM)* o *Redes Neuronales Artificiales*.

Por contra, se pretende realizar un análisis acerca de los diferentes algoritmos necesarios para manejar dichas cantidades ingentes de información, en especial de su manipulación a nivel de operaciones básicas, como operaciones aritméticas, búsqueda o tratamiento de campos ausentes. Para ello, se tratará de acometer dicha problemática teniendo en cuenta estrategias de paralelización, que permitan aprovechar en mayor medida las capacidades de cómputo existentes en la actualidad.

Otro de los aspectos importantes en que se quiere orientar este trabajo es el factor dinámico necesario para entender la información, lo cual conlleva la búsqueda de nuevas estrategias algorítmicas de procesamiento en tiempo real. Por lo tanto, se pretende ilustrar un análisis acerca de las soluciones existentes en cada caso con respecto a la solución estática indicando las ventajas e inconvenientes de la versión dinámica según corresponda.

Índice general

Capítulo 1

Introducción

[TODO]

Capítulo 2

Algoritmos para Streaming

[TODO]

Capítulo 3

Estructuras de Datos de Resumen

3.1. Introducción

El gran crecimiento tecnológico que se está llevando a cabo en la actualidad a todos los niveles está propiciando además un aumento exponencial en cuanto a la cantidad de información que se genera. La reducción de costes en cuanto a la instalación de sensores que permiten recoger información de muchos procesos productivos, así como la obtenición de metadatos a partir del uso de internet y las redes sociales por parte de los usuarios hace que el ritmo de crecimiento en cuanto a información generada por unidad de tiempo haya crecido a un gran ritmo.

Una de las razones que han facilitado dicha tendencia es la disminución de costes de almacenamiento de información a la vez que las capacidades de cómputo necesarias para procesar dicha información han aumentado. Sin embargo, debido al crecimiento exponencial en cuanto al tamaño del conjunto de datos, es necesario investigar nuevas técnicas y estrategias que permitan obtener respuestas satisfactorias basadas en la gran cantidad de información de la que se dispone en un tiempo razonable.

Tradicionalmente, la investigación en el campo de las *bases de datos* se ha centrado en obtener respuestas exactas a distintas consultas, tratando de hacerlo de la manera más eficiente posible, así como de tratar de reducir el espacio necesario para almacenar la información. *Acharya y otros* proponen en el artículo *Join synopses for approximate query answering* [**?**] el concepto de *Approximate Query Processing*. Dicha idea se expone en la subsección 3.1.1.

3.1.1. Approximate Query Processing

El *procesamiento aproximado de consultas*, (*Approximate Query Processing* o **AQP**) se presenta como una estrategia de consulta basada en conceptos y propiedades estadísticas que permiten una gran reducción de la complejidad computacional y espacial necesaria para la resolución de consultas a una base de datos. Por contra, dicha reducción a nivel de complejidad tiene como consecuencia la inserción de un determinado nivel de imprecisión en el resultado a la cual denominaremos tasa de error. Se pretende que dicha tasa de error pueda ser acotada en una desviación máxima determinada por ϵ y se cumpla con un índice de probabilidad δ. Al igual que en capítulos anteriores, en este caso también se presta especial importancia en la minimización del error relativo lo cual consigue que las soluciones mediante el *procesamiento aproximado de consultas* sean válidas tanto para consultas de tamaño reducido como de gran tamaño.

Durante el resto del capítulo se describen y analizan distintas estrategias que permiten llevar a cabo implementaciones basadas en *procesamiento aproximado de consultas* centrando especial atención en los *Sketches* por su similitud con el *Modelo en Streaming* descrito en el capítulo 2. En la sección 3.2 se realiza una decripción a partir de la cual se pretende aclarar las diferencias entre las distintas *estructuras de datos de resumen*. Posteriormente, en la sección **??** se

explican en detalle las cualidades de las estrategias basadas en *Sketching*. En las secciones 3.4, 3.5, 3.6 y 3.7 se habla de *Count-Min Sketch*, *Count Sketch*, *AMS Sketch* e *HyperLogLog* respectivamente.

3.2. Tipos de Estructuras de Datos de Resumen

Para el diseño de soluciones basadas en *procesamiento aproximado de consultas* en bases de datos existen distintas estrategias, las cuales presentan distintas ventajas e inconvenientes tal y como se pretende mostrar en esta sección. Dichas descripciones han sido extraidas del libro *Synopses for massive data* [**?**] redactado por *Cormode y otros*. En las secciones 3.2.1, 3.2.2, 3.2.3 y 3.2.4 se habla de *Sampling*, *Histogram*, *Wavelet* y *Sketches* respectivamente.

3.2.1. Sampling

El *Sampling* o *muestreo* es la estrategia más consolidada entre las que se presentan. Las razones se deben a su simplicidad conceptual así como su extendido uso en el mundo de la estadística. Uno de los primeros artículos en que se trata el muestreo aplicado a bases de datos es *Accurate estimation of the number of tuples satisfying a condition* [**?**] redactado por *Piatetsky-Shapiro* y *Connell*. La intuición en que se basa dicha estrategia es la selección de un subconjunto de elementos denominado *muestra* de entre el conjunto global al cual se denomina *población*. Una vez obtenida la *muestra* del conjunto de datos global cuyo tamaño es significativamente menor (lo cual reduce drásticamente el coste computacional), se realizan los cálculos que se pretendía realizar sobre toda la *población*, a partir de los cuales se obtiene un estimador del valor real que habría sido obtenido al realizarlos sobre el conjunto de datos global.

Para que las estrategias de sumarización de información obtengan resultados válidos o significativos respecto del conjunto de datos, es necesario que se escojan adecuadamente las instancias de la *muestra*, de manera que represente de manera fiel la información global. Para llevar a cabo dicha labor existen distintas estrategias, desde las más simples basadas en la selección aleatoria sin reemplazamiento como otras mucho más sofisticadas basadas en el mantenimiento de *muestras* estratificadas. Sea R la población y $|R|$ el tamaño de la misma. Denominaremos t_j al valor j-ésimo de la población y X_j al número de ocurrencias del mismo en la *muestra*. A continuación se describen distintas técnicas de muestreo:

- **Selección Aleatoria Sin Reemplazamiento**: Consiste en la estrategia más simple de generación de *muestras*. Se basa en la selección aleatoria de un valor entero r en el rango $[1, |R|]$ para después añadir el elemento localizado en la posición r de la *población* al subconjunto de *muestra*. Después repetir dicha secuencia durante n veces para generar una *muestra* de tamaño n. El estimador para la operación *SUMA* se muestra en la ecuación (3.1) además de la desviación de dicho estimador en la ecuación (3.2).

$$Y = \frac{|R|}{n} \sum_j X_j t_j \tag{3.1}$$

$$\sigma^2(Y) = \frac{|R|^2 \sigma^2(R)}{n} \tag{3.2}$$

- **Selección Aleatoria Con Reemplazamiento**: En este caso se supone que la selección de una instancia de la población tan solo se puede llevar a cabo una única vez como mucho, por lo tanto se cumple que $\forall X_j \in 0, 1$. La selección se lleva a cabo de la siguiente manera: se genera de manera aleatoria un valor entero r en el rango $[1, |R|]$ para después añadir el elemento localizado en la posición r de la *población* al subconjunto de *muestra* si este no ha sido añadido ya, sino volver a generar otro valor r. Después repetir dicha secuencia durante n veces para generar una *muestra* de tamaño n. Al igual que en la estrategia anterior, en este caso también se muestra el estimador para la operación *SUMA* en la ecuación (3.3). Nótese que el cálculo es el mismo que en el caso de

la estrategia sin reemplazamiento. Sin embargo, la varianza obtenida a partir de dicha estrategia es menor tal y como se muestra en la ecuación (3.4).

$$Y = \frac{|R|}{n} \sum_j X_j t_j \tag{3.3}$$

$$\sigma^2(Y) = \frac{|R|(|R| - n)\sigma^2(R)}{n} \tag{3.4}$$

- **Bernoulli y Poisson**: Mediante esta alternativa de muestreo se sigue una estrategia completamente distinta a las anteriores. En lugar de seleccionar la siguiente instancia aleatoriamente de entre todas las posibles, se decide generar $|R|$ valores aleatorios r_j independientes en el intervalo $[0, 1]$ de tal manera que si r_j es menor que un valor p_j fijado a priori, la instancia se añade al conjunto de *muestra*. Cuando se cumple que $\forall i, j \ p_i = p_j$ se dice que es un muestreo de *Bernoulli*, mientras que cuando no se cumple dicha condición se habla de muestreo de *Poisson*. El cálculo de la *SUMA* en este caso es muy diferente de los anteriores tal y como se muestra en la ecuación (3.5). La desviación de este estimador se muestra en la ecuación (3.6), que en general presenta peores resultados (mayor desviación) que las anteriores alternativas, sin embargo, esta alternativa posee la cualidad de aplicar distintos pesos a cada instancia de la población, lo que puede traducirse en que una selección adecuada de dichos valores p_j puede mejorar significativamente dichos resultados.

$$Y = \sum_{i \in muestra} \frac{t_i}{p_i} \tag{3.5}$$

$$\sigma^2(Y) = \sum_i (\frac{1}{p_i} - 1) t_i^2 \tag{3.6}$$

- **Muestreo Estratificado**: El muestreo estratificado trata de minimizar al máximo las diferencias entre la distribución del conjunto de datos de la *población* de la *muestra* que se pretende generar. Para ello existen distintas alternativas entre las que se encuentra una selección que actualiza los pesos p_j tras cada iteracción, lo que reduce la desviación de la *muestra*, sin embargo produce un elevado coste computacional en su generación. Por lo tanto se proponen otras estrategia más intuitiva basada en la partición del conjunto de datos de la *población* en subconjuntos disjuntos con varianza mínima entre las instancias que contienen a los cuales se denomina *estratos*. Posteriormente se selecciona mediante cualquiera de los métodos anteriores una *muestra* de cada *estrato*, lo cual reduce en gran medida la desviación típica global del estimador.

La estrategia de sumarización de información mediante *muestreo* tiene como ventajas la independencia de la complejidad con respecto a la dimensionalidad de los datos (algo que como se verá a continuación no sucede con otras alternativas) además de su simplicidad conceptual. También existen cotas de error para las consultas, para las cuales no ofrece restricciones en cuanto al tipo (debido a que se realizan sobre un subconjunto con la misma estructura que el global). El muestre es apropiado para conocer información general acerca del conjunto de datos que cada instancia del mismo posee. Además, presenta la cualidad de permitir la modificación en tiempo real, es decir, se pueden añadir o eliminar nuevas instancias a la muestra conforme se añaden o eliminan del conjunto de datos global.

Sin embargo, en entornos donde el ratio de addiciones/eliminaciones es muy elevado el coste del mantenimiento de la muestra puede hacerse impracticable . El *muestreo* es una buena alternativa para conjuntos de datos homogéneos en los cuales la presencia de valores atípicos es irrelevante. Tampoco obtiene buenos resultados en consultas relacionadas con el conteo de elementos distintos. En las siguientes secciones se describen alternativas que resuelven estas dificultades y limitaciones.

3.2.2. Histogram

Los *histogramas* son estructuras de datos utilizadas para sumarizar grandes conjuntos de datos, pero por contra, tienen un enfoque completamente diferente al que siguen las estrategias de *muestreo* de la sección anterior. En este caso, el concepto es similar a la visión estadística de los histogramas. Consiste en dividir el dominio de valores que pueden tomar las instancias del conjunto de datos de tal manera que se mantiene un conteo del número de instancias pertenecientes a cada partición.

Durante el resto de la sección se describen de manera resumida distintas estrategias de estimación del valor de las particiones así como las distintas estrategias de particionamiento del conjunto de datos. Para llevar a cabo dicha labor es necesario describir la notación que se seguirá: Sea D el conjunto de datos e $i \in [1, M]$ cada una de las categorías de los mismos. Denotaremos por $g(i)$ el número de ocurrencias del valor i. Para referirnos a cada uno de las particiones utilizaremos la notación S_j para $j \in [1, B]$. Nótese por tanto que M representa el número de categorías distintas mientras que B es el número de particiones utilizadas para "comprimir" los datos. La mejora de eficiencia en cuanto a espacio se consigue devido a la presuposición de que $B \ll M$

Cuando se habla de *esquemas de estimación* se trata de describir la manera en que se almacena o trata el contenido de cada una de las particiones S_j del histograma. La razón por la cual este es un factor importante a la hora de caracterizar un histograma es debida a que está altamente ligada a la precisión del mismo.

- **Esquema Uniforme**: Los esquemas que presuponen distribución uniforme se subdividen en dos categorías: *a*) *continous-value asumption* que presupone que todas las categorías i contenidas en la partición S_j presentan el mismo valor para la función $g(i)$ y *b*) *uniform-spread asumption* que presupone que el número de ocurrencias de la partición S_j se localiza distribuido uniformemente al igual que en el caso anterior, pero en este caso entre los elementos de un subconjunto P_j generado iterando con un determinado desplazamiento k sobre las categorías i contenidas en S_j. El segundo enfoque presenta mejores resultados en el caso de consultas de cuantiles que se distribuyen sobre más de una partición S_j

- **Esquema Basado en Splines**: En la estrategia basada en splines se presupone que los valores se distribuyen conforme una determinada función lineal de la forma $y_j = a_j x_j + b_j$ en cada partición S_j de tal manera que el conjunto total de datos D puede verse como una función continua a trozos. Nótese que en este caso se habla de una función lineal, sin embargo puede generalizarse a funciones no lineales.

- **Esquema Basado en Árboles**: Consiste en el almacenamiento de las frecuencias de cada partición S_j en forma de árbol binario, lo cual permite seleccionar de manera apropiada el nivel del árbol que reduzca el número de operaciones necesarias para obtener la estimación del número de ocurrencias según el tamaño rango de la consulta. La razón por la cual se escoje un árbol binario es debida a que se puede reducir en un orden de 2 el espacio necesario para almacenar dichos valores manteniendo únicamente los de una de las ramas del mismo. La razón de ello es debida a que se puede calcular el valor de la otra mediante una resta sobre el valor almacenado en el nodo padre y la rama que si contiene el valor.

- **Esquema Heterogéneo**: El esquema heterogéneo se basa la intuición de que la distribución de frecuencias de cada una de las particiones S_j no es uniforme, por lo tanto sigue un enfoque diferente en cada una de ellas tratanto de minimizar al máximo la tasa de error producida. Para ello existen distintas heurísticas basadas en distancias o teoría de la información entre otros.

Una vez descritas distintas estrategias de estimación del valor de frecuencias de una determinada partición S_j, el siguiente paso para describir un *histograma* es realizar una descripción acerca de las distintas formas de generación de dichas particiones. Para tratar de ajustarse de manera más adecuada a la distribución de los datos, se realiza un *muestreo* con el cual se generan las particiones. A continuación se describen las técnicas más comunes para dicha labor:

- **Particionamiento Heurístico**: Dichas estrategias de particionamiento se basan en distintas heurísticas que en la práctica han demostrado comportamientos aceptables en cuanto a resultados a nivel de precisión, sin embargo, no proporcionan ninguna garantía a nivel de optimalidad. Su uso está ampliamente extendido debido al reducido coste computacional. Dentro de esta categoría las heurísticas más populares son las siguientes:

  - **Equi-Width**: Consiste en la división del dominio de categorías $[1, M]$ en particiones equi-espaciadas unas de otras. Para dicha estrategia tan solo es necesario conocer *a-priori* el rango del posible conjunto de valores. Es la solución con menor coste computacional, a pesar de ello sus resultados a nivel práctico son similares a otras estrategias más sofisticadas cuando la distribución de frecuencias es uniforme.
  - **Equi-Depth**: Esta estrategia de particionamiento requiere conocer la distribución de frecuencias *a-priori* (o aproximarla a partir de algún método de muestreo). Se basa en la división del dominio de valores de tal manera que las particiones tengan la misma frecuencia. Para ello se crean particiones de tamaños diferentes.
  - **Singleton-Bucket**: Para tratar de mejorar la precisión esta estrategia de particionamiento se basa en la utilización de dos particiones especiales, las cuales contienen las categorías de mayor y menor frecuencia respectivamente para después cubrir el resto de categorías restante mediante otra estrategia (generalmente *equi-depth*).
  - **Maxdiff**: En este caso, el método de particionamiento se apoya en la idea de utilizar los puntos de mayor variación de frecuencias mediante la medida $|g(i+1) - g(i)|$, para dividir el conjunto de categorías en sus respectivas particiones, de tal manera que las frecuencias contenidas en cada partición sean lo más homogéneas posibles.

- **Particionamiento con Garantías de Optimalidad**: En esta categoría se enmarcan las estrategias de generación de particiones que ofrecen garantías de optimalidad a nivel de la precisión de resultados en las consultas. Para ello se apoyan en técnicas de *Programación Dinámica* (DP), de tal manera que la selección de las particiones se presenta como un problema de *Optimización*. Sin embargo, dichas estrategias presentan un elevado coste computacional que muchas veces no es admisible debido al gran tamaño del conjunto de datos que se pretende sumarizar. Como solución ante dicha problemática se han propuesto distintas estrategias que se basan en la resolución del problema de optimización, pero sobre una *muestra* del conjunto de datos global, lo cual anula las garantías de optimalidad pero si se escoge de manera adecuada ofrece una buena aproximación hacia ellas.

- **Particionamiento Jerárquico**: Las estrategias de particionamiento jerárquico se basan en la utilización de particiones siguiendo la idea de un árbol binario. Por lo tanto, dichas particiones no son disjuntas entre ellas, sino que se contienen unas a otras. Esto sigue la misma idea que se describió en el apartado de *Esquemas de estimación Basados en Árboles*. Apoyandose en este estilo de particionamiento se consigue que las consultas de rangos de frecuencias tengan un coste computacional menor en promedio (aún en el casos en que el rango sea muy amplio). En esta categoría destacan los histogramas *nLT* (n-level Tree) y *Lattice Histograms*. Estos últimos tratan de aprovechar las ventajas a nivel de flexibilidad y precisión que presentan los histogramas, además de las estrategias jerárquicas de sumarización en que se apoyan las *Wavelets* tal y como se describe en la siguiente sección.

Las ideas descritas en esta sección sobre los *histogramas* son extrapolables conforme se incrementa la dimensionalidad de los datos, en el caso de los esquemas de estimación, esto sucede de manera directa. Sin embargo, en el caso de los esquemas de particionamiento surgen distintos problemas debido al crecimiento exponencial tanto del espacio como del tiempo conforme aumenta el número de dimensiones de los datos.

Los *Histogramas* representan una estrategia sencilla, tanto a nivel de construcción como de consulta, la cual ofrece buenos resultados en un gran número de casos. Dichas estructuras han sido ampliamente probadas para aproximación de consultas relacionadas con suma de rangos o frecuencias puntuales. Tal y como se ha dicho previamente, su comportamiento en el caso unidimensional ha sido ampliamente estudiado, sin embargo, debido al crecimiento exponencial a nivel de complejidad conforme las dimensiones del conjunto de datos aumentan, estas estrategias son descartadas en

muchas ocasiones. Los *Histogramas* requieren de un conjunto de parámetros fijados *a-priori*, los cuales afectan en gran medida al grado de precisión de los mismos (pero cuando se seleccionan de manera adecuada esta solución goza de una gran cercanía al punto de optimalidad), por tanto, en casos en que la estimación de dichos valores necesarios *a-priori* se convierte en una labor complicada existen otras técnicas que ofrecen mejores resultados.

3.2.3. Wavelet

Las estructuras de sumarización denominadas *Wavelets*, a diferencia de las descritas anteriormente, han comenzado a utilizarse en el campo del *procesamiento aproximado de consultas* desde hace relativamente poco tiempo, por lo que su uso no está completamente asentado en soluciones comerciales sino que todavía están en fase de descubrimiento e investigación. Las *Wavelets* (u *ondículas*) se apoyan en la idea de representar la tabla de frecuencias del conjunto de datos como una función de ondas discreta. Para ello se almacenan distintos valores (dependiendo del tipo de *Wavelet*) que permiten reconstruir la tabla de frecuencias. Tal y como se describirá a continuación cuando se describa la *transformada de Haar*, la mejora de eficiencia en cuanto a espacio a partir de esta estructura de sumarización se apoya en el mantenimiento aproximado de los valores que representan el conjunto de datos.

A continuación se describe la *transformada de Haar*, a partir de la cual se presentan las distintas ideas en que se apoyan este tipo de estructuras de sumarización. En los últimos años se ha trabajado en estrategias más complejas como la *Daubechies Wavelet* [**?**] de *Akansu y otros* o la *transformada de Wavelet basada en árboles duales completos* [**?**] de *Selesnick y otros*.

Haar Wavelet Transform

La *Haar Wavelet Transform* (**HWT**) consiste en una construcción de estructura jerárquica que colapsa las frecuencias de las distintas categorías de manera pareada recursivamente hasta llegar a un único elemento. Por tanto, la idea es la similar a la creación de un árbol binario desde las hojas hasta la raiz. Esta estrategia es similar a la que siguen los *Histogramas jerárquicos* de la sección anterior. Además, se aprovecha una idea similar al caso anterior para optimizar el espacio, consistente en almacenar la variación de uno de los nodos hoja con respecto del padre, lo cual permite reconstruir el árbol completo mediante una simple operación.

Para simplificar el entendimiento de la construcción de la *transformada de Haar* se describe un ejemplo extraido del libro *Synopses for massive data* [**?**] de *Cormode y otros*. Supongamos los valores de frecuencias recogidos en $A = [2, 2, 0, 2, 3, 5, 4, 4]$. Para construir la transformada realizaremos la media de elementos dos a dos de manera recursiva, de tal manera que obtenemos para los distintos niveles los resultados de la tabla 3.1. Además, se muestran los coeficientes de detalle, los cuales se obtienen de calcular la diferencia entre el primer y segundo elemento de cada media.

Nivel	Medias	Coeficientes de Detalle
3	$[2, 2, 0, 2, 3, 5, 4, 4]$	$-$
2	$[2, 1, 4, 4]$	$[0, -1, -1, 0]$
1	$[3/2, 4]$	$[1/2, 0]$
0	$[11/4]$	$[-5/4]$

Tabla 3.1: Ejemplo de construcción de *Haar Wavelet Transform*

Nótese que a partir de la media de nivel 0 a a cual denominaremos $c_0 = 11/4$ así como el conjunto de coeficientes de detalle, que denotaremos por $c_1 = -5/4, c_2 = 1/2, ..., c_7 = 0$ es posible reconstruir la tabla de frecuencias A.

Una vez entendida la estrategia de construcción en que se apoya la *transformada de Haar*, se puede apreciar que esta no ofrece ventajas a nivel de coste de almacenamiento respecto del conjunto de frecuencias con el cual ha sido construida. Sin embargo, posee la siguiente cualidad, en la cual se apoya esta estrategia de sumarización: *Para las categorías contiguas en que la variación de frecuencias es muy reducida, los coeficientes de detalle tienden a aproximarse a* 0.

Por la razón descrita en el parrafo anterior, se intuye que dichos coeficientes de detalle pueden ser obviados, de tal manera que el espacio utilizado para el almacenamiento de la *Wavelet* se convierte en sublineal ($o(N)$), en lugar de lineal ($O(N)$) respecto del espacio del conjunto de datos. Para elegir qué coeficientes de detalle se utilizan estrategias que tratan de minimizar el error. Comúnmente, las *Wavelets* han sido construidas a partir del *error quadrático medio* o *norma-L_2*, la cual se describe en la ecuación (3.7). Sin embargo, distintos estudios como el descrito en el artículo *Probabilistic wavelet synopses* [**?**] de *Garofalakis y otros* muestran como esta obtienen medida del error obtiene malos resultados en el caso de sumarización de datos mediante *Wavelets*.

Por tanto, se proponen otras medidas de error como la minimización del máximo error absoluto o relativo, que se describen en las ecuaciones (3.8) y (3.9). También se propone como alternativa la minimización de la *norma-L_p* que se describe en la ecuación (3.10). Dicha medida de error es una generalización del *error cuadrático medio* (caso $p = 2$) a cualquier valor de $p \geq 0$. Por último se muestra en la ecuación (3.11) el caso del cálculo del error mediante la *norma-L_p* con pesos, lo cual permite añadir mayor o menor importancia una determinada categoría sobre otras en la representación mediante *Wavelets*.

$$||A - \widetilde{A}||_2 = \sqrt{\sum_i (A[i] - \widetilde{A}[i])^2} \tag{3.7}$$

$$max_i\{absErr_i\} = max_i\{|A[i] - \widetilde{A}[i]|\} \tag{3.8}$$

$$max_i\{relErr_i\} = max_i\left\{\frac{|A[i] - \widetilde{A}[i]|}{|A[i]|}\right\} \tag{3.9}$$

$$||A - \widetilde{A}||_p = (\sum_i (A[i] - \widetilde{A}[i])^p)^{\frac{1}{p}} \tag{3.10}$$

$$||A - \widetilde{A}||_{p,w} = (\sum_i w_i \cdot (A[i] - \widetilde{A}[i])^p)^{\frac{1}{p}} \tag{3.11}$$

Al igual que en el caso de los *Histogramas*, las *Wavelets* presentan problemas de eficiencia cuando se usa en entornos en los cuales el conjunto de datos está compuesto por una gran número de atributos. Por lo tanto se dice que sufren la *Maldición de la dimensionalidad* (*Curse of Dimensionality*) que provoca un crecimiento en el coste de orden exponencial tanto en espacio como tiempo respecto del número de dimensiones.

Tal y como se puede apreciar, esta estrategia es muy similar a la basada en *Histogramas*, dado que ambas se basan en el almacenamiento de valores que tratan de describir o resumir la tabla de frecuencias de los datos de manera similar. Sin embargo, mientras que en el caso de los *Histogramas* estos destacan cuando se pretende conocer la estructura general de los datos, las *Wavelets* ofrecen muy buenos resultados cuando se pretenden conocer valores atípico o extremos (a los cuales se denomina *Heavy Hitters*).

Por su estrategia de construcción, las *Wavelets* permiten sumarizar una mayor cantidad de información utilizando menos espacio. Además, en el caso de la *transformada de Haar*, que posee la característica de linealidad, se puede adaptar de manera sencilla al *modelo en Streaming*. Tal y como se ha dicho en el párrafo anterior, las desventajas de esta alternativa vienen derivadas en gran medida de los problemas relacionados con el incremento de la dimensionalidad de los datos.

3.2.4. Sketch

Los *Sketches* son una estrategia de sumarización de información enmarcada dentro del *Modelo en streaming* descrito en la sección **??**. Dicha estrategia se caracteriza por procesar cada dato de entrada de manera secuencial, frente a otras alternativas como *Sampling*. Los *Sketches lineales* se caracterizan por procesar los datos de entrada realizando una *transformación lineal* sobre cada elemento y una estructura encargada de sumarizar la información de los sucesos previos.

[TODO modelo streaming]

[TODO tipos de skethes]

- **Estimación de Frecuencias**: [TODO]
- **Elementos Distintos**: [TODO]

[TODO dimensionalidad]

[TODO ventajas e inconvenientes]

[TODO área de investigación para soluciones futuras]

3.3. Bloom Filter

[TODO] *Space/time trade-offs in hash coding with allowable errors* [**?**]

3.4. Count-Min Sketch

[TODO] *An improved data stream summary: the count-min sketch and its applications* [**?**]

3.5. Count Sketch

[TODO] *Finding frequent items in data streams* [**?**]

3.6. AMS Sketch

[TODO] *The space complexity of approximating the frequency moments* [**?**]

3.7. HyperLogLog

[TODO] *Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm* [**?**]

Capítulo 4

Algoritmos para Grafos

4.1. Introducción

[TODO]

4.2. Modelo en Semi-Streaming

[TODO]

Capítulo 5

Reducción de la Dimensionalidad

5.1. Introducción

[TODO]

5.2. Teorema de Johnson-Lindenstrauss

[TODO]

5.3. Busqueda de Vecinos más Cercanos

[TODO]

Capítulo 6

Técnicas de Minería de Datos

6.1. Introducción

[TODO]

6.2. Aprendizaje Supervisado y No Supervisado

[TODO]

6.3. Árboles de Decisión

[TODO]

6.4. Regresión Lineal

[TODO]

6.5. Redes Neuronales

[TODO]

6.6. Manifold Learning

[TODO]

Capítulo 7

Paralelización a Gran Escala

7.1. Introducción

[TODO]

7.2. Sistemas de Ficheros Distribuidos

[TODO]

7.3. Modelo de acceso a Memoria

[TODO]

7.4. Complejidad de la Comunicación

[TODO]

7.5. MapReduce

[TODO]

Apéndice A

Metodología de Trabajo

Apéndice B

¿Cómo ha sido generado este documento?

En este apéndice se describen tanto la estructura como las tecnologías utilizadas para redactar este documento. El estilo visual que se ha aplicado al documento se ha tratado de almoldar lo máximo posible a las especificaciones suministradas en la *guía docente* de la asignatura *Trabajo de Fin de Grado* [**?**].

Este documento ha sido redactado utilizando la herramienta de generación de documentos LaTeX[**?**], en concreto se ha utilizado la distribución para sistemas *OS X* denominada *MacTeX* [**?**] desarrollada por la organización *TEX User Group*. Mediante esta estrategia todas las labores de compilación y generación de documentos *PDF* (tal y como se especifica en la guía docente) se realizan de manera local. Se ha preferido esta alternativa frente a otras como la utilización de plataformas online de redacción de documentos LaTeX como *ShareLateX* [**?**] u *Overleaf* [**?**] por razones de flexibilidad permitiendo trabajar en lugares en que la conexión a internet no esté disponible. Sin embargo, dichos servicios ofrecen son una buena alternativa para redactar documentos sin tener que preocuparse por todos aquellos aspectos referidos con la instalación de la distribución u otros aspectos como un editor de texto. Además garantizan un alto grado de confiabilidad respecto de pérdidas inesperadas.

Junto con la distribución LaTeX se han utilizado una gran cantidad de paquetes que extienden y simplifican el proceso de redactar documentos. Sin embargo, debido al tamaño de la lista de paquetes, esta será obviada en este apartado, pero puede ser consultada visualizando el correspondiente fichero `thestyle.sty` del documento.

Puesto que la alternativa escogida ha sido la de generar el documento mediante herramientas locales es necesario utilizar un editor de texto así como un visualizador de resultados. En este caso se ha utilizado *Atom* [**?**], un editor de texto de propósito general que destaca sobre el resto por ser desarrollado mediante licencia de software libre (*MIT License*) y estar mantenido por una amplia comunidad de desarrolladores además de una extensa cantidad de paquetes con los cuales se puede extender su funcionalidad. En este caso, para adaptar el comportamiento de *Atom* a las necesidades de escritura de texto con latex se han utilizados los siguientes paquetes: *latex* [**?**], *language-latex* [**?**], *pdf-view* [**?**] encargados de añadir la capacidad de compilar ficheros latex, añadir la sintaxis y permitir visualizar los resultados respectivamente.

Puesto que el *Trabajo de Fin de Grado* se refiere a algo que requiere de un periodo de tiempo de elaboración largo, que además sufrirá una gran cantidad de cambios, se ha creido conveniente la utilización de una herramienta de control de versiones que permita realizar un seguimiento de los cambios de manera organizada. Para ello se ha utilizado la tecnología *Git* [**?**] desarrollada originalmente por *Linus Torvalds*. En este caso en lugar de confiar en el entorno local u otro servidor propio se ha preferido utilizar la plataforma *GitHub* [**?**], la cual ofrece un alto grado de confiabilidad respecto de posibles perdidas además de alojar un gran número de proyectos de software libre. A pesar de ofrecer licencias para estudiantes que permiten mantener el repositorio oculto al público, no se ha creido necesario en este caso, por lo cual se puede acceder al través de la siguiente url: `https://github.com/garciparedes/tfg-big-data-algorithms` [**?**]

```
.
+-- document
|   +-- bib
|   |   +-- article.bib
|   |   +-- ...
|   +-- img
|   |   +-- logo_uva.eps
|   |   +-- ...
|   +-- tex
|   |   +-- appendices
|   |   |   +-- how_it_was_build.tex
|   |   |   +-- ...
|   |   +-- chapters
|   |   |   +-- introduction.tex
|   |   |   +-- ...
|   |   +-- bibliography.tex
|   |   +-- ...
|   +-- document.tex
|   +-- ...
+-- notes
|   +-- readme.md
|   +-- ...
+-- summary
|   +-- summary.tex
|   +-- ...
+-- README.md
+-- ...
```

Figura B.1: Árbol de directorios del repositorio

Una vez descritas las distintas tecnologías y herramientas utilizadas para la elaboración de este trabajo, lo siguiente es hablar sobre la organización de ficheros. Todos los ficheros utilizados para este documento (obviando las referencias bibliográficas) han sido incluidos en el repositorio indicado anteriormente [**?**].

Para el documento, principal alojado en el directorio `/document/` se ha seguido una estructura modular, dividiendo los capítulos, apéndices y partes destacadas como portada, bibliografía o prefacio entre otros en distintos ficheros, lo cual permite un acceso sencillo a los mismos. Los apéndices y capítulos se han añadido en los subdirectorios separados. Para la labor de combinar el conjunto de ficheros en un único documento se ha utilizado el paquete *subfiles*. El fichero raiz a partir del cual se compila el documento es `document.tex`. La importación de los distintos paquetes así como la adaptación del estulo del documento a los requisitos impuestos se ha realizado en `thestyle.sty` mientras que el conjunto de variables necesarias como el nombre de los autores, del trabajo, etc. se han incluido en `thevars.sty`.

En cuanto al documento de resumen, en el cual se presenta una vista panorámica acerca de las distintas disciplinas de estudio relacionadas con el *Big Data* se ha preferido mantener un único fichero debido a la corta longitud del mismo. Este se encuentra en el directorio `/summary/`.

Por último se ha decidido añadir otro directorio denominado `/notes/` en el cual se han añadido distintas ideas de manera informal, así como enlaces a distintos cursos, árticulos y sitios web en que se ha basado la base bibliográfica del trabajo. En la figura B.1 se muestra la estructura del repositorio en forma de árbol.