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."[Wik17]

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

2.1. Introducción

En este capítulo se trata de realizar una descripción en profundidad acerca de los *Algoritmos en Streaming* desde una perspectiva tanto a teórica como práctica. Para ello se describirá el modelo de cómputo en que se enmarcan dichos algoritmos (Modelo en Streaming) en la sección 2.2 además de su estructura básica en la sección 2.3. El motivo de dicha descripción se debe a que los *Algoritmos para Streaming* presentan un conjunto de peculiaridades respecto de la gran mayoría de algoritmos utilizados comunmente. [TODO introducir el resto de secciones]

Para realizar una primera aproximación acerca de en qué consiste esta categoría algorítmica es necesario realizar una diferenciación entre distintos conceptos relacionados con ella, que pueden producir confusiones debido a su similitud o abusos previos del lenguaje. Por lo tanto, a continuación se describen conceptos relacionados con los *Algoritmos en Streaming* que permitirán introducir al lector en el contexto del problema. Además, se realiza una diferenciación acerca de los factores que se pretenden optimizar a partir de esta estrategia de diseño de algoritmos.

2.1.1. Computación en Tiempo Real

El primer concepto que se describe es **Computación en Tiempo Real**, que tal y cómo describen Shin y Ramanathan [SR94] se carácteriza por tres términos que se describen a continuación:

- **Tiempo***(time)*: En la disciplina de *Computación en Tiempo Real* el tiempo de ejecucción de una determinada tarea es especialmente crucial para garantizar el correcto desarrollo del cómputo, debido a que se asume un plazo de ejecucción permitido, a partir del cual la solución del problema deja de tener un valor óptimo. Shin y Ramanathan[SR94] diferencian entre tres categorías dentro de dicha restricción, a las cuales denominan *hard*, *firm* y *soft*, dependiendo del grado de relajación de la misma.

- **Confiabilidad***(correctness)*: Otro de los puntos cruciales en un sistema de *Cómputación en Tiempo Real* es la determinación de una unidad de medida o indicador acerca de las garantias de una determinada solución algorítmica para cumplir lo que promete de manera correcta en el tiempo esperado.

- **Entorno***(environment)*: El último factor que indican Shin y Ramanathan[SR94] para describir un sistema de *Computación en Tiempo Real* es el entorno del mismo, debido a que este condiciona el conjunto de tareas y la periodicidad en que se deben llevar a cabo. Debido a esta razón, realizan una diferenciación entre: *a*) tareas periódicas *periodic tasks* las cuales se realizan secuencialmente a partir de la finalización de una ventana de tiempo, y *b*) tareas no periódicas *periodic tasks* que se llevan a cabo debido al suceso de un determinado evento externo.

2.1.2. Problemas Dinámicos

Una vez completada la descripción acerca de lo que se puede definir como *Computación en Tiempo Real*, conviene realizar una descripción desde el punto de vista de la *teoría de complejidad computacional*. Para definir este tipo de problemas, se utiliza el término *problemas dinámicos*, los cuales consisten en aquellos en los cuales es necesario recalcular su solución conforme el tiempo avanza debido a variaciones en los parámetros de entrada del problema (Nótese que dicho término no debe confundirse con la estrategia de *programación dinámica* para el diseño de algoritmos). Existen distintas vertientes dependiendo del punto de vista desde el que se estudien, tanto de la naturaleza del problema (soluciones dependientes temporalmente unas de otras o soluciones aisladas) como de los parámetros de entrada (entrada completa en cada nueva ejecución o variación respecto de la anterior). Los *Algoritmos para Streaming* están diseñados para resolver *problemas dinámicos*, por lo que en la sección 2.2, se describe en profundidad el modelo en que se enmarcan.

A continuación se indican los principales indicadores utilizados para describir la complejidad de una determinada solución algorítmica destinada a resolver un problema de dicha naturaleza:

- Espacio: Cantidad de espacio utilizado en memoria durante la ejecución del algoritmo.
- Inicialización: Tiempo necesario para la inicialización del algoritmo.
- Procesado: Tiempo necesario para procesar una determinada entrada.
- Pregunta[TODO Buscar mejor palabra]: Tiempo necesario para procesar la solución a partir de los datos de entrada procesados hasta el momento.

2.1.3. Algoritmos Online vs Algoritmos Offline

Una vez descrita la problemática de *Computación en Tiempo Real* en la sección 2.1.1 y la categoría de *Problemas Dinámicos* en la sección 2.1.2, en esta sección se pretende ilustrar la diferencia entre los *Algoritmos Online* y los *Algoritmos Offline*. Para ello, se ha seguido la diferenciación propuesta por Karp [Kar92], en la cual se plantea el problema de la siguiente manera (Se utilizará la misma notación que sigue Muthukrishnan[Mut05] para tratar mantener la consistencia durante todo el documento): Sea A el conjunto de datos o eventos de entrada, siendo cada $A[i]$ el elemento *i-ésimo* del conjunto. En el caso de los *Algoritmos Online* supondremos que es el elemento recibido en el instante i. A continuación se muestran las características de cada subgrupo:

- **Algoritmos Offline**: Esta categoría contiene todos los algoritmos que realizan el cómputo suponiendo el acceso a cualquier elemento del conjunto de datos A durante cualquier momento de su ejecución. Además, en esta categoría se impore la restricción de que el A debe ser invariante respecto del tiempo, lo que impone que para la adaptación del resultado a cambios, este tenga que realizar una nueva ejecución desde su estado inicial. Nótese que por tanto, dentro de este grupo se engloba la mayoría de algoritmos utilizados comunmente.
- **Algoritmos Online**: Son aquellos que calculan el resultado a partir de una secuencia de sucesos $A[i]$, los cuales generan un resultado dependiente del valor, y posiblemente de los sucedidos anteriormente. A partir de dicha estrategia, se añade una componente dinámica, la cual permite que tamaño del conjunto de datos de entrada A no tenga impuesta una restricción acerca de su longitud. Por contra, en este modelo no se permite conocer el cuceso $A[i+1]$ en el momento i. Esto encaja perfectamente en el modelo que se describirá en la sección 2.2.

Según la diferenciación que se acaba de describir, estas dos estrategias de diseño de algoritmos encajan en disciplinas distintas de diseño de algoritmos, teniendo una gran ventaja a nivel de eficiencia en el caso estático los *Algoritmos Offline*, pero quedando prácticamente inutilizables cuando la computación es en tiempo real, donde es mucho más apropiado el uso de estrategias de diseño de *Algoritmos Online*.

Como medida de eficiencia para los *Algoritmos Online*, Karp [Kar92] propone el **Ratio Competitivo**, el cual se define como la cota inferior del coste de cualquier nueva entrada con respecto de la que tiene menor coste. Sin embargo, dicha medida de eficiencia no es comúnmente utilizada en el caso de los *Algoritmos para Streaming* por la componente estocástica de los mismos, para los cuales son más apropiadas medidas probabilistas. A continuación se describen las ventajas de estos respecto de su vertiente estática.

2.1.4. Algoritmos Probabilistas

Los *Algoritmos Probabilistas* son una estrategia de diseño que emplea en un cierto grado de aleatoriedad en alguna parte de su lógica. Estos utilizan distribuciones uniformes de probabilidad para tratar de conseguir un incremento del rendimiento en su caso promedio. A continuación se describen los dos tipos de algoritmos probabilísticos según la clasificación realizada por Babai [Bab79]:

- **Algoritmos Las Vegas**: Devuelven un resultado incorrecto con una determinada probabilidad, pero avisan del resultado incorrecto cuando esto sucede. Para contrarrestrar este suceso basta llevar a cabo una nueva ejecución del algoritmo, lo cual tras un número indeterminado de ejecuciones produce un resultado válido.
- **Algoritmos Monte Carlo**: Fallan con un cierto grado de probabilidad, pero en este caso no avisan del resultado incorrecto. Por lo tando, lo único que se puede obtener al respecto es una indicador de la estimación del resultado correcto hacia la que converge tras varias ejecuciones. Además, se asegura una determinada cota del error ϵ, que se cumple con probabilidad δ.

La razón anecdótica por la cual Babai [Bab79] decidió denominar dichas categorías de algoritmos de esta manera se debe a lo siguiente (teniendo en cuenta el contexto de lengua inglesa): cuando se va a un casino en *Las Vegas* y se realiza una apuesta el *croupier* puede decir si se ha ganado o perdido porque habla el mismo idioma. Sin embargo, si sucede la misma situación en *Monte Carlo*, tan solo se puede conocer una medida de probabilidad debido a que en este caso el *croupier* no puede comunicarlo por la diferencia dialéctica.

2.1.5. Algoritmos Online Probabilistas vs Deterministas

La idea subyacente acerca del diseño de los *Algoritmos Online* es la mejora de eficiencia con respecto a sus homónimos estáticos cuando el conjunto de valores de entrada es dependiente de los resultados anteriores. Sin embargo, existen casos en que la frecuencia de ejecución del algoritmo, debido a una alta tasa de llegada de sucesos de entrada, las soluciones deterministas se convierten en alternativas poco escalables.

Dicha problemática se ha incrementado de manera exponencial debido al avance tecnológico y la gran cantidad de información que se está generando en la actualidad. Este fenómeno ha convertido en algo necesario el diseño de estrategias basadas en sucesos probabilísticos que reduzcen en gran medida el coste computacional eliminando el determinismo de la solución.

2.2. Modelo en Streaming

En esta sección se describen los aspectos formales del *Modelo en Streaming*. Para ello se ha seguido la representación definida por Muthukrishnan [Mut05]. Lo primero por tanto, es definir lo que es un flujo de datos o *Data Stream* como una "secuencia de señales digitalmente codificadas utilizadas para representar una transmisión de información" [Ins17]. Muthukrishnan [Mut05] hace una aclaración sobre dicha definición y añade la objeción de que los datos de entrada deben tener un ritmo elevado de llegada. Debido a esta razón existe complejidad a tres niveles:

- **Transmisión**: Ya que debido a la alta tasa de llegada es necesario diseñar un sistema de interconexiones que permita que no se produzcan congestiones debido a la obtención de los datos de entrada.
- **Computación**: Puesto que la tarea de procesar la gran cantidad de información que llega por unidad de tiempo produce cuellos de botella en el planteamiento. Por lo que es necesario implementar técnicas algorítmicas con un reducido nivel de complejidad computacional que para contrarrestar dicha problemática.
- **Almacenamiento**: Debido a la gran cantidad de datos que se presentan en la entrada, deben existir técnicas que permitan almacenar dicha información de manera eficiente. Esto puede ser visto desde dos puntos de vista diferences: *a*) tanto desde el punto de vista del espacio, tratando de minimizar el tamaño de los datos almacenados, maximizando la cantidad de información que se puede recuperar de ellos, *b*) como desde el punto de vista del tiempo necesario para realizar operaciones de búsqueda, addición, eliminación o edición.

[TODO ejemplo de los tres niveles de complejidad a partir del análisis meteorológico]

Una vez descritos los niveles de complejidad a los que es necesario enfrentarse, en los apartados 2.2.1, 2.2.2 y 2.2.3 se realiza una descripción acerca de los distintos modelos que propone Muthukrishnan [Mut05]. La notación descrita en dichos apartados será seguida durante el resto del capítulo. Para ello nos basaremos en el siguiente formalismo:

Sea $a_1, a_2, ...$ el conjunto de posibles elementos de entrada con cardinalidad N (posiblemente no acotada tal y como se verá en el caso de las *Series Temporales* (2.2.1)), se debe cumplir que $\forall i, \forall j / i \neq j \implies a_i \neq a_j$. Se ha decidido introducir el símbolo "$\cdot$" como comodín siguiendo la misma interpretación que utilizan los sistemas *UNIX* en el contexto de las *Expresiones regulares*[GNU17], de tal manera que $a_\cdot$ indica que el elemento al que nos estamos refiriendo puede ser cualquier valor.

Lo siguiente es añadir la componente dinámica al formalismo. Por tanto, diremos que $a_\cdot[t]$ se refiere al elemento recibido en el instante de tiempo t. La restricción más relevante que se impone a un *Modelo en Streaming* es la siguiente: Sea $a_\cdot[1], a_\cdot[2], ..., a_\cdot[t], ...$ un flujo de entrada (*Input Stream*), de tal manera que cada elemento debe tener un orden de llegada secuencial respecto de t. Esto quiere decir que el elemento previo a la llegada de a_i debe ser a_{i-1} y por inducción, el siguiente será a_{i+1}.

2.2.1. Modelo de Serie Temporal

El *Modelo de Serie Temporal* o *Time Series Model*

2.2.2. Modelo de Caja Registradora

[TODO]

2.2.3. Modelo de Molinete

[TODO]

2.3. Estructura básica

[TODO]

2.4. Medidas de Análisis

[TODO]

Capítulo 3

Estructuras de Datos de Resumen

[TODO]

Capítulo 4

Algoritmos para Grafos

[TODO]

Capítulo 5

Técnicas de Minería de Datos

[TODO]

Capítulo 6

Paralelización a Gran Escala

[TODO]

Apéndice A

Metodología de Trabajo

Apéndice B

¿Cómo ha sido generado este documento?

Bibliografía

[Bab79] László Babai. Monte-carlo algorithms in graph isomorphism testing, 1979.

[GNU17] GNU, Grep. Regular expresions: Fundamental structure, March 2017. `https://www.gnu.org/software/grep/manual/html_node/Fundamental-Structure.html`.

[Ins17] Institute for Telecommunication Sciences. Definitions: Data stream, March 2017. `https://www.its.bldrdoc.gov/fs-1037/dir-010/_1451.htm`.

[Kar92] Richard M. Karp. On-line algorithms versus off-line algorithms: How much is it worth to know the future? In *Proceedings of the IFIP 12th World Computer Congress on Algorithms, Software, Architecture - Information Processing '92, Volume 1 - Volume I*, pages 416–429, Amsterdam, The Netherlands, The Netherlands, 1992. North-Holland Publishing Co.

[Mut05] S. Muthukrishnan. Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.*, 1(2):117–236, August 2005.

[SR94] K. G. Shin and P. Ramanathan. Real-time computing: a new discipline of computer science and engineering. *Proceedings of the IEEE*, 82(1):6–24, Jan 1994.

[Wik17] Wikipedia. Big Data, February 2017.