UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

REGLAS DE ASOCIACIÓN PARA LÍNEAS ESPECTRALES

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

NICOLÁS MARTÍN MIRANDA CASTILLO

PROFESOR GUÍA:
GUILLERMO CABRERA VIVES

MIEMBROS DE LA COMISIÓN:
GONZALO NAVARRO BADINO
PABLO GUERRERO PÉREZ

Este trabajo ha sido parcialmente financiado por el DÉCIMO NOVENO CONCURSO DE PROYECTOS DE INVESTIGACIÓN Y DESARROLLO FONDEF 2011, PROYECTO FONDEF D11I1060, y el CENTRO DE MODELAMIENTO MATEMÁTICO DE LA UNIVERSIDAD DE CHILE (CMM)

SANTIAGO DE CHILE
2015

Resumen

Parte importante de la labor astronómica consiste en analizar observaciones de radiaciones electromagnéticas en la forma de espectros de frecuencia, procedentes del espacio y emitidas por objetos tales como estrellas, galaxias y otros. A partir de estos espectros se puede identificar una serie de propiedades y características de los objetos de los cuales provienen; en particular, las líneas espectrales (tanto de emisión como de absorción) presentes resultan ser un indicador de las especies (átomos, moléculas, isótopos, etc.) presentes en su composición química.

En el presente trabajo se llevó a cabo con el fin de obtener un cierto tipo de asociaciones lógicas, llamadas *reglas de asociación*, entre líneas espectrales presentes a lo largo de distintos espectros de frecuencia. En particular, se busca aplicar a conjuntos de espectros de frecuencia obtenidos a partir de datos de observaciones astronómicas, para así obtener información de las relaciones existentes entre estas líneas bajo distintas medidas de interés y relevancia estadística.

Para ello se llevó a cabo, en el lenguaje de programación *Python*, una implementación de algoritmos de Aprendizaje de Reglas de asociación, o *Association Rule Learning (ARL)*; en particular los algoritmos *Apriori* y *FP-Growth*. La aplicación final, que hace uso de estos algoritmos, permite al usuario observar las reglas obtenidas bajo requerimientos mínimos de *soporte* y *confianza* de ellas, ordenarlas según estas dos medidas junto con su *lift*, y mostrar las que posean un cierto elemento en particular en su antecedente, consecuente o en ambos.

La aplicación y sus algoritmos se probaron sobre datos simulados y, posteriormente, sobre datos reales de observaciones en el espectro visible obtenidas del *Sloan Digital Sky Survey (SDSS)*, previo un pre-procesamiento adecuado de estos. Los resultados obtenidos muestran un considerable mejor desempeño (de por lo menos la mitad del tiempo total de ejecución) por parte del algoritmo *Apriori* por sobre *FP-Growth* para valores pequeños de soporte (cercanos a 0.15). Esto puede deberse, principalmente, al tamaño reducido del universo de ítems (lineas espectrales detectadas) posibles presentes en cada transacción (espectro de frecuencias); lo cual hace perder sustancialmente la ventaja comparativa que posee *FP-Growth* al utilizar una estructura de datos tipo árbol.

Se espera a futuro poder realizar el proceso de ARL a partir datos en otras frecuencias del espectro electromagnético; como por ejemplo, los datos radioastronómicos del *Atacama Large Millimeter/submillimeter Array (ALMA)*. Junto con esto, se espera más adelante poder mejorar la aplicación en términos de su interfaz gráfica y usabilidad.

A mi padre.

Agradecimientos

A propósito del trabajo que aquí se presenta, no quisiera dejar pasar esta ocasión sin agradecer a Guillermo Cabrera, mi guía a lo largo de este proyecto, por su paciencia, instrucción y excelente disposición a la hora de permitirme ver y aprender muchas cosas nuevas. Muchas gracias, también, al profesor Diego Mardones. Sin su asesoramiento en materias científicas que incluyen (pero no se reducen solo) a la astronomía, y su constante ayuda en general, este trabajo no habría sido posible.

En términos más personales, profundos y generales, a mi familia. A Liliana, mi madre, por el cariño sin medidas ni reservas que siempre me ha brindado. A Rocío, mi hermana, por su alegría contagiosa y optimismo, que en más de una ocasión me han sacado adelante. Y, por supuesto, a Sergio, mi padre, por su apoyo incondicional, por su ejemplo, compañía y enseñanzas invaluables sobre nunca darse por vencido, sin dejar de disfrutar del día a día. No estaría aquí de no ser por ustedes.

A mis amigos de siempre y de ahora, en recuerdo pero sobre todo en presencia. Gracias por haber compartido conmigo tantos buenos momentos, risas, ideas, conversaciones, y por estar aun ahí para mí, a pesar de lo divergentes que son a veces los senderos de la vida.

A todos los creadores, escritores, profesores, artistas, personas comunes y anónimas que, mediante sus obras y ejemplos, me han enseñado el valor de pensar por uno mismo, ser altruísta, ver más allá de lo evidente, sorprenderse con la realidad, imaginar sin temores y apreciar el mundo del que todos somos parte.

A todos ustedes, muchas gracias.

Tabla de contenido

Indice de ilustraciones

Introducción

[...] we may in time ascertain the mean temperature of heavenly bodies, but I regard this order of facts as for ever excluded from our recognition. We can never learn their internal constitution [...]

Auguste Comte, *Astronomy, Ch. I: General View*, 1835

Contexto y motivación

En los últimos tiempos, y en gran parte debido al explosivo desarrollo tecnológico, han surgido numerosos campos en los cuales se ha requerido el uso de procesamiento masivo de datos e inteligencia computacional con el fin de automatizar y auxiliar el proceso de generación de nuevo conocimiento. La astronomía es, sin lugar a dudas, uno de ellos. Esto se debe, en parte, al explosivo desarrollo de nuevas tecnologías que ponen al alcance de la comunidad científica una cantidad nunca antes vista de datos; los cuales contienen abundante información sobre el universo, su composición, estructura, origen y destino.

Un claro ejemplo de esto lo constituye el *Atacama Large Millimiter/sub-millimiter Array (ALMA)* [Wootten and Thompson, 2009], un interferómetro radio-astronómico que consiste en un arreglo de 66 antenas que observan el espacio en las bandas milimétricas y submilimétricas del espectro electromagnético. Ubicado en el desierto de Atacama, en el norte de Chile, es parte de uno de los proyectos científicos más importantes del último tiempo; en el cual se ha hecho uso de tecnologías de punta por parte de investigadores, ingenieros y técnicos expertos en diversas áreas del conocimiento, tales como la astronomía, la computación científica y de alto rendimiento, la electrónica, entre otros.

La tecnología involucrada en el proyecto *ALMA* ha permitido, entre otras cosas, obtener datos de alta resolución provenientes de distintas fuentes u objetos del espacio observable desde la tierra. La radiación electromagnética emitida por estos objetos, en bandas de frecuencia de radio, son captadas por el arreglo de antenas y posteriormente procesadas por equipos de alta capacidad con el fin de obtener los espectros electromagnéticos correspondientes. Estos, a su vez pueden ser analizados directamente o utilizarse para generar imágenes de

alta calidad.

Parte principal de la importancia de estos espectros de radiación electromagnética es que dan información valiosa sobre la composición química de los objetos de los que esta proviene. Esto se debe a que los átomos que componen estos objetos emiten o absorben una mayor cantidad de energía en frecuencias muy específicas. Por lo tanto, un espectro en particular tendrá rangos estrechos de mayor o menor intensidad en ciertas frecuencias dependiendo de los elementos químicos de los que está compuesto el objeto del que proviene.

La detección de lineas espectrales es un problema de interés en sí, y que puede llegar a ser muy complejo dependiendo de en qué bandas de frecuencia se esté trabajando. Sin embargo, se puede seguir obteniendo información valiosa de los objetos observados a partir de estas líneas ya detectadas. Esto incluye potencialmente respuestas a preguntas como: ¿de qué forma se relacionan ciertos tipos de líneas entre sí? ¿Existe una mayor correlación de presencia de líneas de ciertos isótopos o moléculas en particular? ¿Hay una mayor presencia de líneas de ciertas especies en algunos objetos que en otros? ¿Qué nos dice esto de la composición de los objetos y de su química subyacente?

Existe, en el dominio de la minería de datos, el concepto de *reglas de asociación*; las cuales corresponden a asociaciones lógicas entre conjuntos de elementos o ítems. Si bien estos inicialmente fueron concebidos con el fin de resolver problemas pertenecientes al ámbito del comercio y las ventas, hoy en día son aplicados en los más diversos contextos. Es, por lo tanto, una de las finalidades principales de este trabajo, el mostrar que la espectroscopía astronómica no es la excepción, y que es posible extraer reglas de asociación entre líneas espectrales obtenidas a partir de observaciones de objetos del espacio.

Se espera que el generar de manera automática reglas de asociación entre líneas moleculares facilite, a futuro, el análisis de la naturaleza de los objetos observados, y las caracterísicas de su composición química; al averiguar cómo se relacionan entre sí los elementos, átomos, moléculas e isótopos presentes en las sustancias que los componen. Esto, sobre todo, en vista de que hoy en día el volumen de datos generados a partir de observaciones astronómicas no deja de aumentar.

Objetivos

Objetivo General

- Implementar un sistema de aprendizaje de reglas de asociación, o *Association Rule Learning (ARL)*, que permita obtener relaciones lógicas entre líneas espectrales presentes dentro de un conjunto de datos de espectroscopía astronómica.

Objetivos Específicos

- Implementar un sistema de ARL genérico que permita aplicarse a datos provenientes de diversos orígenes.
- Obtener reglas de asociación entre líneas espectrales obtenidas a partir de datos reales.
- Visualizar las reglas de asociación, presentes en el conjunto de datos, que sean de mayor interés según medidas estadísticas.
- Filtrar las reglas de asociación encontradas en un conjunto de datos de espectroscopía astronómica según las líneas que las componen.

Descripción de la solución

Si bien existen diversas técnicas de clasificación y caracterización de puntos en un espacio multidimensional (en nuestro caso objetos descritos por parámetros), para resolver las preguntas anteriores se requiere más bien de una herramienta que permita encontrar relaciones explícitas entre los parámetros en sí, y que permita asignar medidas de relevancia estadística a estas relaciones.

Para ello se planteó el uso de *Association Rule Learning (ARL)*, o Aprendizaje de Reglas de Asociación, como una herramienta que puede dar respuesta directa a algunas de las interrogantes mencionadas anteriormente, y ayudar a obtener información clave para el proceso de utilizar otras técnicas en el largo plazo.

El Aprendizaje de Reglas de Asociación como técnica se ubica dentro del área de la minería de base de datos, y su concepción original fue el ser aplicada a sistemas de puntos de venta con el fin de encontrar las relaciones más comunes entre artículos comprados por los clientes. Sin embargo, con el tiempo se ha convertido en una de las herramientas más utilizadas de su área, en una diversa gama de contextos.

En el presente trabajo se llevó a cabo el uso de esta técnica con el fin de encontrar relaciones comunes entre líneas espectrales a través de distintos espectros de frecuencia. Ahora bien, la naturaleza innata de estos es más bien contínua y las líneas en sí mismas poseen diversos parámetros que las caracterizan. Por lo tanto, este caso dista mucho de la binaridad del problema original para el cual se pensó ARL. Sin embargo, como se muestra a lo largo de este informe, si se asume que se realizó con anterioridad un buen trabajo de detección de líneas y se efectúa un pre-procesamiento adecuado de los datos, el algoritmo de ARL arroja resultados que están en concordancia con la química subyecente.

En particular, se utilizó una implementación de dos de los algoritmos más utilizados de ARL: *Apriori* y *FP-Growth*. Luego, se obtuvo una base de datos de líneas espectrales ya detectadas (pero no necesariamente asociadas a alguna especie [átomo, isótopo, etc.]) correspondientes a observaciones del *Sloan Digital Sky Survey (SDSS)*, un sondeo espectroscópico del espacio realizado con un telescopio óptico. Sobre este conjunto de datos se procedió a realizar un pre-procesamiento que, entre otros, consta de filtrar las líneas según su brillo o razón señal a ruido. Luego, se efectuaron particiones según las características de los objetos

de procedencia (como tipo de objeto o estructura estelar, cercanía, etc.). Finalmente, sobre estas se procedió a aplicar los algoritmos de ARL.

Los resultados obtenidos fueron efectivamente reglas de asociación entre líneas espectrales que resultaron tener mayor relevancia sobre el conjunto de datos bajo distintas criterios, al aplicarse sobre conjuntos de datos reales obtenidos a partir de observaciones en el espectro visible. Se logró observar conjuntos de especies que están presentes con mayor frecuencia, y las reglas que estos generan; con sus medidas estadísticas correspondientes. Además se logró inferir conclusiones sobre el desempeño de los algoritmos implementados al comparar su eficiencia sobre datos reales con su eficiencia sobre datos simulados.

Queda para desarrollo a futuro una implementación más general del procedimiento para así aplicar los algoritmos a datos obtenidos en otras bandas de frecuencia, como por ejemplo, las observaciones radioastronómicas de ALMA; que por sus caracterísiticas, promete un mayor número de datos sobre los cuales obtener reglas de asociación para líneas espectrales.

Capítulo 1

Marco Teórico

1.1. Ciencia de datos: espectroscopía astronómica

Durante el siglo XIX nace la astrofísica moderna. Fue entonces que, por primera vez, se logró medir distancias estelares; que revelaron lo lejanos que se encuentran estos objetos de la tierra. Surgió, también en aquel siglo, la *espectroscopía física*, que permitió la identificación de elementos químicos a través de *líneas espectrales*. A partir de esto nace la química moderna, con el descubrimiento de la tabla periódica de los elementos. Gracias a estos avances es que, posteriormente, llega a surgir la mecánica cuántica en el siglo XX y, junto con ello, la clasificación espectral de las estrellas.

En el año 1814, el científico Joseph von Fraunhofer (1787 - 1826), mediante el uso de prismas de alta calidad construidos por él mismo, logró difractar un rayo de luz solar y proyectarlo hacia un muro blanco. Además de los colores característicos del arcoíris, observados de esta manera desde los tiempos de Newton, vio en la proyección resultante muchas líneas oscuras. Procedió, luego, a catalogar meticulosamente la longitud de onda exacta de cada una de estas líneas, que hasta el día de hoy se conocen como líneas de Fraunhofer, y asignó letras a las más notorias. De esta forma, Fraunhofer registró el primer espectro astronómico de alta resolución. En la Figura 1.1 se puede apreciar un espectro de los catalogados por Fraunhofer.

Posteriormente, procedió a realizar el mismo experimento, pero esta vez utilizando un rayo de luz proveniente de la estrella roja cercana Betelgeuse, y observó que el patrón de líneas oscuras cambiaba considerablemente. Fraunhofer concluyó correctamente que estas se encuentran de cierta forma relacionadas con la composición del objeto observado. En efecto, algunas de las líneas observadas por Fraunhofer se deben a las especies (e.g átomos, iones, moléculas) que componen la atmósfera terrestre.

Sin embargo, el gran paso en la comprensión general de las observaciones de Fraunhofer llegó a mediados del siglo XIX de la mano del trabajo de los científicos Gustav Kirchhoff (1824 - 1887) y Robert Bunsen (1811 - 1899), quienes estudiaron el color de la luz emitida al poner distintos metales en llamas. Al hacer esto, descubrieron que, en ciertos casos, la longitud de

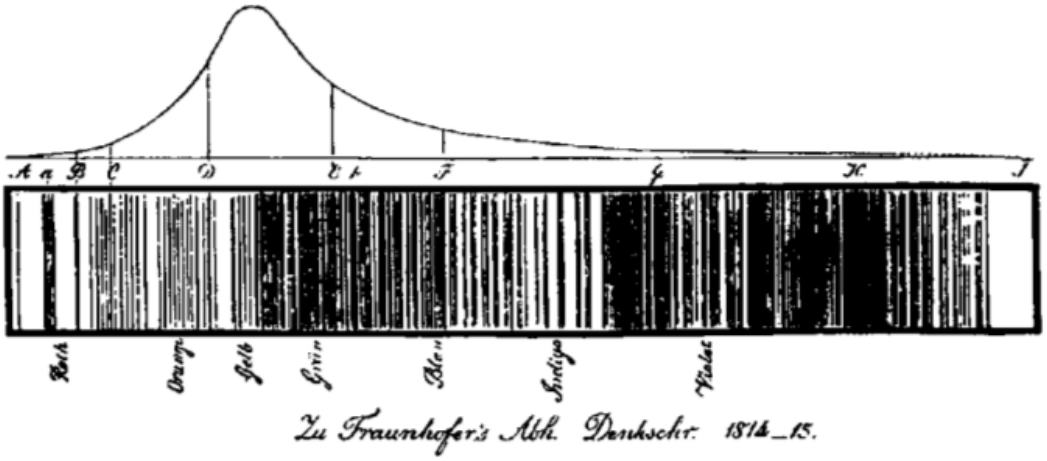

Figura 1.1: Espectro solar registrado por Fraunhofer [Tennyson, 2010].

onda de la luz emitida coincidía exactamente con las líneas observadas por Fraunhofer. Estos experimentos demostraron que las líneas de Fraunhofer son una consecuencia directa de la composición atómica del sol.

En el siglo XX se llegó a comprender de manera más profunda la razón de la existencia de estas líneas, denominadas *líneas espectrales*, gracias a la revolución que significó la llegada de la mecánica cuántica. Los desarrollos en materia de espectroscopía han estado, desde entonces, estrechamente ligados a los de aquel campo de la física.

Si se observa cuidadosamente ciertos objetos, tales como los planetas Marte o Júpiter, o estrellas tales como Betelgeuse, se puede apreciar que estos objetos tienden a tener un cierto color. Basta utilizar instrumentos de bajo poder resolutivo para separar la luz que llega desde estos objetos a la tierra en colores de amplio espectro. A su vez, el observar estos colores entrega información sobre la temperatura del objeto. Por ejemplo, las estrellas azules poseen mayor temperatura que las rojas. Objetos que emiten rayos X, como la corona solar, son muy calientes, mientras que objetos fríos emitirán radiación en longitudes de onda mayores; por ejemplo, en forma de ondas de radio.

La mejor forma de obtener información astrofísica detallada de objetos del cielo es mediante observaciones de alta resolución espectral. Observaciones llevadas a cabo con equipos de tal capacidad permiten obtener, no solamente la posición central de una línea dentro del espectro, sino también su forma. Mediante este procedimiento se puede inferir propiedades del objeto, tales como su composición química, su temperatura, la abundancia de las especies que lo componen y que se encuentran emitiendo radiación, el movimiento de las especies y del objeto en sí, la presión y densidad local, el campo magnético presente, entre otros.

Esto se lleva a cabo con equipos de alto poder resolutivo y sensibilidad. Dos ejemplos de estos son, el telescopio óptico SDSS que se encuentra en el Apache Point Observatory (APO, ubicado en Nuevo México, Estados Unidos) y con el cual se lleva a cabo el *Sloan Digital Sky Survey (SDSS)*; y, en mayor medida, el interferómetro radioastronómico *Atacama Large Millimeter/submillimeter Array (ALMA)* ubicado en el norte de Chile.

1.1.1. Atacama Large Millimeter Array (ALMA)

El *Atacama Large Millimeter Array (ALMA)* [alm, 2014] [Wootten and Thompson, 2009] es un interferómetro de señales de radio ubicado en el desierto de Atacama, en el norte de Chile. Es un proyecto llevado a cabo mediante una asociación de organizaciones de Norteamérica, Europa y el Este de Asia. Comenzó sus observaciones científicas en la segunda mitad del año 2011. Es, por lejos, el mayor y más importante radiotelescopio construido hasta la fecha. Se encuentra realizando observaciones preliminares desde marzo del año 2013, y se espera que opere al cien por ciento de su capacidad desde marzo del 2017.

ALMA realiza observaciones captando radiación electromagnética proveniente del espacio en bandas milimétricas y submilimétricas en sus longitudes de onda, que corresponden a ondas de radio. Debido a que en condiciones normales la humedad del ambiente y del cielo absorbe gran parte de este tipo de radiación, es crucial para el funcionamiento de los telescopios el estar ubicados en un lugar seco; y el más idóneo en ese sentido es, sin dudas, el llano de Chajnantor en el desierto de Atacama, a más de 5000 metros de altura.

Debido al diseño de ALMA, en muchas de sus observaciones se detectará una abundancia de líneas espectrales; lo cual puede ser un resultado complementario al objetivo principal de una observación en particular, y por ende, puede no ser analizado por el o la astrónomo(a) que lo propuso.

Con el tiempo se espera ocurra una eventual acumulación de grandes cantidades de datos espectrales de ALMA. Esto abre la oportunidad de desarrollar nuevas técnicas de estudio basados en la minería de datos u otras técnicas de computación poco usadas por los astrónomos. De ahí que en el presente trabajo se busque implementar algoritmos de aprendizaje de reglas de asociación, o *Association Rule Learning (ARL)*, para el estudio masivo de datos espectroscópicos

Gran parte de los datos obtenidos desde ALMA son guardados en estructuras de datos llamadas cubos de datos tipo ALMA (o *ALMA Data Cubes*), como el que se observa en la Figura 1.2, que contienen información de distintos puntos de observación del cielo a distintas frecuencias. Los cubos de datos tipo ALMA, como estructura de datos, contienen valores indexados en tres coordenadas. Dos de las coordenadas son espaciales, y corresponden al equivalente a una imagen normal de dos dimensiones, en el sentido que describen puntos del cielo (o del espacio observable desde la tierra). La tercera coordenada corresponde al rango de frecuencias en el que se está detectando radiación electromagnética. Por lo tanto, si se fijan las dos coordenadas espaciales (se fija un punto en el espacio) y se extraen todos los valores en la tercera coordenada de aquel punto, se obtiene el espectro de frecuencias observado en ese punto del espacio.

A partir de ALMA se generan enormes cantidades de datos, los cuales necesariamente deben procesarse por parte de sistemas automatizados de extracción y análisis con el fin de facilitar a los investigadores el inferir información útil a partir de estos.

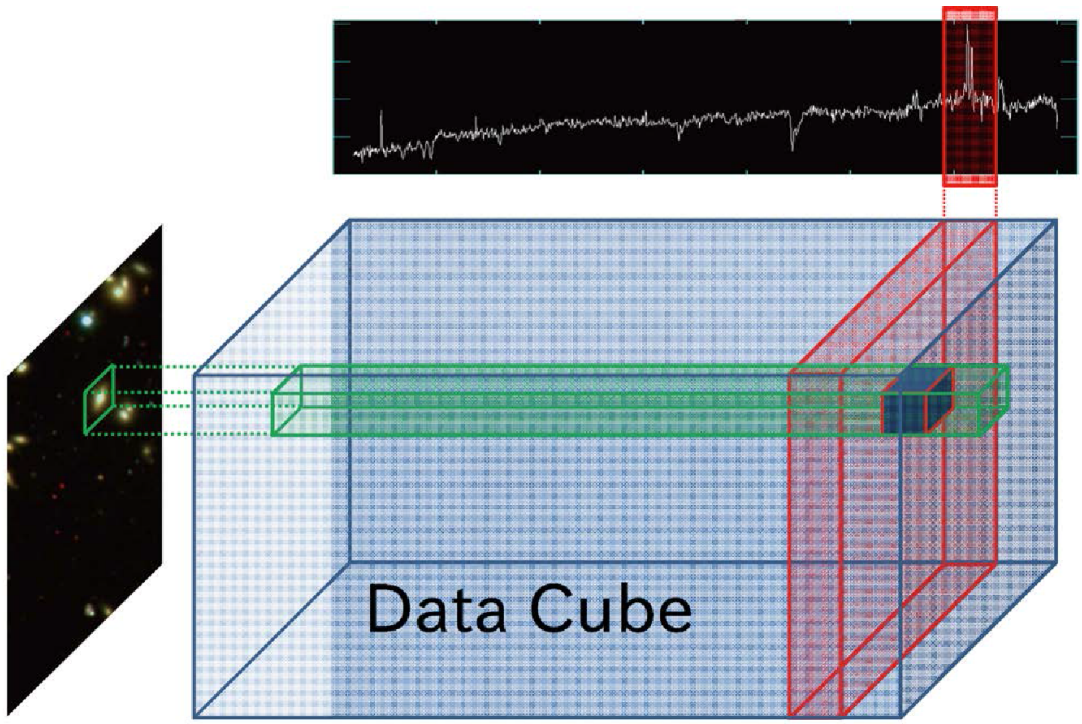

Figura 1.2: Representación gráfica de un cubo de datos tipo ALMA. Dos de sus cordenadas son espaciales mientras la tercera corresponde al dominio de las frecuencias [Eguchi, 2013].

1.1.2. Sloan Digital Sky Survey (SDSS)

Dado que se espera obtener datos de ALMA para el uso de técnicas tales como el aprendizaje de reglas de asociación a partir del año 2017, se requiere una base de datos espectroscópicos pre-existente con el fin de poner a prueba el sistema desarrollado en el presente trabajo.

El *Sloan Digital Sky Survey (SDSS)* [York et al., 2000] es un proyecto de inspección y estudio del espacio llevado a cabo mediante el uso de un telescopio óptico ubicado en el observatorio Apache Point (APO), Nuevo México, Estados Unidos. La recolección de datos comenzó en el año 2000, y las imágenes finales de los datos publicados cubren un 35 % del cielo, con observaciones fotométricas de 500 millones de objetos y espectros ópticos de 1 millón de objetos.

Los espectros del SDSS cubren desde 3600 a 10400 Angstroms (Å)[1] con una resolución de 1 Å[2]. Los objetos estudiados son principalmente galaxias, incluyendo *quásares* y *AGN* (un 80 % del total de datos), y el resto son estrellas de distinto tipo (20 % del total) cuyos espectros se encuentran dominados por muchas líneas de absorción, como el que se muestra en la Figura 1.3. Los espectros de regiones de gas o de galaxias, por otra parte, poseen pocas líneas de absorción. El SDSS tiene en sus catálogos un universo de casi 50 líneas espectrales posibles previamente identificadas, presentes dentro de su rango de detección.

[1] `https://www.sdss3.org/instruments/boss_spectrograph.php#Parameters`
[2] $1\,\text{Å} = 10^{-10}\,m = 10^{-1}\,nm$

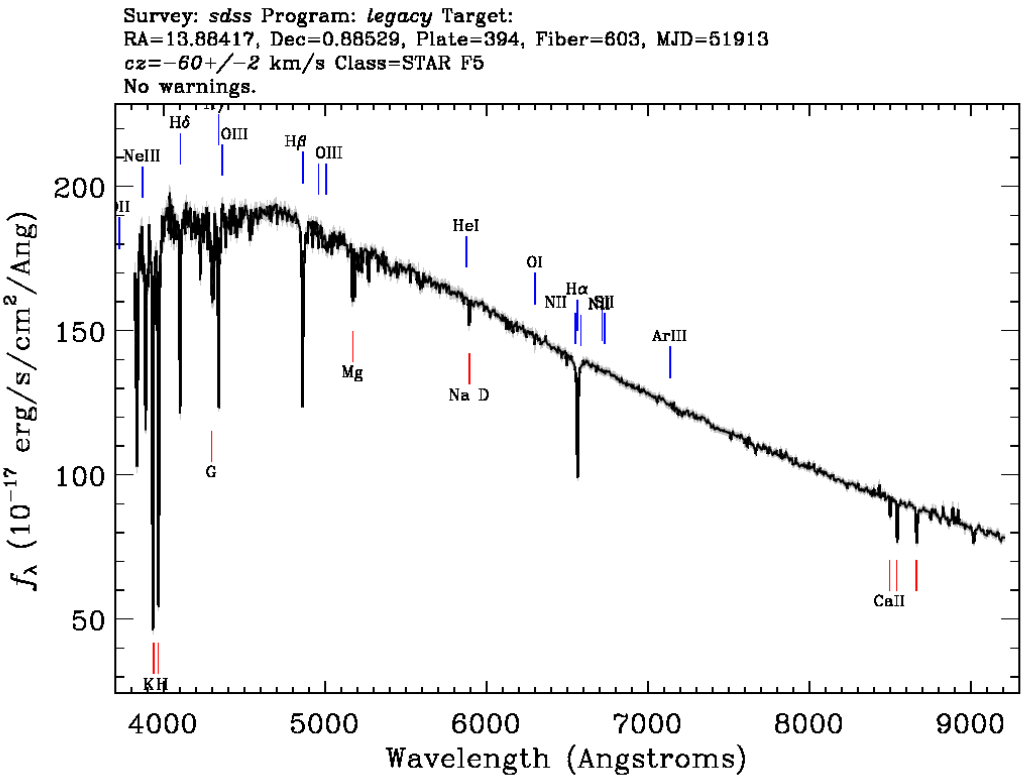

Figura 1.3: Espectro de frecuencia (flujo versus longitud de onda) de un objeto estelar del SDSS con us líneas identificadas. Se puede apreciar que el espectro es dominado por líneas de absorción. Nótese que existen líneas de absorción no identificadas cerca de $\lambda = 8800$ Å (imagen obtenida desde *sdss.org*).

Los datos de SDSS se hacen disponibles mediante publicaciones regulares o *data releases* a través de internet. La última publicación llevada a cabo fue la correspondiente al data release 10 (DR10), con fecha de julio del 2013. Los datos de todos los data releases se encuentran en un servidor *Microsoft SQL Server* y pueden accederse mediante diversas interfaces o APIs presentes en el sitio web de SDSS. En particular, existe una interfaz web llamada *CasJobs* que permite realizar consultas en lenguaje *SQL* a un servidor que encola la petición, la ejecuta y guarda los resultados en una base de datos asignada al usuario.

Para probar los algoritmos y el sistema implementados en el presente trabajo, en particular, se utilizó el *data release 7 (DR7)* como fuente de datos, ya que es el último *data release* que contiene información completa sobre las líneas detectadas en los distintos espectros de frecuencia.

1.2. Reglas de asociación

El aprendizaje mediante reglas de asociación, o *Association Rule learning (ARL)*, es sin lugar a dudas uno de los métodos más populares y mejor estudiados dentro de la minería de datos. Basta para ello ver que el artículo seminal de Agrawal et al. [Agrawal et al., 1993],

donde se sentaron las bases de la teoría subyacente, es uno de los más citados del área; según el catálogo y herramienta de búsqueda de publicaciones científicas *Google Scholar*.

La motivación principal de ARL en su concepción fue el encontrar relaciones lógicas entre los artículos adquiridos por usuarios en puntos de venta del tipo *"Si un cliente compra los artículos A y B, entonces es muy probable que también compre el artículo C"*. Sin embargo, la teoría de fondo que se desarrolló con el tiempo tiene una gran cantidad de aplicaciones en los más diversos ámbitos.

1.2.1. Definición formal

Sea $\mathcal{I} = \{\mathrm{i}_1, \mathrm{i}_2, \mathrm{i}_3, \ldots, \mathrm{i}_n\}$ un universo de ítems posibles. Se denomina, entonces a un conjunto $X \subseteq \mathcal{I}$ como *conjunto de ítems* o *itemset*. Se tiene, además un conjunto de transacciones $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, donde $T_\mathrm{i} \subseteq \mathcal{I},\ \forall \mathrm{i} \in [1, m]$. Dados un conjunto de ítems X y una transacción T_i, se dice que la trasacción T_i *satisface* X si y solo si $X \subseteq T_\mathrm{i}$.

Una *regla de asociación* es, entonces, una relación entre dos conjuntos X e Y, donde $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$, y $X \cap Y = \emptyset$. Esta relación se denota de la forma $X \Rightarrow Y$. A X se denomina el *antecedente* de la regla y a Y se denomina el *consecuente* de la regla.

Existen una serie de medidas para cuantificar la relevancia de una regla de asociación. A continuación se definen algunas de ellas.

El *soporte* de un conjunto de ítems X, o $supp(X)$, se define como

$$supp(X) = \frac{|\mathcal{T}_X|}{|\mathcal{T}|} \quad \text{, tal que} \quad \mathcal{T}_X = \{T \in \mathcal{T} : X \subset T\},$$

donde $|X|$, cuando X es un conjunto finito cualquiera, es el número de elementos que posee X. Vale decir, el soporte corresponde a la fracción del total de transacciones en la que está presente el conjunto.

A su vez, el soporte de una regla de asociación $X \Rightarrow Y$, o $supp(X \Rightarrow Y)$, se define como

$$supp(X \Rightarrow Y) = supp(X \cup Y),$$

vale decir, corresponde a la fracción del total de transacciones en las cuales está presente tanto el antecedente como el consecuente de la regla simultáneamente[3].

La *confianza* de una regla de asociación $X \Rightarrow Y$, denotada por $conf(X \Rightarrow Y)$, se define como

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)},$$

[3]Debe tenerse en mente que la expresión $supp(X \cup Y)$ indica la fracción del total de transacciones en las cuales está presente **tanto** el antecedente como el consecuente de la regla **simultáneamente**, y **no** de aquellas en las cuales está presente el antecedente **o** el consecuente. El soporte de un conjunto decrece en la medida que el número de elementos que contiene aumenta

es decir, indica en qué fracción de las transacciones en las cuales está presente el antecedente la regla se cumple (i.e. está presente también el consecuente de la regla).

El *lift* de una regla de asociación $X \Rightarrow Y$, denotado por $\mathit{lift}(X \Rightarrow Y)$, se define como

$$\mathit{lift}(X \Rightarrow Y) = \frac{\mathit{conf}(X \Rightarrow Y)}{\mathit{supp}(Y)} = \frac{\mathit{supp}(X \cup Y)}{\mathit{supp}(X) \times \mathit{supp}(Y)}.$$

La intuición detrás del concepto de lift tiene lugar al interpretar las medidas descritas anteriormente desde un punto de vista probabilístico. Tomando el conjunto $\mathcal{T}$ como un universo de posibles resultados, o espacio muestral, se tiene que

$$\mathit{supp}(X) = P(X) \quad \text{y} \quad \mathit{conf}(X \Rightarrow Y) = P(Y|X).$$

Desde este punto de vista, la medida de lift indica qué tan bien la presencia del antecedente de una regla lograría predecir la presencia del consecuente. Por lo tanto, si la presencia del antecedente y del consecuente en una transacción cualquiera son eventos estadísticamente independientes (i.e. la ocurrencia de uno no afecta la probabilidad de que el otro ocurra), se tendrá que $\mathit{lift}(X \Rightarrow Y) = 1$; y este valor irá variando en la medida que ambos eventos sean más dependientes entre sí.

Por ejemplo, supongamos que se tiene el siguiente conjunto de transacciones

TID	**Items**
1	a, c
2	a, d
3	b, c
4	b, d

donde *TID* es el número identificador de la transacción. Luego, para este caso, se tiene que

$$\mathit{lift}(\{a\} \Rightarrow \{c\}) = \frac{\mathit{supp}(\{a\} \cup \{c\})}{\mathit{supp}(\{a\}) \times \mathit{supp}(\{c\})} = \frac{1/4}{1/2 \times 1/2} = 1,$$

lo cual indica que la que la ocurrencia de que una transacción cualquiera satisfaga $\{a\}$ es estadísticamente independiente de que una transacción cualquiera satisfaga $\{b\}$.

En cambio, en el siguiente conjunto de transacciones

TID	**Items**
1	a, c
2	a, d
3	b, c
4	b, c

se tiene que

$$\mathit{lift}(\{a\} \Rightarrow \{c\}) = \frac{\mathit{supp}(\{a\} \cup \{c\})}{\mathit{supp}(\{a\}) \times \mathit{supp}(\{c\})} = \frac{1/4}{1/2 \times 3/4} = 2/3 < 1,$$

lo cual quiere decir que hay una mayor razón de transacciones que satisfacen $\{c\}$ dentro del total de transacciones que dentro del conjunto de transacciones que satisfacen $\{a\}$.

Finalmente, en el conjunto de transacciones

TID	Items
1	a, c
2	a, d
3	b, d
4	b, d

se cumple que

$$\mathit{lift}(\{a\} \Rightarrow \{c\}) = \frac{\mathit{supp}(\{a\} \cup \{c\})}{\mathit{supp}(\{a\}) \times \mathit{supp}(\{c\})} = \frac{1/4}{1/2 \times 1/4} = 2 > 1,$$

lo cual indica que hay una mayor razón de transacciones que satisfacen $\{c\}$ dentro del conjunto de transacciones que satisfacen $\{a\}$ que dentro del total de transacciones.

1.2.2. Algoritmos principales

Algoritmo *Apriori*

En el mismo artículo seminal de ARL por Agrawal et al. [Agrawal et al., 1993], se presentó el algoritmo *Apriori*. El algoritmo *Apriori* recibe como entrada un conjunto de transacciones, y tiene como objetivo encontrar y retornar todos aquellos conjuntos presentes que cumplan con el requisito de soporte mínimo indicado, también llamados *conjuntos frecuentes*. Este algoritmo hace uso de las propiedades de clausura descendiente de la frecuencia de los conjuntos con respecto a sus subconjuntos con el fin de optimizar el proceso de generación de conjuntos de ítems frecuentes.

Por ejemplo, supongamos que se cuenta con un conjunto de transacciones, y que cada una contiene ítems pertenecientes a un universo de solo 4 posibles, $\mathcal{I} = \{0, 1, 2, 3\}$. Luego, en principio, para extraer los conjuntos frecuentes a partir de estas transacciones, por cada uno de los conjuntos que es posible generar con este universo de 4 ítems posibles (llamados *conjuntos candidatos*), se debe recorrer cada una de las transacciones, ver si la transaccion satisface este conjunto, y de ser así incrementar un contador. Luego de terminar este proceso para cada uno de los conjuntos posibles, se tendrá el número de veces que cada uno de estos se encuentra dentro del conjunto de transacciones, y teniendo el número total de estas, se puede obtener de forma directa el soporte de estos conjuntos. Por ejemplo, en la Figura 1.4 se observa todos los conjuntos candidatos que se pueden generar a partir de $\mathcal{I}$.

El problema radica en que el número de conjuntos candidatos crece de manera exponencial en el número de ítems del universo posible. En efecto, si el número de ítems del universo es n, entonces a partir de este es posible generar $2^n + 1$ conjuntos. Por tanto, para un universo de 100 elementos, existen nada menos que $1{,}26 \times 10^{30}$ conjuntos candidatos; y debe, por tanto, recorrerse el total de transacciones este número de veces.

No obstante, es posible reducir el número de conjuntos candidatos utilizando la propiedad de *clausura descendiente* de los conjuntos frecuentes, tambien llamado *principio Apriori*. Esta

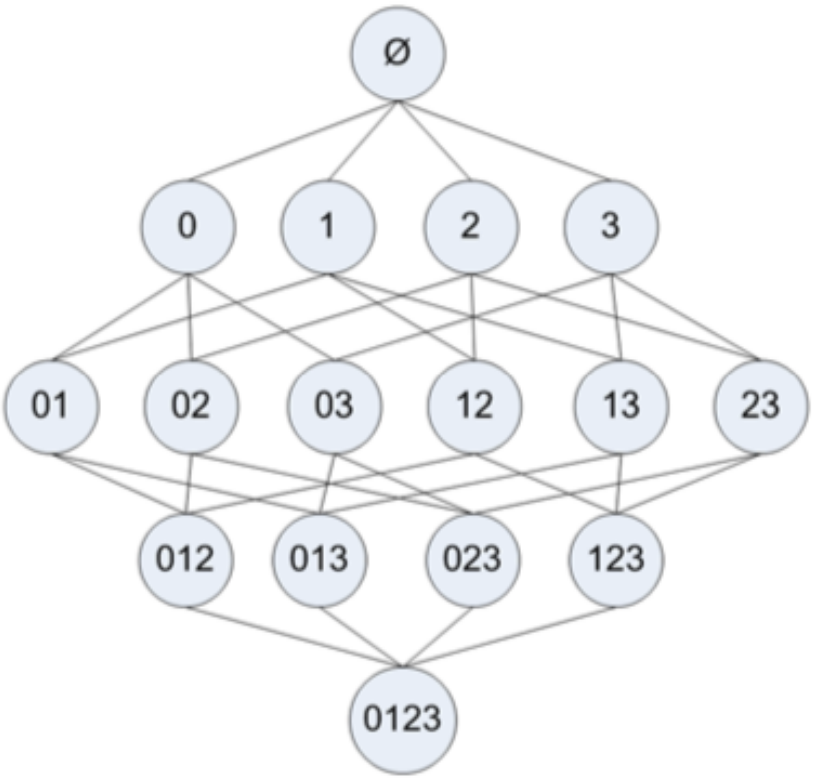

Figura 1.4: Grafo que muestra todos los conjuntos posibles generados a partir del conjunto universo {0, 1, 2, 3}[Harrington, 2012]

propiedad asegura que si un conjunto dado es, en efecto, frecuente, entonces necesariamente todos sus subconjuntos también lo son. O, expresado de forma recíproca, si un conjunto dado resulta no ser frecuente, entonces necesariamente todos sus superconjuntos tampoco lo son. Esta última expresión es la que resulta más relevante para nuestro caso. Esto implica que luego de generar un conjunto candidato y verificar si es frecuente verificando el número de transacciones que lo satisfacen, si se comprueba que este conjunto no es frecuente (vale decir, no cumple con el requisito de soporte mínimo), entonces necesariamente ninguno de los superconjuntos posibles que lo contienen será frecuente, y por tanto no será necesario obtener sus soportes correspondientes contando el número de transacciones que los satisfacen; como se aprecia en la Figura 1.5.

Esta propiedad permite reducir considerablemente el número de conjuntos candidatos y, por tanto, optimizar el algoritmo final; ya que no será necesario recorrer el total de transacciones tantas veces como se planteó originalmente. Para poder utilizar esta propiedad y beneficiarse de la optimización correspondiente, es necesario generar los conjuntos candidatos comenzando por aquellos que poseen menos elementos, y a partir de estos generar todos los superconjuntos posibles.

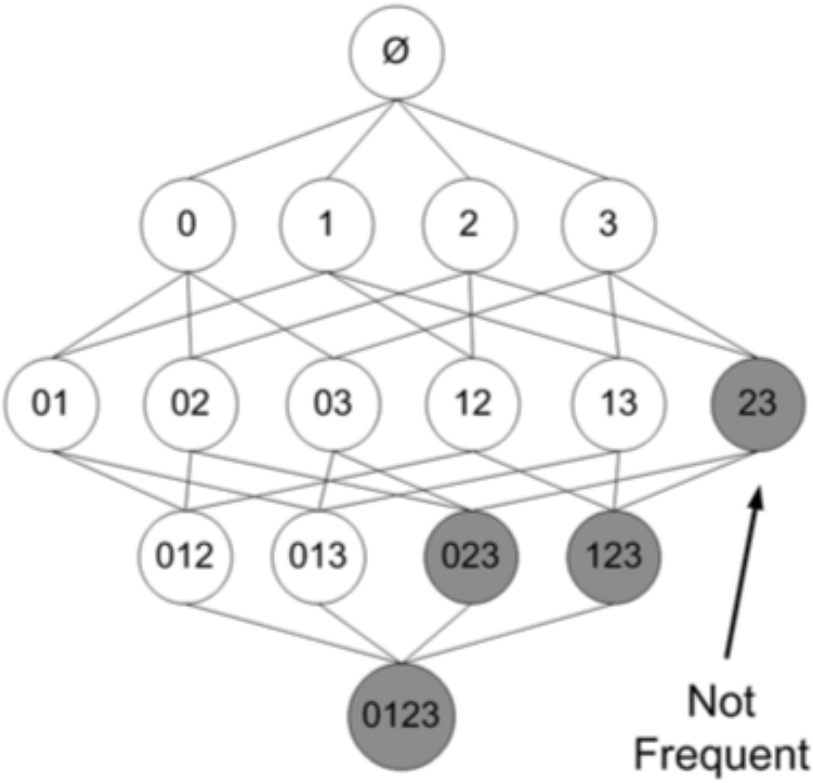

Figura 1.5: Grafo que muestra todos los conjuntos posibles generados a partir del conjunto universo {0, 1, 2, 3}. Los conjuntos en gris son aquellos que de inmediato se sabe no son frecuentes si el conjunto {2, 3} resulta no serlo [Harrington, 2012].

El algoritmo *Apriori*, por lo tanto, en terminos generales resulta ser el siguiente

Algoritmo 1: Algoritmo *Apriori*

Data: Conjunto de transacciones $\mathcal{T}$
Result: Conjunto de ítems frecuentes $\mathcal{L}$
$\mathcal{L}_1 \leftarrow$ {conjuntos frecuentes de 1 solo ítem}
for $k = 2$; $\mathcal{L}_{k-1} \neq \emptyset$; $k++$ **do**
  $C_k =$ `apriori-gen`$(\mathcal{L}_{k-1})$
  $\forall$C $\in \mathcal{C}_k$, C.count $= 0$
  for *transacciones* $T \in \mathcal{T}$ **do**
    $\mathcal{C}_T =$ `subset`$(\mathcal{C}_k, T)$
    for *candidatos* $C \in \mathcal{C}_T$ **do**
      C.count $\leftarrow$ C.count $+ 1$
  $\mathcal{L}_k = \{C \in \mathcal{C}_k$: C.count $\geq$ minsup$\}$
$\mathcal{L} \leftarrow \bigcup_k \mathcal{L}_k$

Donde $\mathcal{L}_k$ corresponde a la colección de conjuntos frecuentes con k elementos, los cuales tienen un contador asociado; y $\mathcal{C}_k$ consiste en la colección de conjuntos candidatos con k elementos, que tienen también un contador asociado. La función `apriori-gen` es la encargada de generar una colección de conjuntos frecuentes de tamaño $k + 1$ a partir de una colección

de conjuntos candidatos datos de tamaño k. La función `subset` se encarga de recibir una colección de conjuntos de ítems frecuentes $\mathcal{C}_k$ y una transacción T y de retornar una colección de ítems $\mathcal{C}_T = \{C \in \mathcal{C}_k \,:\, C \subseteq T\}$.

Posteriormente, se lleva a cabo la extracción de reglas a partir de la colección de conjuntos frecuentes $\mathcal{L}$. Para ello, se utiliza el algoritmo *Apriori* de generación de reglas, que se detalla a continuación.

Algoritmo 2: Algoritmo *Apriori* de generación de reglas

Data: Colección de conjuntos frecuentes $\mathcal{T}$
Result: Colección de reglas de asociación $\mathcal{R}$
forall the *conjuntos frecuentes* $l_k \in \mathcal{T}$, $k \geq 2$ **do**
  yield $\mathcal{R}$ = `genRules`(l_k, l_k)

Procedimiento genRules(l_k: conjunto de k ítems, a_m: conjunto de m ítems)

$A = \{a_{m-1} \,:\, a_{m-1} \subset a_m\}$
forall the $a_{m-1} \in A$ **do**
  conf = `support`(l_k)/`support`(a_{m-1})
  if *conf* $\leq$ *minconf* **then**
    yield regla $a_{m-1} \Rightarrow (l_k - a_{m-1})$, con confianza = conf y soporte = `support`(l_k)
    if $m - 1 > 1$ **then**
      `genRules`(l_k, a_{m-1})

Básicamente, lo que hace el algoritmo *Apriori* de generación de reglas es recibir una colección de conjuntos frecuentes de distintos tamaños, e invocar al procedimiento `genRules` sobre pares de conjuntos que sean del mismo tamaño k. A su vez, el procedimiento `genRules` recibe dos conjuntos cualquiera de ítems l_k y a_m, obtiene todos los subconjuntos de a_m de tamaño $m-1$ (a_{m-1}), y genera reglas que tengan a estos a_{m-1} como antecedente y la diferencia entre l_k y a_{m-1} como consecuente; siempre y cuando cumplan con el requisito de confianza mínima. Luego, si m sea mayor o igual a 2, el procedimiento se llama a sí mismo recursivamente con l_k y a_{m-1} como argumento, con el fin de generar esta vez reglas cuyo antecedente tenga $m-1$ elementos. De esta manera se generan reglas con todas las combinaciones posibles de antecedente y consecuente que tengan la confianza mínima indicada.

La teoría indica que la complejidad del algoritmo Apriori está acotada por $\mathcal{O}(\mathcal{C}_{sum} \times |\mathcal{T}|)$, donde $\mathcal{C}_{sum}$ es la suma de los tamaños del total de conjuntos candidatos considerados y $|\mathcal{T}|$ denota el tamaño del conjunto de transacciones.

Algoritmo *FP-Growth*

Más recientemente, Han et al. introdujeron el uso de una estructura de datos llamada *Frequent Pattern Tree* [Han et al., 2004] en la extracción de conjuntos de ítems frecuentes a partir de conjuntos de transacciones. Con esto dieron origen al algoritmo *FP-Growth*.

Un *Frequent Pattern Tree (FP-Tree)* es una estructura de datos de tipo árbol, que consiste en un nodo raíz que tiene como sus hijos a *sub-árboles de prefijos de ítems*. Cada nodo del *sub-árbol de prefijo de ítem* contiene tres campos: el nombre del ítem al cual el nodo representa, un contador que registra el número de transacciones que satisfacen la rama del árbol que va de la raíz hasta este nodo, y un puntero al siguiente nodo del FP-Tree que contenga el mismo nombre de ítem o un puntero vacío si no existe tal nodo.

A su vez, el FP-Tree posee una estructura de datos auxiliar denominada *tabla de encabezados*. Cada entrada en esta tabla posee dos campos. El primero es el nombre del ítem y el segundo es un puntero al primer nodo del FP-Tree que posee el mismo nombre de ítem.

Supongamos, por ejemplo, que se cuenta con el siguiente conjunto de transacciones:

TID	**Ítemes**
1	r, z, h, j, p
2	z, y, x, w, v, u, t, s
3	z
4	r, x, n, o, s
5	y, r, x, z, q, t, p
6	y, z, x, e, q, s, t, m

Supongamos que se desea extraer de estas transacciones aquellos conjuntos frecuentes que cumplan un soporte mínimo de 0.5; vale decir, en este caso, que estén presentes en al menos 3 transacciones. El procedimiento para generar el FP-Tree es, entonces, el siguiente. En primer lugar, se extrae a partir de las transacciones todos los ítems presentes y se ordenan por orden de frecuencia. En este caso, el resultado es el conjunto $I = \{z, r, x, y, s, t, p, q, h, j, w, v, u, n, o, \text{e}, m\}$. Luego, se elimina de este conjunto todos aquellos ítems que no cumplan con el requisito mínimo de soporte deseado, obteniendo como resultado $I = \{z, r, x, y, s, t\}$

Luego, en cada transacción se ordenan sus conjuntos de ítems según su frecuencia, y posteriormente se filtran aquellos ítems que no cumplan con el soporte mínimo deseado. Se genera, de esta forma, un nuevo conjunto de transacciones cuyos ítems se encuentran ordenados según frecuencia y que poseen solamente ítems frecuentes, como se observa en la siguiente tabla:

TID	**Ítemes originales**	**Ítemes ordenados y filtrados**
1	r, z, h, j, p	z, r
2	z, y, x, w, v, u, t, s	z, x, y, s, t
3	z	z
4	r, x, n, o, s	x, s, r
5	y, r, x, z, q, t, p	z, x, y, r, t
6	y, z, x, e, q, s, t, m	z, x, y, s, t

Una vez listo esto, se puede comenzar con el procedimiento de construcción del árbol en sí. Se comienza por insertar el nodo raíz, cuyo nombre es vacío o *null*. Luego se comienza a añadir los conjuntos frecuentes a partir de las transacciones con ítems ordenados y filtrados. Estas son sucesivamente añadidas al árbol de tal manera que cada ítem resulte ser hijo del

ítem anterior según el orden en que se encuentra en la transacción. Ahora bien, si el ítem a añadir ya se encuentra presente como hijo del nodo actual, entonces en vez de agregar un nuevo nodo con el mismo nombre, simplemente se incrementa el contador del nodo ya existente y se inserta el siguiente ítem en la transacción como hijo de este. En la Figura 1.6 se aprecia parte de este proceso.

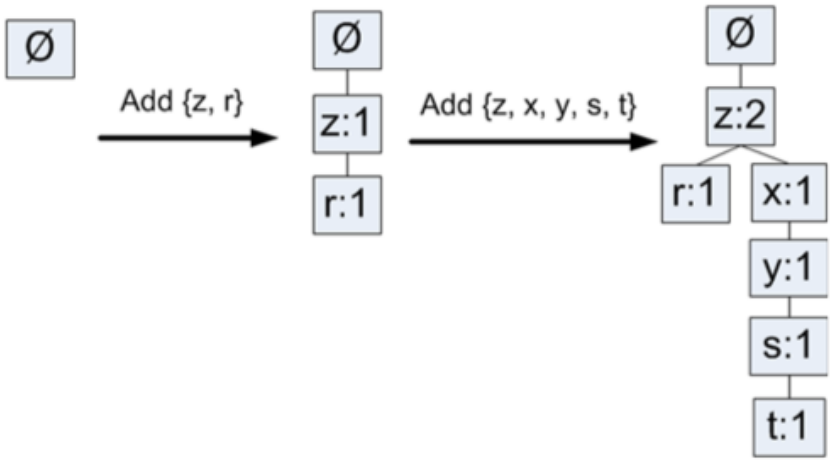

Figura 1.6: Proceso de construcción del FP-Tree del ejemplo. Aquí se puede apreciar cómo ocurre el mecanismo de bifurcación de ramas del FP-Tree al insertar las dos primeras transacciones [Harrington, 2012].

En la Figura 1.7 se muestra el FP-Tree y la tabla de encabezados que se obtiene al final de llevar a cabo el proceso de construcción con los datos de ejemplo.

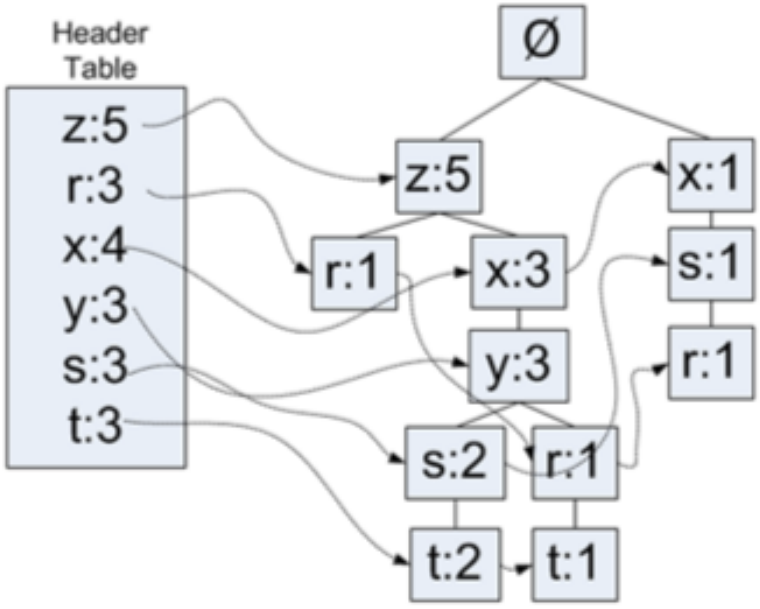

Figura 1.7: FP-Tree y headertables construidos a partir de los datos de ejemplo, con un soporte de 0.5 [Harrington, 2012].

Luego, se procede a extraer los conjuntos frecuentes a partir del FP-Tree, para lo cual no

es necesario hacer uso del conjunto de transacciones originales. Este proceso de extracción consta de los siguientes pasos:

1. Obtener conjunto de patrones condicionales a partir del FP-Tree.

Por cada uno de los conjuntos frecuentes de un solo elemento presentes en la tabla de encabezados, se extrae su *conjunto de patrones condicionales*, que es una colección de *ramas de prefijo*. Cada una de estas ramas de prefijo es un conjunto de ítems presentes en un camino del árbol original que termina en un cierto ítem. Vale decir, son todos los ítems que se encuentran desde la raíz del árbol hasta justo antes del nodo correspondiente a un ítem dado. A cada uno de estos va asociado, además, un contador que posee el mismo valor del mismo contador presente en el nodo donde está presente aquel ítem. Entonces, por ejemplo, el conjunto frecuente $\{r\}$ se encuentra tres veces en el árbol original. Por lo tanto hay tres *ramas de prefijo*; vale decir, tres caminos que van desde el nodo raíz hasta un nodo que contenga a $\{r\}$. Por lo tanto, se tiene que su conjunto de patrones condicionales es una colección que contiene tres conjuntos o ramas de prefijo: $\{x, s\}$:1, $\{z, x, y\}$:1 y $\{z\}$:1 (el número después de los dos puntos indica el contador asociado a r en su nodo correspondiente); tal y como se muestra en la Figura 1.8.

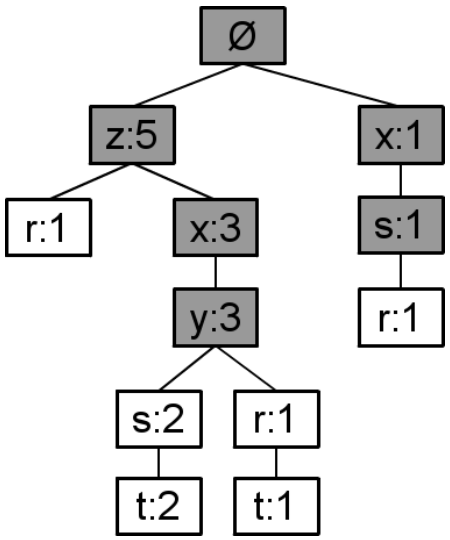

Figura 1.8: Conjunto de ítems que forman parte del conjunto de patrones condicionales del conjunto $\{r\}$; vale decir, todos aquellos que están presentes desde la raíz del árbol hasta justo antes de donde esté presente el ítem r. Por tanto, en este caso el conjunto de patrones condicionales de $\{r\}$ lo conforman los conjuntos $\{x, s\}$, $\{z, x, y\}$ y $\{z\}$.

2. A partir de una de sus ramas de prefijo, construir un FP-Tree condicional.

A partir de las ramas de prefijo que se encuentran dentro del conjunto de patrones condicionales se construye, entonces, un nuevo FP-Tree de la misma manera que se construyó el original a partir de las transacciones. A este nuevo FP-Tree se le denomina *FP-Tree condicional*. Tal como en el caso original, este FP-Tree contiene solo los ítems que cumplen con el requisito de soporte mínimo dentro de la rama de prefijo actual. Luego, se añade a la colección de conjuntos frecuentes los conjuntos que consisten de la unión entre los ítems que se encuentran en la tabla de encabezados y el prefijo a partir del cual se obtuvieron las ramas de prefijo con las que se construyó el FP-Tree actual.

Para ilustrarlo mejor, continuemos con el mismo ejemplo de la parte anterior. Como se mostró, las ramas de prefijo asociadas al conjunto $\{r\}$ conforman el conjunto de patrones condicionales $\{x, s\}$:1, $\{z, x, y\}$:1 y $\{z\}$:1. Sabemos desde ya que $\{r\}$ es un conjunto frecuente, y, por lo tanto, lo agregamos a nuestra colección de conjuntos frecuentes. A partir de estos crearemos un nuevo árbol similar al original; un *FP-Tree condicional*. Trataremos este conjunto de patrones condicionales como si fuese un conjunto de transacciones, y el procedimiento es similar al original, salvo que en esta ocasión los ordenaremos en forma inversa; vale decir, del ítem menos frecuente al más frecuente. El resultado es el siguiente:

Contador	**Patrón condicional ordenado**
1	s, x
1	y, x, z
1	z

Posteriormente, y al igual que en el procedimiento original, procedemos a calcular el soporte de cada ítem y eliminar aquellos que no cumplan con el soporte de 0.5; o sea, que no estén presentes en al menos 3 transacciones. La diferencia es que, esta vez, tenemos cuidado de considerar además el contador asociado a cada patrón condicional. Como se aprecia en la tabla anterior, $\{r\}$ e $\{y\}$ están presentes en 1 patrón condicional (cada uno con un contador de 1 asociado), $\{x\}$ y $\{z\}$ sólo en 2. Por lo tanto, el árbol generado por este conjunto de patrones es vacío.

Intentemos ahora con otro conjunto frecuente de un ítem: $\{t\}$. Sabemos que de antemano que este es un conjunto frecuente, y, por ende, lo agregamos a la colección de conjuntos frecuentes. Luego, procedemos a obtener el conjunto de patrones condicionales de $\{t\}$. Este corresponde a $\{z, x, y, s\}$:2 y $\{z, x, y, r\}$:1. Al igual que en la iteración anterior, ordenamos en sentido creciente de soporte y filtramos los ítemes no frecuentes del conjunto de patrones condicionales. El resultado de esto es:

Contador	**Patrón cond. ordenado**	**Patrón cond. ordenado y filtrado**
2	s, y, x, z	y, x, z
1	r, y, x, z	y, x, z

Se puede observar en la tabla que se eliminaron los subconjuntos $\{s\}$ y $\{r\}$ por tener solamente un contador en total de 2 y 1, respectivamente, en sus patrones condicionales. Luego, con los patrones resultantes se procede a construir un FP-Tree condicional, como se observa en la Figura 1.9.

3. Repetir los pasos anteriores de manera recursiva hasta que el FP-Tree condicional actual tenga un solo elemento.

Una vez hecho esto, se repite el proceso de manera recursiva, construyendo un nuevo FP-Tree condicional a partir de las ramas de prefijo para cada uno de los ítems frecuentes presentes en los patrones condicionales hasta que el FP-Tree construido en la recursión actual sea vacío.

Siguiendo con el ejemplo, el FP-Tree condicional resultante, que se observa en el extremo derecho de la Figura 1.9, esta compuesto de los conjuntos de ítemes frecuentes $\{y\}$, $\{x\}$ y $\{z\}$; cada uno de ellos con un contador total asociado de 3. Para el paso recursivo, lo que se

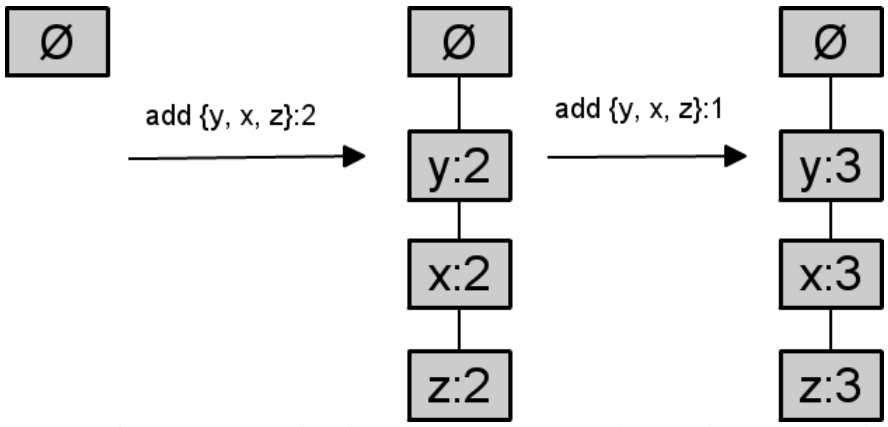

Figura 1.9: Proceso de construcción de un FP-Tree condicional a partir del conjunto de patrones condicionales $\{y, x, z\}$:2 y $\{y, x, z\}$ [Harrington, 2012].

hace es seleccionar uno de estos conjuntos y realizar una unión con el conjunto original, $\{t\}$. De esta forma, si seleccionamos $\{z\}$, el conjunto resultante será $\{z, t\}$. Luego, procedemos a agregar este conjunto a nuestra colección de conjuntos frecuentes. Y una vez hecho esto, procedemos a extraer el conjunto de patrones condicionales de $\{z\}$ en el FP-Tree condicional. Al hacerlo se obtiene sólamente el conjunto $\{y, x\}$:2. A partir de este conjunto se vuelve a construir un FP-Tree condicional como en el paso anterior. Esta recursión se repite hasta que el FP-Tree generado sea vacío.

Una vez finalizadas las recursiones sobre todos los conjuntos de ítemes de un elemento, se procede a retornar la colección de conjuntos frecuentes resultante; en la cual fuimos guardando cada uno de los conjuntos frecuentes que encontramos en el proceso recursivo.

La complejidad teórica del algoritmo *FP-Growth* resulta ser más difícil de expresar analíticamente que la del algoritmo *Apriori*, ya que depende tanto del número de ítemes frecuentes, su frecuencia y de la profundidad del FP-Tree generado en cada paso, entre otros. Esto último, a su vez, depende de las características internas de los datos más bien que solamente de la cantidad de estos. Junto con esto, el algoritmo realiza operaciones de diverso tipo, como comparaciones, recorrer nodos, inicializar estructuras de datos, y no solamente recorrer los datos originales [Kosters et al., 2003].

1.2.3. Otros algoritmos, implementaciones y aplicaciones

Posteriormente, Agrawal et al. presentaron el algoritmo *AprioriTid*, cuyas mejores características fueron combinadas con el algoritmo *Apriori* para crear el algoritmo *AprioriHybrid*, de orden de complejidad lineal en el número de transacciones [Agrawal et al., 1994]. Luego se han realizado más desarrollos en ARL orientado a transacciones secuenciales de clientes de puntos de ventas [Agrawal and Srikant, 1995].

Savasere et al. introdujeron el algoritmo *Partition* [Savasere et al., 1995] con el fin de extraer reglas de asociación en base de datos, el cual presenta reducciones en las operaciones de la CPU y de entrada/salida, y que además facilita la paralelización. Posteriormente se creó el algoritmo *Dynamic Itemset Counting (DIC)* [Brin et al., 1997b], que realiza menos

lecturas sobre los datos que los algoritmos previos, y que utiliza la métrica de *Convicción* a la hora de generar reglas de asociación. Luego, Park et al. presentaron un algoritmo que hace uso de funciones de Hashing con el fin de generar reglas candidatas [Park et al., 1995]. Se han realizado, también, adaptaciones de los algoritmos previos con el fin de realizar ARL en datos de tipo cuantitativo [Srikant and Agrawal, 1996].

Esfuerzos posteriores se han realizado con el fin de profundizar en los fundamentos teóricos subyacentes en ARL (e.g. definiendo el conjunto de posibles ítems como una estructura algebráica llamada *retículo*) [Zaki and Ogihara, 1998], y con el fin de extender la noción de reglas de asociación a correlaciones [Brin et al., 1997a] y taxonomías [Srikant and Agrawal, 1996].

Luego de esto, se han hecho numerosas implementaciones y optimizaciones a los algoritmos más utilizados en ARL, como, por ejemplo, el algoritmo Apriori [Bodon, 2010]; así como implementaciones que facilitan el mantener la privacidad de cada una de las fuentes de datos que participan en el proceso [Evfimievski et al., 2004].

Desde su concepción, el método de ARL ha sido aplicado en numerosas áreas, tales como la detección de intrusiones [Lee and Stolfo, 2000] y anomalías [Patcha and Park, 2007] [Chandola et al., 2009], educación [Romero and Ventura, 2007] [Romero et al., 2008], química [Dehaspe et al., 1998], privacidad de datos [Ghinita et al., 2008], búsqueda en la web [Ferragina and Gulli, 2008], tráfico en redes [Estan et al., 2003], computación social [Li et al., 2008], búsqueda semántica [Cohen et al., 2007], biología [Kramer et al., 2001] [Carmona-Saez et al., 2007], salud [Karabatak and Ince, 2009] [Chaves et al., 2011], medios de comunicación [Davidson et al., 2010] [Kobilarov et al., 2009], y la investigación forense [Iqbal et al., 2013]. Junto con esto, se han realizado numerosas investigaciones sobre el estado actual de ARL y sus posibles desarrollos a futuro dentro del marco de métodos automatizados de generación de conocimiento [Han et al., 2007].

Si bien existen numerosos esfuerzos por utilizar minería de datos y Machine Learning en diversos ámbitos de la astronomía (en particular, en detección, clasificación y caracterización de líneas moleculares en espectros de emisión [Škoda and Vážnỳ, 2011]), hasta la fecha no se ha propuesto abiertamente el uso de ARL sobre datos extraídos de espectros de frecuencia.

Sin embargo, se han realizado avances en ampliar los conceptos subyacentes en ARL con el fin de aplicar el método en campos más diversos [Brin et al., 1997a]. Específicamente, una rama de investigación ha desarrollado lo que se denomina *Weighted Association Rule Learning* [Wang et al., 2000] [Cai et al., 1998]. Este método permite asociar medidas de interés arbitrario a priori a ciertos conjuntos de datos. Si bien esto hace que se pierdan propiedades de clausura que son útiles a la hora de generar algoritmos eficientes, permite trabajar con conjuntos de transacciones de los cuales a algunos se desea dar más relevancia, a la hora de generar reglas de asociación, que a otros.

Capítulo 2

Especificación del Problema

2.1. Descripción del problema

El problema a resolver, esencialmente, es el de encontrar reglas de asociacion entre líneas moleculares detectadas en espectros de frecuencia obtenidos a partir de observaciones astronómicas.

Para ello, se asume que las líneas ya han sido detectadas en los distintos espectros; vale decir, se sabe que están presentes y se conocen sus posiciones dentro del rango de frecuencias. En la práctica eso puede ser muy difícil de lograr, sobre todo en circunstancias donde potencialmente pueden existir una alta cantidad de líneas espectrales y estas pueden interferir unas con otras en la señal final, lo que se conoce como *blending*.

Sin embargo, no es necesario que todas las líneas se encuentren ya identificadas; vale decir, que se sepa a qué especie (átomo, molécula, etc.) se encuentran asociadas. Actualmente existen herramientas que son capaces de ajustar modelos físicos conocidos con anterioridad a datos espectrales con el fin de identificar las líneas en ellos presentes.

2.2. Requisitos de la solución y casos de uso

A continuación se enuncian los requerimientos del sistema:

1. **Obtener reglas de asociación entre líneas de emisión espectrales [esencial].** El sistema debe generar reglas de asociación entre líneas de emisión presentes en espectros, independientemente de si estos pertenecen a una misma o a distintas moléculas o átomos, o si no han sido aun identificadas.
2. **Permitir al usuario observar las reglas generadas, y desplegarlas a este ordenadas según distintas medidas de relevancia [esencial].**

3. **Permitir al usuario guardar las reglas de asociación generadas [esencial].**

Una vez extraídas las reglas de asociación, el usuario debe poder revisarlas y guardarlas para su revisión posterior.

4. **Permitir al usuario aplicar los mismos algoritmos de reglas de asociación a datos de diversas fuentes [esencial].**
   Se desea que el sistema de extracción sea lo más general posible, de modo tal de poder aplicarlo a datos de líneas espectrales extraídos de distintos *surveys*, bases de datos, sistemas de modelamiento y detección de líneas, entre otros.
5. **El sistema debe ser ejecutable en un ambiente de computación de alto rendimiento [deseable].**
6. **El sistema debe ser compatible con plataformas de observatorios virtuales [deseable].**
7. **Implementar una interfaz gráfica de usuario [opcional].**

2.2.1. Casos de Uso

En la Figura 2.1 se muestra un diagrama con los casos de uso preliminares del sistema a desarrollar, y a continuación se describen estos en detalle junto con sus actores.

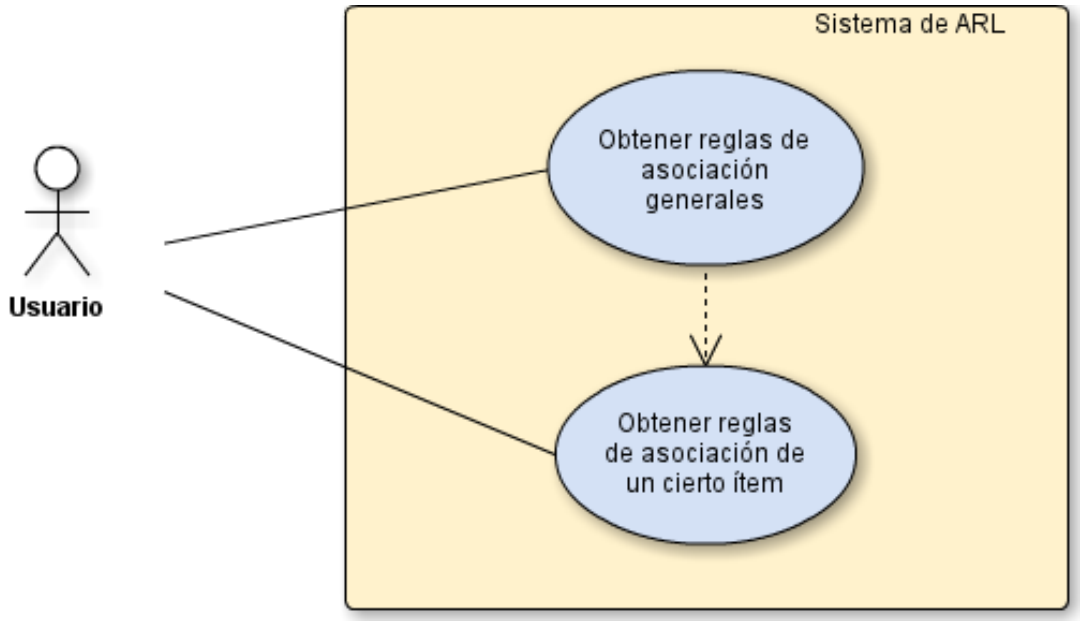

Figura 2.1: Diagrama de casos de uso del sistema.

Actores

Para este sistema existe solo un tipo de actor, dado que todos los usuarios finales tendrán acceso a las mismas funcionalidades. Este usuario será el encargado de seleccionar el conjunto de datos que quiere ingresar al sistema, en forma de transacciones de líneas moleculares. Cada transacción poseerá las líneas identificadas en un espectro en particular. Este usuario ingresará estos datos al sistema y luego seleccionará los parámetros de detección de reglas

que desee. Una vez ejecutados los algoritmos correspondientes, el usuario podrá observar las reglas generadas y, si así lo desea, ajustar nuevamente los parámetros para obtener mejores resultados sobre el mismo conjunto de datos.

Desde un punto de vista práctico, el usuario objetivo posee conocimientos técnicos sobre espectroscopía, sabe hacer uso de un terminal o línea de comandos, y puede manejar tablas en formato de valores separados por comas (CSV).

Descripción de casos de uso

En la siguiente tabla se muestra una descripción detallada de los casos de uso y se indica, de ser así, a qué requerimiento está asociado.

ID	Caso de uso	Descripción	Tipo	Ref.
1	Obtener reglas de asociación generales	El usuario obtiene reglas de asociación extraídas a partir de un conjunto de transacciones de líneas espectrales y las filtra u ordena mediante soporte, confianza o *lift*	Esencial	1,2,3,4
2	Obtener reglas de asociación de un cierto ítem	El usuario obtiene reglas de asociación extraídas a partir de un conjunto de transacciones de líneas espectrales, selecciona solo aquellas que posean un cierto ítem en su antecedente y/o consecuente, y las ordena mediante soporte, confianza o *lift*.	Esencial	1,2,3,4

Capítulo 3

Descripción de la Solución

A continuación se describe la solución implementada para el presente proyecto. Se detalla aquí la estructura, diseño y funcionamiento del sistema y la aplicación realizados con el fin de cumplir con los requerimientos descritos anteriormente.

3.1. Arquitectura de software

Dado que, para fines del proyecto, se requería de una herramienta con la cual se puediese llevar a cabo una serie de pruebas en distintos contextos, se optó por dividir el sistema en dos paquetes distintos; cada uno con una función específica, e interfaces bien definidas, con el fin de facilitar su posterior extensión y reutilización. En la Figura 3.1 se muestra un diagrama con la arquitectura general del sistema.

A continuación se detallan sus paquetes, módulos, e interfaces y explica sus funciones.

3.1.1. Paquete de Association Rule Learning (ARL)

El paquete de *Association Rule Learning (ARL)* es el encargado de realizar el aprendizaje mediante reglas de asociación en sí; vale decir, de recibir un conjunto de datos con transacciones y de retornar reglas de asociación generadas a partir de aquel conjunto.

En las siguientes secciones se especifican los formatos de entrada y salida de este paquete junto con una descripción de los módulos que lo componen.

Módulo de interfaz de usuario/controlador

El módulo de interfaz de usuario y controlador es el encargado de recibir directamente del usuario los parámetros de entrada correspondientes. Este módulo contiene métodos, clases

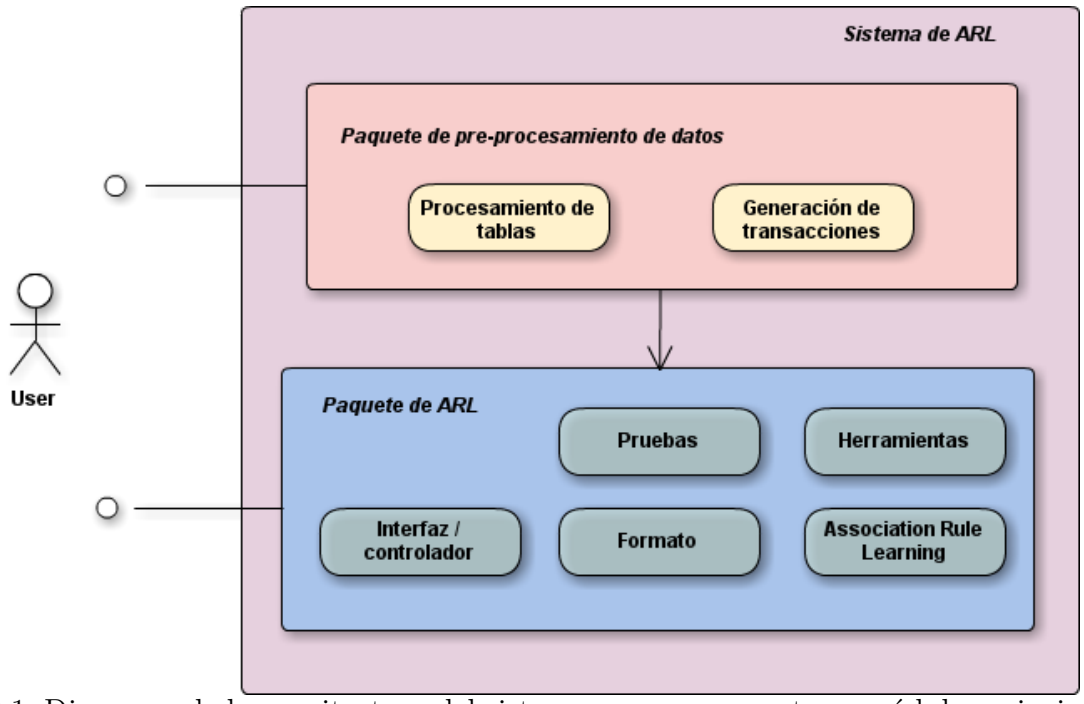

Figura 3.1: Diagrama de la arquitectura del sistema, con sus paquetes y módulos principales.

y funciones que reciben los parámetros del usuario, abren y leen los archivos de entrada adecuados, los procesan de acuerdo al formato especificado, y hacen entrega de los datos al módulo principal de ARL.

Este módulo es el encargado, además, de recibir las reglas de asociación y de entregarlas al módulo de formato para luego retornarlas al usuario en un archivo correspondiente.

Módulo de formato

Es el módulo encargado de analizar los archivos de entrada leídos por el módulo de interfaz de usuario, extraer la información pertinente de ellos según el formato especificado, y retornar los datos en una estructura adecuada para luego ser procesados por el módulo principal de ARL. A su vez, este módulo realiza, además la labor inversa; vale decir, recibe las reglas de asociación en una estructura de datos estándar para luego entregarlas al módulo de interfaz en el formato requerido por el usuario.

Hasta el momento los formatos soportados son valores separados por coma, o *comma separated values (CSV)* para archivos de entrada, y CSV o tabla en formato LaTeX para archivos de salida.

Módulo principal de ARL

El módulo principal de ARL es el encargado de llevar a cabo el algoritmo de aprendizaje mediante reglas de asociación en sí. En su parte lógica, consta de dos sub-módulos principales.

El primero es es sub-módulo encargado de extraer los conjuntos de ítems frecuentes; vale decir, aquellos que cumplen con el requerimiento de soporte mínimo. Y el segundo es el sub-módulo de generación de reglas, que es el encargado de recibir los conjuntos de ítems frecuentes y generar, a partir de ellos, las reglas de asociación que cumplen con el requerimiento de confianza mínima indicado.

Módulo de pruebas

Se encuentra dentro de este paquete, además, un módulo de testeo de los algoritmos de ARL sobre datos de prueba de pequeña envergadura; con el fin de realizar chequeos periódicos del funcionamiento correcto de estos algoritmos en la medida que se realizan cambios, mejoras o refactorizaciones sobre su código fuente.

Módulo de herramientas

Finalmente, se encuentra el módulo de herramientas generales, que consta de una serie de funciones de uso frecuente por parte de otros módulos del paquete; tales como operaciones sobre listas anidadas, búsqueda de llaves sobre diccionarios específicos, entre otros.

3.1.2. Paquete de procesamiento de datos

Debido a que en la mayoría de las ocasiones los datos sobre los cuales se desea aplicar los algoritmos de reglas de asociación no se encuentran en los formatos o estructuras necesarias, se procedió a implementar un paquete de pre-procesamiento. Este contiene una serie de scripts y métodos cuya función principal es extraer los datos desde sus fuentes originales, opcionalmente inferir aquella información que sea relevante, y guardarla en archivos cuyo formato sea comprensible para el paquete de aprendizaje de reglas de asociación.

En su implementación actual, este paquete se encuentra enfocado, en su mayor parte, en trabajar sobre datos extraídos a partir del Sloan Digital Sky Survey (SDSS).

A continuación se enumeran algunos de sus componentes más importantes.

Queries SQL

Una colección de queries relevantes para ejecutar en las bases de datos de SDSS y extraer los datos sobre los cuales obtener las reglas de asociación.

Módulo de procesamiento de tablas

Contiene una serie de scripts cuyo fin es recibir un archivo de tabla de base de datos en formato CSV y procesar los datos que contiene; por ejemplo, eliminando ciertas filas, añadiendo columnas calculadas a partir de las ya existentes, entre otros. Los resultados son guardados en un nuevo archivo de tabla en formato CSV.

Módulo de generación de transacciones

Este módulo contiene scripts cuya función es recibir un archivo de tabla de base de datos en formato CSV, y a partir de él generar un archivo CSV que contenga una transacción por cada fila; cada una de estas con una lista de ítems en formato adecuado para ser recibido por el paquete de ARL.

3.2. Diseño de clases

En la Figura 3.2 se observa un diagrama con las clases más importantes dentro del paquete de Association Rule Learning y sus relaciones. Estas son las encargadas abstraer la implementación de las distintas funciones y estructuras de datos de las cuales hacen uso los algoritmos de ARL tal como se describieron con anterioridad en el marco teórico.

A continuación se detallan las clases de objetos más importantes del sistema.

3.2.1. Clase *ItemSet*

La clase *ItemSet* es la encargada de mantener información sobre un conjunto de ítems y abstraer su estructura de datos subyacente. Cada instancia de esta clase corresponde a un conjunto de ítems distinto, y contiene campos que guardan la información más reciente sobre su soporte (calculado sobre un cierto conjunto de transacciones) y punteros a meta-datos con información adicional sobre los ítems en sí. Su interfaz asegura que se pueda realizar de forma adecuada, visto desde un punto de vista matemáticamente abstracto, las operaciones más comúnes de conjuntos de elementos; como comprobar pertenencia, sumar conjuntos, diferencia entre conjuntos, entre otros.

3.2.2. Clase *AssociationRule*

La clase *AssociationRule* es la que define la estructura y comportamiento de las reglas de asociación. Cada instancia de esta clase corresponde a una regla de asociación en particular, extraída a partir de un cierto conjunto de datos. Cada regla de asociación consta de dos objetos de la clase *ItemSet*; uno para el antecedente y otro para el consecuente de la regla.

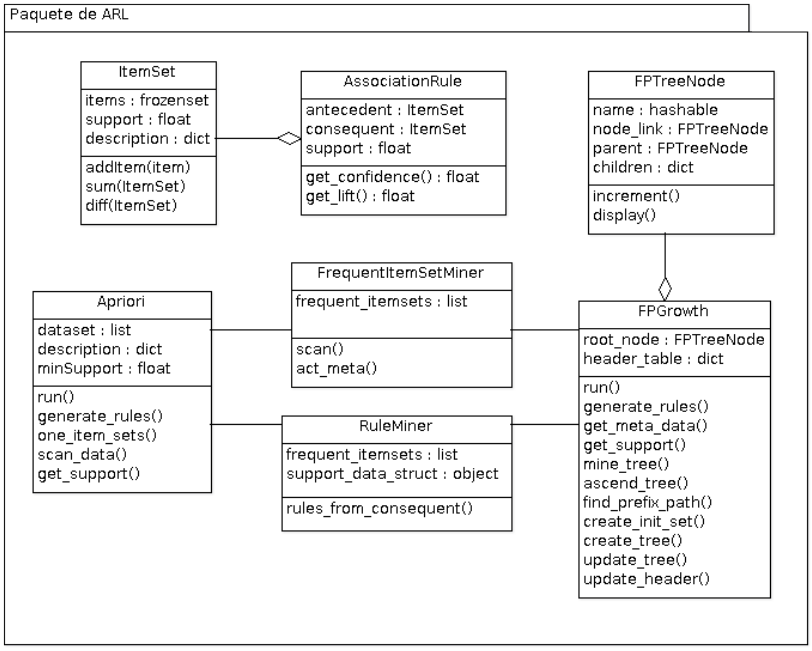

Figura 3.2: Diagrama de clases más importantes del paquete de ARL.

Además contiene un campo que codifica su soporte, junto con métodos para calcular sus medidas de relevancia, tales como su confianza y lift.

3.2.3. Clase *FrequentItemSetMiner*

La clase *FrequentItemSetMiner* es la encargada de abstraer y guardar información sobre el proceso de extraer a partir de las transacciones aquellos conjuntos de ítems que cumplan con un requisito de soporte mínimo dado. Cada instancia de esta clase corresponde a un proceso de extracción distinto, conteniendo campos y estructuras de datos para los algoritmos involucrados, su estado actual y su resultado.

En su implementación actual, esta clase es heredada por dos sub-clases. Una correspondiente al algoritmo *Apriori*, y otra al algoritmo *FP-Growth*. Cada una contiene su propia implementación de los métodos principales, definidos en su clase padre, junto con sus propias funciones auxiliares y estructuras de datos correspondientes.

3.2.4. Clase *RuleMiner*

La clase *RuleMiner* es la que abstrae el proceso de generar de asociación a partir de conjuntos frecuentes de ítems encontrados por los algoritmos de extracción de ítems frecuentes (*Apriori* y *FP-Growth*. Cada instancia de esta clase corresponde a un proceso de extracción distinto; básicamente el mismo en todo los casos salvo en ciertos detalles, como algunas funciones auxiliares y referencias a estructuras de datos, dependiendo de si los conjuntos fueron extraídos mediante uno u otro algoritmo.

3.3. Detalles de implementación

La implementación del sistema se llevó a cabo en el lenguaje de programación Python [pyt, 2014]. Se realizó una implementación propia de los algoritmos antes descritos, con algunas adaptaciones para su funcionamiento correcto en el contexto de este proyecto; y se hizo uso de paquetes externos con el fin de hacer más simple el manejo de archivos CSV y la implementación de la interfaz por línea de comando.

3.3.1. Extracción de conjuntos de ítems frecuentes

Como se dijo anteriormente, para la extracción de conjuntos de ítems frecuentes se procedió a realizar la implementación de los algoritmos *Apriori* y *FP-Growth*. Ambos algoritmos reciben las transacciones en una misma estructura de datos y retornan los conjuntos frecuentes también en una misma estructura en ambos casos. Pero cada una de estas clases posee sus propios métodos, definidos por los algoritmos en general.

En general, para ambos algoritmos la estructura de datos más utilizada para la implementación subyacente en los objetos correspondientes a conjuntos frecuentes, candidatos, antecedentes y consecuentes por igual, fue la de *frozensets*. Esta clase de objetos, además de permitir las operaciones matemáticas de conjuntos clásicas, tales como sumas y diferencias de conjuntos, permite que a los objetos se les pueda aplicar una función de *hashing*; y, por lo tanto, utilizar los conjuntos como llaves de diccionario en forma de tablas de hash, y de esta forma, por ejemplo, indexar por conjunto distintas estructuras de datos auxiliares.

3.3.2. Extracción de reglas de asociación

La extracción de reglas de asociación a partir de conjuntos frecuentes se llevó a cabo mediante una implementación del algoritmo *Apriori* de generación de reglas. El sistema retorna al usuario reglas de asociación que cumplan con las medidas mínimas de soporte y confianza que él requiera. Estas serán mostradas en orden decreciente de soporte, confianza o *lift*, según se requiera. Además, el sistema permite al usuario mostrar sólamente aquellas reglas en las que esté presente un cierto ítem en el antecedente o en el consecuente de ellas.

Entrada y salida

El paquete de *Association Rule Learning (ARL)* recibe como entrada un archivo de tabla en formato de valores separados por coma o *comma separated values (CSV)*. Este archivo debe tener el siguiente formato en cada una de sus filas

```
<TID>,"<ItemList>"
```

donde *<TID>* es el identificador de la presente transacción, e *<ItemList>* es una lista de identificadores únicos de los ítems presentes en la transacción separados por comas. Tal como se indica, esta lista debe ir rodeada por comillas dobles en el archivo de entrada. A continuación se muestra un ejemplo de archivo de entrada válido.

```
000001,"15,2,44"
000002,"5,4,23,67,43,234"
000003,"66,3,53,23"
```

Adicionalmente, se puede especificar para cada transacción un tipo o clase a la que pertenece, o de la cual se origina, con el fin de realizar estadísticas pertinentes con las reglas generadas. De ser así, el archivo de entrada debe tener el siguiente formato en cada una de sus filas,

```
<TID>,<Class>,"<ItemList>"
```

donde, en esta ocasión, se añade en la segunda posición el campo *<Class>*, que consiste en una secuencia de caracteres válidos que identifique de manera unívoca la clase a la cual la transacción pertenece. A continuación un ejemplo de entrada válida en este formato.

```
000001,MORNING,"15,2,44"
000002,MORNING,"5,4,23,67,43,234"
000003,NIGHT,"66,3,53,23"
```

Esta lista es leída y procesada dentro del paquete de ARL y luego entregada en una estructura de datos correspondiente al algoritmo indicado, que obtendrá las reglas de asociación presentes en el conjunto de transacciones. Estas reglas, por defecto, serán retornadas en un archivo de texto en formato CSV con la siguiente estructura en cada una de sus líneas.

```
<N>,"<Antecedent>","<Consequent>",<Support>,<Confidence>,<Lift>
```

Donde N es un número identificador de la regla de asociación, *<Antecedent>* es una lista de ítems separados por coma correspondientes al antecedente de la regla, *<Consequent>* es una lista de ítems separados por coma correspondientes al consecuente de la regla, *<Support>* es un valor de punto flotante entre 0 y 1 correspondiente al soporte de la regla, *<Confidence>* es un valor de punto flotante entre 0 y 1 correspondiente a la confianza de la regla, y *<Lift>* es un valor de punto flotante entre mayor o igual a 0 correspondiente al lift de la regla. A continuación un ejemplo de este formato de archivo de salida.

```
1,"15,33","2,89,91",0.21,0.85,2.31
2,"12,33,44","5,23,31",0.23,0.81,3.3
```

Si, además, en los datos de entrada se especificó una clase para cada transacción, entonces el archivo de salida tendrá el siguiente formato

```
<N>,"<Antecedent>","<Consequent>",<Support>,<Confidence>,
 <Lift>,"<ClassCount>"
```

en donde *<ClassCount>* es una lista de valores separados por comas con el siguiente formato

```
<Class01>:<Count01>,<Class02>:<Count02>,...
```

donde *<Class01>* es el identificador de la primera clase, *<Count01>* es un número entero que indica cuántas de las transacciones que satisfacen la regla actual pertenecen a esta primera clase, y así sucesivamente con todas las clases posibles. A continuación un ejemplo de archivo de salida con el formato recién descrito.

```
1,"15,33","2,89,91",0.21,0.85,2.31,"MORNING:210,NIGHT:15"
2,"12,33,44","5,23,31",0.23,0.81,3.3,"MORNING:20,NIGHT:91"
```

3.4. Interfaz de usuario

La interfaz del usuario con el paquete principal de ARL y con los scripts del paquete de pre-procesamiento de datos, se realiza mediante un terminal o línea de comandos. Los parámetros, con los cuales se invoca cada uno de estos, siguen la sintaxis estándar *de facto* de la mayoría de los sistemas tipo UNIX. En la implementación de estas interfaces se priorizó la claridad de las instrucciones por sobre lo conciso de estas, y se favorece la escritura de resultados a archivo; haciendo uso de la salida estándar solo en casos de errores y avisos del funcionamiento del sistema.

Cada uno de los archivos de entrada o interfaz de los módulos puede ser invocado con el parámetro *-h* y se desplegará un texto de ayuda con los parámetros disponibles y sus funcionalidades.

Por ejemplo, el archivo de entrada del paquete de ARL, llamado *spelar.py*, tiene la siguiente sintaxis de invocación:

```
spelar.py [-h] [-d DESCRIPTIONS] [-l LATEX | -c CSV] (-ap | -fp)
 [-m MAX] [--by_supp] [--by_conf] [--by_lift] [--in_ant ITEM]
 [--in_con ITEM] in_file min_supp min_conf
```

[1]

Donde las opciones son:

- `-h`: Desplegar texto de ayuda.

[1]Parámetros rodeados por corchetes son opcionales. La barra vertical indica parámetros mutuamente excluyentes entre sí.

- `-d`: Permite especificar la ubicación de un archivo en formato CSV (`DESCRIPTIONS`) que contenga una descripción para cada identificador de ítem, para mostrar en las reglas resultantes y así ayudar a hacer más clara su semántica al usuario. Un ejemplo de archivo de descripciones es el siguiente:

  ```
  id,description
  1857,AlIII_1857
  8500,CaII_8500
  8544,CaII_8544
  8665,CaII_8665
  ```

- `-l`: Permite especificar la ubicación de un archivo (`LATEX`) en el cual escribir en formato LaTeX las reglas extraídas.
- `-c`: Permite especificar la ubicación de un archivo (`CSV`) en el cual escribir en formato CSV las reglas extraídas.
- `-ap`: Utilizar algoritmo Apriori para generar conjuntos frecuentes.
- `-fp`: Utilizar algoritmo FP-Growth para generar conjuntos frecuentes.
- `-m`: Permite especificar un número máximo `MAX` de reglas a retornar.
- `--by_supp`: Desplegar reglas ordenadas por soporte.
- `--by_conf`: Desplegar reglas ordenadas por confianza.
- `--by_lift`: Desplegar reglas ordenadas por lift.
- `--in_ant`: Mostrar sólo las reglas que posean el ítem `ITEM` en su antecedente.
- `--in_con`: Mostrar sólo las reglas que posean el ítem `ITEM` en su consecuente.
- `in_file`: Archivo de entrada en formato CSV con las transacciones.
- `min_supp`: Soporte mínimo de las reglas a extraer. Valor de punto flotante entre 0 y 1.
- `min_conf`: Confianza mínima de las reglas a extraer. Valor de punto flotante entre 0 y 1.

Capítulo 4

Validación de la Solución

4.1. Validación mediante simulaciones

Una vez realizada la implementación del sistema y de los algoritmos de ARL, se realizó una serie de pruebas con datos simulados. Estos fueron generados de manera aleatorea, para así tener una base con la cual comparar el desempeño en la práctica de los algoritmos con datos reales; los cuales se alejan más de la aleatoriedad absoluta al poseer una estructura más definida, grupos de ítems comunes entre las transacciones, entre otros.

Para generar estos datos, en primer lugar se procedió a definir un universo posible de ítems para ser incluídos en cada una de las transacciones. En particular, se seleccionó un subconjunto n de ítems a partir del mismo universo de ítems presentes en los datos reales (ver tabla A.1). Luego, se procedió a generar las transacciones seleccionando aleatoreamente ítems a partir de aquel conjunto de la siguiente manera. Cada uno de estos n ítems tiene una probabilidad p de estar presente en una transacción dada, y una probabilidad $(1-p)$ de no estar presente en esta. Por lo tanto, la presencia de un cierto ítem en una transacción fija cualquiera es, esencialmente, lo que se conoce como un *ensayo de Bernoulli*, y, por ende, la probabilidad de que una transacción dada tenga una cantidad k de ítems puede expresarse con una función de distribución probabilística de la forma

$$\mathbb{P}[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

con $0 \leq k \leq n$, la que se conoce como *distribución binomial*, cuyo valor esperado corresponde a np y su varianza a $np(1-p)$.

Utilizando este método, se generó aleatoreamente conjuntos de 100.000 transacciones tomando ítems a partir de un subconjunto de tamaño $n = 10$ del universo de ítems. Se procedió entonces a realizar una serie de pruebas de los algoritmos *Apriori* y *FP-Growth* con estos datos, primero dejando fijo el soporte mínimo y variando la probabilidad p de los ítems de estar presentes en cada transacción, y posteriormente dejando fija este p y variando el soporte

mínimo de ambos algoritmos.

En la Figura 4.1 se observa los tiempos de ejecución en promedio al dejar fijo un soporte mínimo de 0.15 para los algoritmos, para $p \in 0{,}2, 0{,}3, \ldots, 0{,}8$, repitiendo el experimento 5 veces. Se observa que para $p \leq 0{,}6$ el algoritmo *FP-Growth* posee un desempeño levemente mejor que el del algoritmo *Apriori*, para luego ser superado con creces por este último cuando $p > 0{,}6$. Esto puede deberse, principalmente, a que al estar presente la mayoría de los ítems con alta probabilidad, estos se repiten mucho más de una transacción a otra, y el utilizar una estructura de datos tipo arbol deja de ser una ventaja, aproximándose más a una simple lista enlazada.

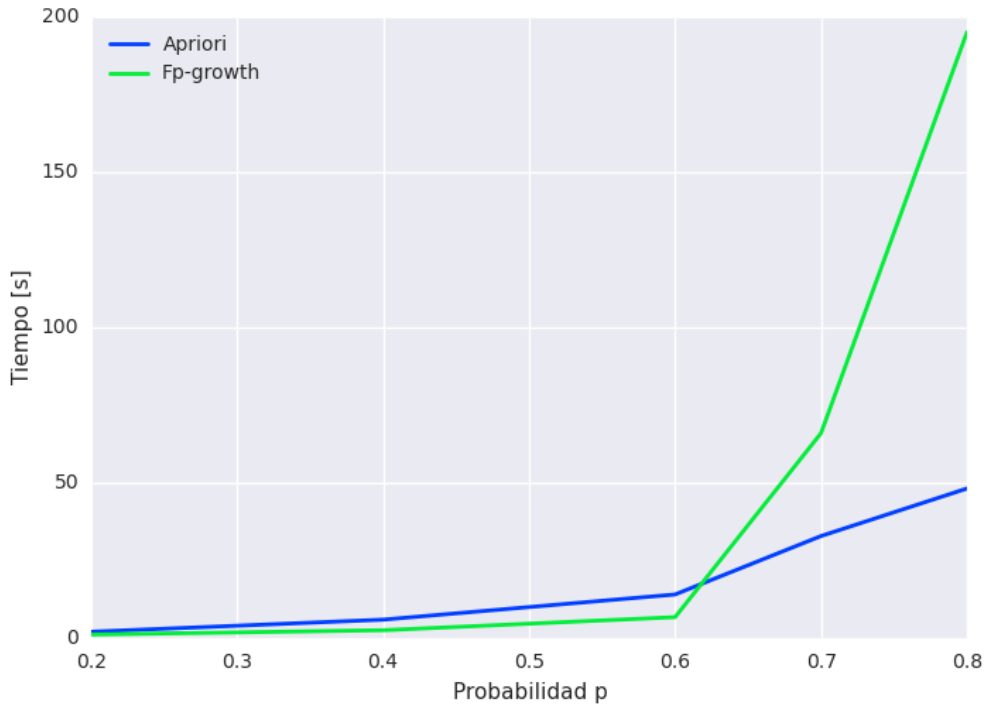

Figura 4.1: Grafico de tiempos de ejecución de algoritmos *Apriori* y *FP-Growth* sobre datos simulados; utilizando soporte mínimo 0.15 y distintas probabilidades de presencia de ítems en una transacción dada.

Un fenómeno similar, pero en forma inversa, puede apreciarse en la Figura 4.2, que muestra los tiempos de ejecución en promedio de 5 repeticiones del experimento de generar conjuntos de 100.000 transacciones con un valor $p = 0{,}8$ y ejecutar sobre estos los algoritmos de ARL con distintos soportes mínimos. Al igual que en el caso anterior, lo más probable es que esto ocurra debido a que al utilizar un bajo soporte, se considera una mayor cantidad de ítems en la generación de conjuntos frecuentes, y estos se repiten mucho más de un conjunto a otro, lo cual hace perder considerable ventaja al algoritmo *FP-Growth* con respecto a *Apriori*.

Una vez obtenidos estos resultados referenciales, se procedió a realizar la validación de los algoritmos sobre conjuntos de datos reales.

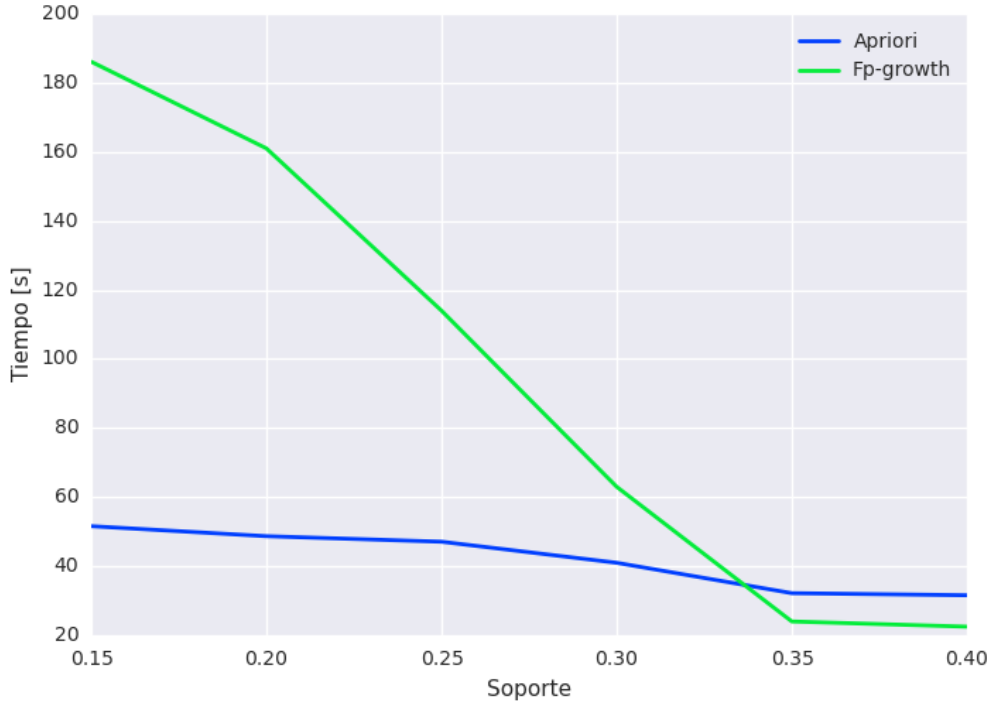

Figura 4.2: Grafico de tiempos de ejecución de algoritmos *Apriori* y *FP-Growth* sobre datos simulados; utilizando soporte mínimo 0.15 y distintas probabilidades de presencia de ítems en una transacción dada.

4.2. Antecedentes de datos de prueba reales

Una vez lista la implementación de la mayor parte del sistema y los algoritmos de ARL, se procedió a realizar una prueba de concepto con datos reales. Si bien el objetivo ideal sería aplicar estos algoritmos a datos de observaciones en bandas de baja frecuencia (como las de bandas de radio de *ALMA*) donde existe una gran cantidad de líneas presentes por cada espectro, se decidió realizar la prueba de concepto de este proyecto sobre datos del *Sloan Digital Sky Survey (SDSS)* por las siguientes razones:

1. Si bien el universo de líneas presentes en cada espectro es bastante reducido (48 líneas), la mayoría de estas se encuentran bien identificadas.
2. Las líneas presentes en el espectro visible son bien conocidas, y en general se posee información completa sobre sus características, tales como su temperatura.

Ahora bien, hubo que tener en mente de forma constante que se está trabajando con un universo reducido de ítems (líneas espectrales) al momento de analizar los resultados de estas pruebas.

Para acceder a los datos de *SDSS* se utilizó la interfaz web del sistema *CasJobs*, que recibe consultas en lenguaje SQL y guarda los resultados en una base de datos asociada a la cuenta del usuario. En particular se hizo uso de los datos del *data release 7 (DR7)*, que es el último en contener tablas con información específica sobre las líneas espectrales.

En particular, se utilizó dos tablas pertenecientes al DR7: *SpecObj* y *SpecLineAll*. La tabla *SpecObj* contiene información de los objetos astronómicos sobre los cuales se ha realizado mediciones espectroscópicas. De esta tabla se extrayeron los siguientes campos:

- **specObjID**: Identificador del objeto astronómico.
- **zStatus**: Flag que indica el método mediante el cual se calculó el *redshift* del objeto.
- **objTypeName**: El tipo de objeto (e.g. galaxia, estrella, quasar), determinado mediante imágenes.
- **specClass**: El tipo de objeto, determinado mediante su espectro.
- **mag_0**. **mag_1** y **mag_2**: Magnitud de emisión en tres frecuencias distintas.
- **z**: *Redshift* o corrimiento al rojo del objeto debido al efecto Doppler.
- **zErr**: Error de *Redshift* del objeto.

A su vez, la tabla *SpecLineAll* contiene información sobre cada una de las líneas presentes en cada uno de estos objetos. De esta tabla se extrayeron los campos:

- **SpecLineID**: Código identificador único de línea espectral.
- **wave**: Posición central de la línea espectral observada, en longitud de onda (Armstrongs), dentro del espectro.
- **waveErr**: Error en la posición central de la línea espectral.
- **restWave**: Posición central de la línea espectral teórica o medida en laboratorio.
- **lineID**: Identificador de línea espectral (identifica una línea de una especie en particular).
- **category**: 1 si la línea se detectó mediante el uso de ajuste de modelos luego de aplicar un filtro (o *transformada wavelet*) con el fin de determinar el *redshift* de las líneas de emisión y 2 si la línea se detectó una vez que el objeto fue clasificado y su *redshift* determinado.
- **height**: Altura de la función gaussiana ajustada a la línea.
- **heightErr**: Error de la función gaussiana ajustada a la línea.
- **ew**: Ancho equivalente de la línea. Es una medida del área integrada entre la línea espectral y el contínuo a longitudes de onda adyacentes. Indica el brillo o intensidad normalizada de la línea espectral.
- **ewErr**: Error del ancho equivalente.
- **z**: *Redshift* de la línea[1].
- **zErr**: Error de *redshift*.

Ahora bien, la tabla *SpecObj* del DR7 de SDSS posee en total de 1.053.144 filas. Esto indica que aquel *data release* contiene información espectroscópica de más de un millón de objetos. Cabe recalcar que el caso general del sistema de ARL aplicado a lineas espectrales asume que cada transacción corresponde a una observación o lectura de un espectro de frecuencias; y, por tanto, varios espectros pueden estar asociados a un mismo objeto astronómico. Sin embargo, dado que para el caso de los datos de SDSS puede que las líneas pertenecientes a cada objeto se hayan obtenido en diversas observaciones, se tomará cada **objeto** como una transacción,

[1]Si es distinto la redshift del objeto en la tabla *SpecObj* entonces la línea está mal identificada

y no la observación particular de un objeto. Por lo tanto, al hacer una operación *JOIN* entre las tablas *SpecObj* y *SpecLineAll*, se obtendrá la lista de todas las líneas espectrales con información del objeto astronómico del cual provienen. La idea es, entonces, utilizar cada uno de los objetos como una transacción, y las líneas asociadas a cada uno de ellos como sus ítems. Se utilizará el campo *lineID* de la tabla *SpecLineAll* como identificador de cada uno de estos ítems; dado que dos líneas asociados a distintos objetos pueden tener el mismo valor en *lineID*, cosa que no ocurre con el identificador único **SpecLineID**.

En efecto, existe en el DR7 una tabla llamada *SpecLineNames*A.1 que enumera los 49 valores que puede tomar el campo *lineID*. Cada uno de estos corresponde a una línea de una especie en particular. Algunos de estos valores son:

Valor	**Nombre**
1857	AlIII_1857
8500	CaII_8500
8544	CaII_8544
8665	CaII_8665
1335	CII_1335
2326	CII_2326
...	...

A continuación se numeran los objetos de la tabla *SpecObj* según el tipo de objeto determinado mediante su espectro (campo *specClass*).

specClass	**Tipo de objeto**	**Número de objetos**
0	Desconocido	11566
1	Estrella	85564
2	Galaxia	807118
3	Cuasi-estelar (*quasar*)	94994
4	*Quasar* de alto *redshift*	7584
6	Estrella tardía	46318

4.3. Selección y pre-procesamiento de datos

Para fines de esta prueba de concepto y validación del sistema se escogió realizar la extracción de reglas de asociación a partir de objetos de tipo estelar (*specClass* 1 o 6), principalmente debido a que para objetos de este tipo el *redshift* en general debería ser más bajo y, aproximadamente, dentro del mismo rango; lo cual permite una muestra más uniforme de longitudes de onda de las líneas.

En total, existen 52.570.585 líneas asociadas a los 131882 objetos de tipo estelar presentes en el *data release 7*. Esto supone un claro problema técnico, dado que el sistema *CasJobs* no permite descargar tablas de tal envergadura. Por lo tanto, debe realizarse un proceso de selección lo más sistemático posible.

En primer lugar, se consideró el conjunto de 131.882 objetos de tipo estelar. En la Figura

4.3 se puede apreciar una selección del histograma del *redshift* de estos objetos. Se puede apreciar que la mayoría de los objetos se encuentran cercanos a 0 y unos pocos se encuentran distribuidos en valores mayores. Se decidió por tanto, eliminar estos objetos de mayor *redshift* (y por tanto más lejanos) con el fin de trabajar sólo con aquellos objetos más cercanos. Se decidió por utilizar solo los objetos que tengan un *redshift* menor que 0.002.

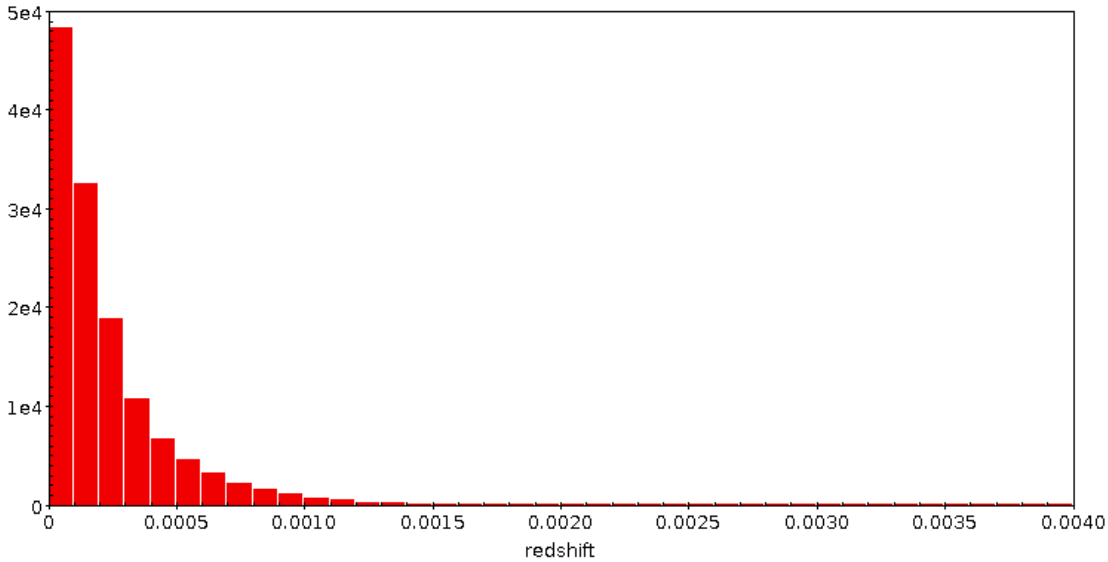

Figura 4.3: Histograma de *redshift* de objetos estelares.

Ahora bien, con el fin de reducir de forma más considerable el número de líneas a analizar, se decidió filtrar estas y dejar sólo las más brillantes. Para esto, se utilizó los valores de *ancho equivalente (ew)*[2] de cada una de las líneas, y se calculó una nueva medida a la que se denominó *razón señal a ruido (SNR)* que consiste en la razón entre el ancho equivalente y su error (*ewErr*). Para comprobar si este valor es un filtro efectivo del número de líneas, se tomó una muestra de 1 millón de líneas del total asociado a objetos estelares, y se produjo el histograma acumulativo de la Figura 4.4. Observando esta figura, se puede apreciar que, en efecto, esta nueva medida introducida es un parámetro efectivo de selección de líneas (cerca del 20 % de las líneas de la muestra tiene un SNR mayor que 5 y por lo tanto lo consideramos como detecciones fidedignas).

Seleccionando, del total de líneas asociadas a objetos estelares, aquellas que estén asociadas a objetos con redshift menor que 0.002 y que tengan un *SNR* mayor que 5, se obtiene un total de 1.189.817 líneas asociadas a 120.250 objetos estelares.

Sin embargo, algunas de estas líneas no poseen un identificador *lineID* y otras que sí lo poseen se encuentran mal identificadas. La razón de por qué ocurre esto se muestra en la Figura 4.5. Como ahí se puede apreciar, existe un gran número de líneas cuyo *redshift* (indicado por el campo **z** de la tabla *SpecLineAll*) tiene como valor -9999. Esto no tiene sentido alguno desde el punto de vista físico, e indica sencillamente un valor nulo o inexistente.

Incluso en muchas de las 979.173 líneas que resultan de filtrar aquellas que poseen valores nulos de *redshift*, este valor aún así no concuerda con el *redshift* del objeto; tomando el *redshift*

[2]El ancho equivalente es una medida del área integrada entre la línea espectral y el contínuo a longitudes de onda adyacentes. Indica el brillo o intensidad normalizada de la línea espectral.

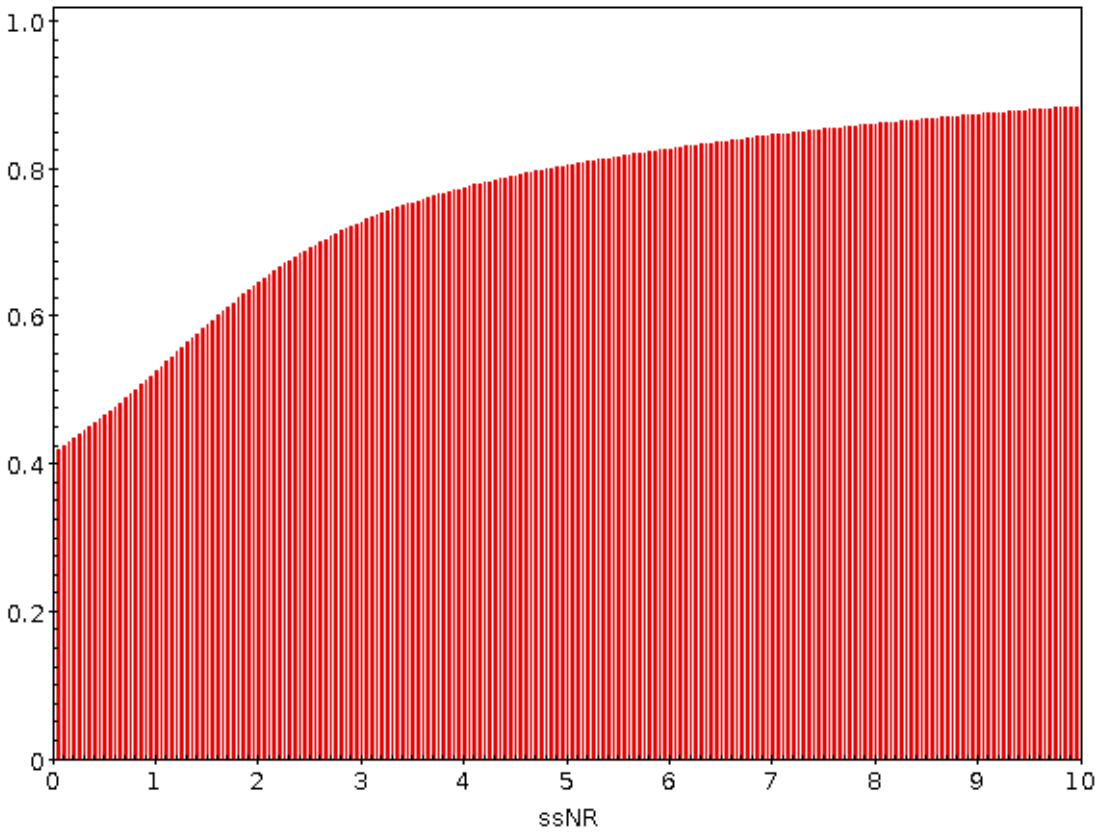

Figura 4.4: Histograma acumulativo de líneas asociadas a objetos estelares por su *SNR*.

de la línea valores que llegan hasta 5, cuando el del objeto correspondiente se encuentra mucho más cercano a 0, como se observa en la Figura 4.6

Dado que el identificador de línea *lineID* corresponde a una aproximación de la posición central de la línea espectral teórica o medida en laboratorio en Armstrongs (campo *restWave* de la tabla *specLineAll*) al entero más cercano, y que este último valor se calcula a partir de la posición central observada (campo *wave*) y el *redshift* de la línea (campo z), se entiende que si el valor de *redshift* no es el correcto, entonces finalmente el identificador de línea tampoco lo será.

Por eso, como parte del pre-procesamiento de los datos se prefirió, para aquellas líneas con *lineID* inexistente o *redshift* erróneo, volver a calcular un *restWave* a partir del *wave* utilizando el *redshift* del objeto en vez del de la línea; y de ahí asignarle un nuevo *lineID* resultante de aproximar el *restWave* al entero más cercano. El cálculo del *wave* de la línea a partir de su *restWave* y el *redshift* del objeto se llevó a cabo mediante la fórmula

$$\lambda_{restWave} = \frac{\lambda_{wave}}{1 + z_{obj}}$$

donde $\lambda_{restWave}$ corresponde al campo *restWave* de la línea, λ_{wave} a su *wave* y z_{obj} al *redshift* del objeto.

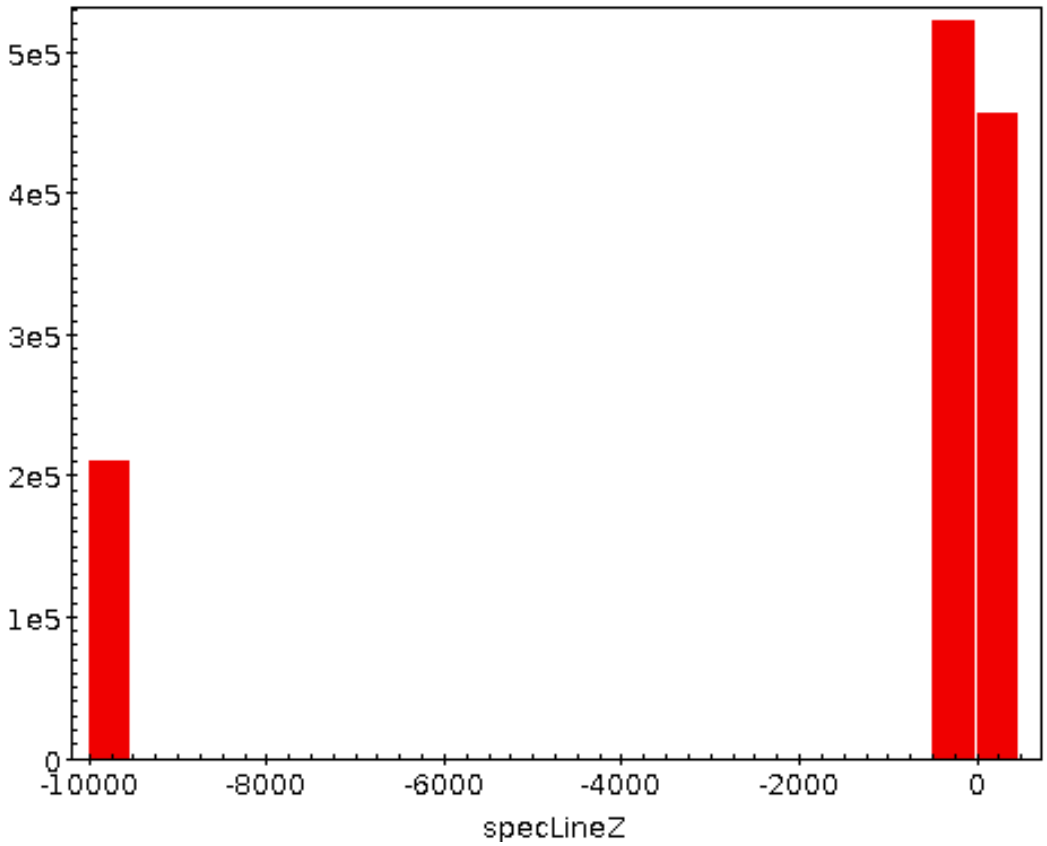

Figura 4.5: Histograma de *redshift* de las líneas espectrales seleccionadas.

4.4. Resultados

Al aplicar a los datos anteriores ya procesados el algoritmo de ARL, con un soporte mínimo de 0.15 y confianza mínima de 0.7, se produjo un total de 5181 reglas, generadas a partir de 576 conjuntos de ítems frecuentes. Las 25 reglas con mayor soporte se muestran en la siguiente tabla.

N	Rule	Supp	Conf	Lift
1	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 6565(Ha\_6565) \}$	0.41	0.90	1.69
2	$\{ 6565(Ha\_6565) \} \Rightarrow \{ 4863(Hb\_4863) \}$	0.41	0.77	1.69
3	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 4342(Hg\_4342) \}$	0.40	0.87	2.03
4	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 4863(Hb\_4863) \}$	0.40	0.93	2.03
5	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 3970(H\_3970) \}$	0.39	0.84	1.81
6	$\{ 3970(H\_3970) \} \Rightarrow \{ 4863(Hb\_4863) \}$	0.39	0.83	1.81
7	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 3970(H\_3970) \}$	0.37	0.87	1.88
8	$\{ 3970(H\_3970) \} \Rightarrow \{ 4342(Hg\_4342) \}$	0.37	0.80	1.88
9	$\{ 3970(H\_3970) \} \Rightarrow \{ 6565(Ha\_6565) \}$	0.37	0.79	1.49
10	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 6565(Ha\_6565) \}$	0.37	0.86	1.61
11	$\{ 4103(Hd\_4103) \} \Rightarrow \{ 4342(Hg\_4342) \}$	0.37	0.94	2.19
12	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 4103(Hd\_4103) \}$	0.37	0.86	2.19
13	$\{ 4103(Hd\_4103) \} \Rightarrow \{ 3970(H\_3970) \}$	0.36	0.92	1.99

14	$\{ 3970(H\_3970) \} \Rightarrow \{ 4103(H\mathrm{d}\_4103) \}$	0.36	0.78	1.99
15	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 4342(Hg\_4342), 6565(Ha\_6565) \}$	0.36	0.79	2.14
16	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 4863(Hb\_4863), 6565(Ha\_6565) \}$	0.36	0.84	2.04
17	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 3970(H\_3970), 4863(Hb\_4863) \}$	0.35	0.83	2.15
18	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 3970(H\_3970), 4342(Hg\_4342) \}$	0.35	0.77	2.07
19	$\{ 3970(H\_3970) \} \Rightarrow \{ 4342(Hg\_4342), 4863(Hb\_4863) \}$	0.35	0.76	1.92
20	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 4103(H\mathrm{d}\_4103) \}$	0.35	0.77	1.97
21	$\{ 4103(H\mathrm{d}\_4103) \} \Rightarrow \{ 4863(Hb\_4863) \}$	0.35	0.90	1.97
22	$\{ 4863(Hb\_4863) \} \Rightarrow \{ 3970(H\_3970), 6565(Ha\_6565) \}$	0.35	0.76	2.07
23	$\{ 3970(H\_3970) \} \Rightarrow \{ 4863(Hb\_4863), 6565(Ha\_6565) \}$	0.35	0.75	1.83
24	$\{ 4342(Hg\_4342) \} \Rightarrow \{ 4103(H\mathrm{d}\_4103), 4863(Hb\_4863) \}$	0.35	0.81	2.29
25	$\{ 4103(H\mathrm{d}\_4103) \} \Rightarrow \{ 4342(Hg\_4342), 4863(Hb\_4863) \}$	0.35	0.88	2.23

Como es de esperarse, las reglas con más soporte poseen pocos elementos tanto en su antecedente como en su consecuente. Los valores tanto de confianza como de *lift* observados dentro de este conjunto muestran que las líneas con alto soporte tienden a aparecer juntas en la mayoría de las ocasiones, y que, en general, tanto el antecedente como el consecuente muestran una alta dependencia entre sí.

A continuación se muestra una tabla con las 25 reglas de mayor confianza.

N	**Rule**	**Supp**	**Conf**	**Lift**
1	$\{ 3836(Oy\_3836), 3889(He I\_3889), 3935(K\_3935), 4103(H\mathrm{d}\_4103), 4342(Hg\_4342), 6565(Ha\_6565) \} \Rightarrow \{ 3970(H\_3970), 4863(Hb\_4863) \}$	0.19	1.00	2.59
2	$\{ 3836(Oy\_3836), 3889(He I\_3889), 3935(K\_3935), 4103(H\mathrm{d}\_4103), 6565(Ha\_6565) \} \Rightarrow \{ 3970(H\_3970), 4863(Hb\_4863) \}$	0.19	1.00	2.59

3	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 3935(K\_3935) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
4	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 4103(Hd\_4103) \\ 4306(G\_4306) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.15	1.00	2.59
5	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 4103(Hd\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.22	1.00	2.59
6	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 3935(K\_3935) \\ 4103(Hd\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
7	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 4103(Hd\_4103) \\ 4306(G\_4306) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.15	1.00	2.59
8	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3935(K\_3935) \\ 4103(Hd\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
9	$\left\{\begin{array}{c} 3935(K\_3935) \\ 4103(Hd\_4103) \\ 4306(G\_4306) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.15	1.00	2.59
10	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.22	1.00	2.59
11	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 4103(Hd\_4103) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.22	1.00	2.59

12	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 3935(K\_3935) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
13	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3935(K\_3935) \\ 4103(Hd\_4103) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
14	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 4306(G\_4306) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.16	1.00	2.59
15	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3935(K\_3935) \\ 4306(G\_4306) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.15	1.00	2.59
16	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 3935(K\_3935) \\ 4103(Hd\_4103) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.20	1.00	2.59
17	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 3935(K\_3935) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.21	1.00	2.59
18	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3935(K\_3935) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.21	1.00	2.59
19	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 4306(G\_4306) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.16	1.00	2.59
20	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 4103(Hd\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.23	1.00	2.59
21	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 3889(HeI\_3889) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.23	1.00	2.59
22	$\left\{\begin{array}{c} 3935(K\_3935) \\ 4103(Hd\_4103) \\ 4306(G\_4306) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.16	1.00	2.59

23	$\left\{\begin{array}{c} 3889(HeI\_3889) \\ 4103(H\mathrm{d}\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.24	1.00	2.58
24	$\left\{\begin{array}{c} 3836(Oy\_3836) \\ 4103(H\mathrm{d}\_4103) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.23	1.00	2.58
25	$\left\{\begin{array}{c} 3935(K\_3935) \\ 4103(H\mathrm{d}\_4103) \\ 4342(Hg\_4342) \\ 6565(Ha\_6565) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 4863(Hb\_4863) \end{array}\right\}$	0.23	1.00	2.58

El sistema permite, además, filtrar las reglas de tal modo que se muestren solamente aquellas en las que se encuentre un cierto ítem en el antecedente o en el consecuente de la regla. A modo de ejemplo, a continuación se muestran las 5 reglas con mayor confianza que contienen a la línea *CaII_8544* en el antecedente.

N	**Rule**	**Supp**	**Conf**
1	$\left\{ 8544(CaII\_8544) \right\} \Rightarrow \left\{ 5177(Mg\_5177) \right\}$	0.27	0.75
2	$\left\{\begin{array}{c} 4863(Hb\_4863) \\ 8544(CaII\_8544) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 6565(Ha\_6565) \end{array}\right\}$	0.17	0.86
3	$\left\{\begin{array}{c} 3970(H\_3970) \\ 8544(CaII\_8544) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 4863(Hb\_4863) \\ 6565(Ha\_6565) \end{array}\right\}$	0.17	0.89
4	$\left\{\begin{array}{c} 3970(H\_3970) \\ 8544(CaII\_8544) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3935(K\_3935) \\ 6565(Ha\_6565) \end{array}\right\}$	0.17	0.88
5	$\left\{\begin{array}{c} 3935(K\_3935) \\ 8544(CaII\_8544) \end{array}\right\} \Rightarrow \left\{\begin{array}{c} 3970(H\_3970) \\ 6565(Ha\_6565) \end{array}\right\}$	0.17	0.96

En la Figura 4.7 se muestra el resultado de realizar medidas de tiempo de ejecución de los algoritmos Apriori y FP-Growth para distintos niveles de soporte mínimo sobre estos datos.

4.5. Observaciones y análisis

Una vez observados estos resultados obtenidos a partir de espectros de la selección de objetos estelares del SDSS se pueden extraer las siguientes conclusiones.

Al seleccionar reglas con alto soporte se privilegia aquellas con líneas espectrales comunes a una gran cantidad de espectros. En particular, la mayor parte de las reglas son entre líneas del hidrógeno, que es el elemento más abundante en las estrellas y además tiene una serie de líneas espectrales en el visible. Existen pocas líneas de otros elementos presentes en estas reglas de alta confianza, como por ejemplo la línea H del calcio. Claramente estas se detectan en una gran cantidad de las estrellas con líneas brillantes del hidrógeno.

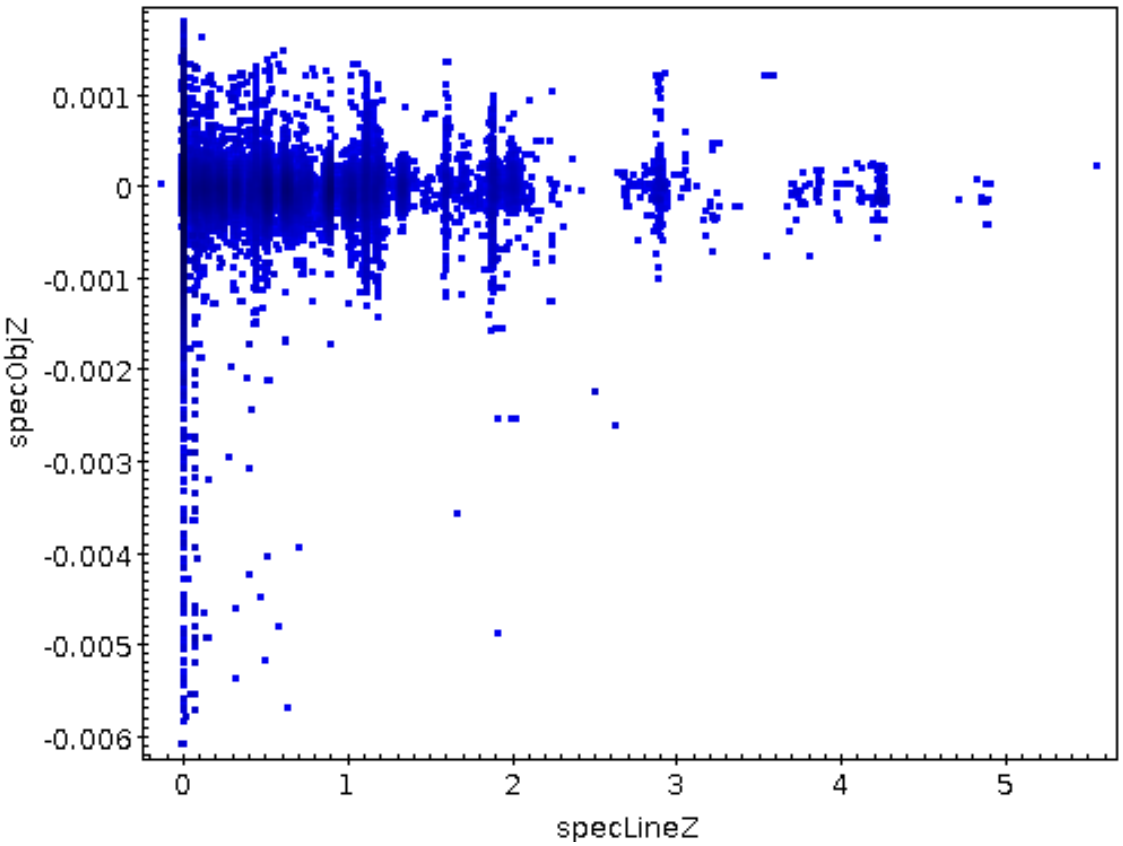

Figura 4.6: Gráfico de *redshift* de las líneas espectrales seleccionadas vs el del objeto al que pertenecen; una vez filtrados aquellas con valores inválidos de *redshift*

Además, se observa que al reducir el soporte y seleccionar por confianza, se encuentran conjuntos de líneas altamente correlacionados, pero presentes en una menor fracción del conjunto total de espectros. Por ejemplo, las líneas O, H_{e}, G y K aparecen, y están muy correlacionadas con las líneas H_a, H_b y H, entre otras.

En síntesis, puede decirse que es preferible, con el fin de no encontrar solo relaciones triviales o comúnes en demasía, buscar reglas por alta confianza y bajo soporte; siempre y cuando se cuente con un número muy grande de transacciones, dado que en la medida que este número crece se hace más interesante buscar soporte relativamente bajo con alta confianza.

En cuanto al desempeño y eficiencia de los algoritmos, en primera instancia sorprende el hecho que para valores de soporte menor que 0.15 los tiempos de ejecución del algoritmo *Apriori* sean mucho menores que los del algoritmo *FP-Growth*, siendo que este último fue concebido como una optimización del primero.

Sin embargo, y tal como indica la literatura al respecto [Kosters et al., 2003], las complejidades entre *Apriori* y *FP-Growth* no son directamente comparables; y, por tanto, no existe una garantía de que uno de ellos tenga mejor desempeño que el otro en todos los casos posibles. Más bien, *FP-Growth* demostrará una ventaja comparativa considerable en algunos casos, y en otros se verá ampliamente superado por *Apriori*.

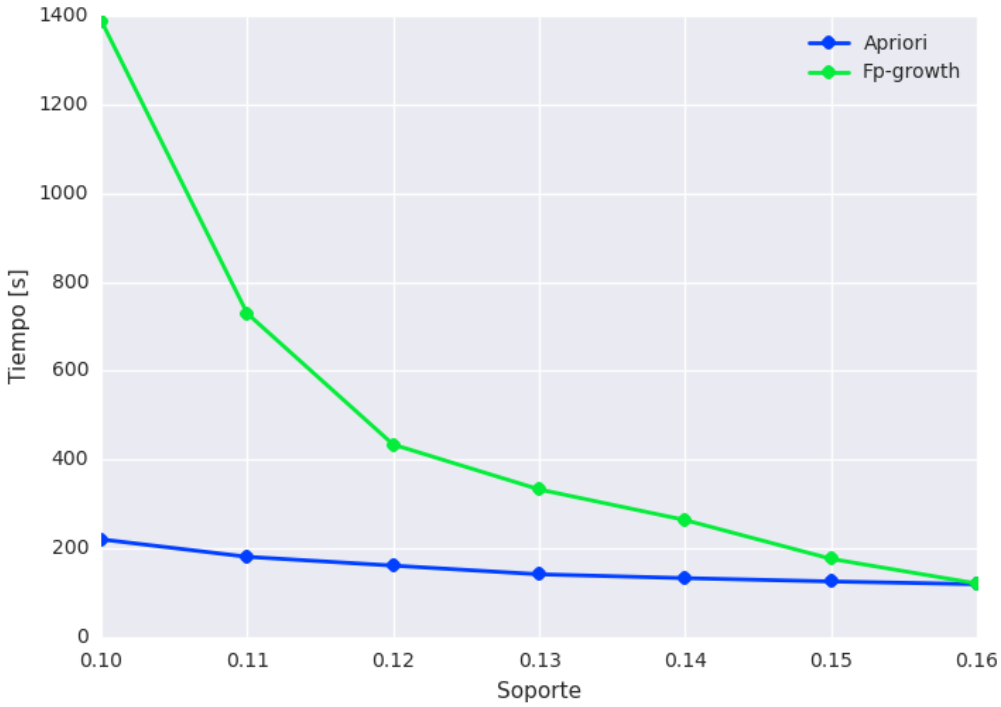

Figura 4.7: Grafico de tiempos de ejecución de algoritmos *Apriori* y *FP-Growth* para distintas medidas de soporte

En particular, los datos con los que se validaron los algoritmos resultan ser singulares en términos de lo reducido que es el universo de ítemes y el alto soporte que poseen la mayoría de los conjuntos de un ítem; los cuales se repiten mucho de una transacción a otra. Esto, junto con el hecho de que relativamente se trata de datos de mediana cantidad (del orden de 100.000 transacciones), hace que para el algoritmo *Apriori* no sea tan costoso recorrer el conjunto de datos, y que las operaciones de manipulación de árboles del algoritmo *FP-Growth* comiencen a ser más preponderante. A esto cabe agregar la posibilidad de que las operaciones de conjuntos de Python, que son clave en la operación del algoritmo *FP-Growth*, no se encuentren debidamente optimizadas para los requerimientos de este.

Conclusión

El objetivo del presente trabajo fue obtener *reglas de asociación* entre líneas espectrales, detectadas en espectros de frecuencia obtenidos a partir de observaciones astronómicas. Para ello, se realizó la implementación de un sistema de aprendizaje de reglas de asociación, o *Association Rule Learning (ARL)*, para conjuntos de transacciones. El sistema permite al usuario generar reglas que cumplan con medidas mínimas de relevancia estadística, tales como soporte y confianza, y posteriormente ser desplegadas en orden según estas mismas medidas. Junto con esto, el usuario es capaz de requerir al sistema que despliegue solamente aquellas reglas en las que esté presente un cierto ítem en su antecedente o consecuente; generando, de esta manera, más valor a los resultados en vista de su relevancia para el usuario.

Se implementó los algoritmos *Apriori* y *FP-Growth* con el fin de extraer conjuntos frecuentes a partir de un conjunto de transacciones. Posteriormente, se utilizó el algoritmo *Apriori* de generación de reglas para generar reglas de asociación a partir de estos conjuntos frecuentes. El desarrollo del sistema se realizó con miras a una arquitectura de software modular, que permitiera la aplicación de los algoritmos de ARL a datos lo más genéricos posibles. Dentro de un paquete principal de ARL se definieron clases que corresponden a abstracciones de las estructuras de datos de los que hace uso cada uno de los algoritmos, y de los métodos principales de extracción de conjuntos frecuentes y generación de reglas.

En particular, se enfocó su uso a datos provenientes de mediciones espectroscópicas astronómicas; con el fin de encontrar asociaciones lógicas entre líneas espectrales. Para ello se procedió a realizar pruebas de concepto sobre una base de datos de espectros ópticos del *Sloan Digital Sky Survey (SDSS)* en su *Data Release 7 (DR7)*, previo un pre-procesamiento y análisis de los datos. A partir de un conjunto de 1.189.817 de líneas espectrales asociadas a 120.250 objetos de tipo estelar se logró extraer 576 conjuntos de ítemes frecuentes, a partir de los cuales se generó un total de 5.181 reglas de asociación. Se aprecia un alto soporte en reglas que contienen especies tales como oxígeno y hidrógeno; y una alta correlación entre algunas especies entre sí, tales como el calcio y el hidrógeno.

A la luz de estos resultados se pudo comprobar que, al aumentar el tamaño de los datos, la medida de relevancia estadística de soporte deja de ser suficiente. Reglas con un bajo soporte, en tales circunstancias, siguen siendo de gran interés; siempre y cuando muestren alta calificación bajo otras medidas, tales como la confianza. Esto resulta muy importante de tener en mente a la hora de utilizar herramientas de este tipo dentro del área del procesamiento masivo de datos.

Quedó en evidencia, además, la superioridad del algoritmo *Apriori* por sobre *FP-Growth*,

que tuvo un tiempo de ejecución de a lo más la mitad que el de este último sobre datos de prueba reales y simulaciones, para requisitos soporte mínimo menores a 0.15. Esto debido, potencialmente, a las características de los datos con las cuales se efectuó la validación. A futuro es desable realizar más pruebas sobre distintos tipos de datos y número de transacciones con el fin de averiguar con certeza si este resultado se debe a las transacciones en sí, al número límitado de ítems del universo posible o a la implementación misma de los algoritmos, y si el resultado se mantiene al aumentar la cantidad de datos.

La relevancia y el impacto del presente trabajo se aprecia mejor en el marco de proyectos como el del *Atacama Large Millimeter Array (ALMA)*. Dentro de los próximos años este comenzará a generar grandes cantidades de datos de espectroscopía astronómica, los cuales inevitablemente se irán acumulando con el tiempo. El hecho de que muchos de estos datos se obtengan como consecuencia, y no como objetivo principal de muchas de las observaciones por parte de los astrónomos, es un indicador de la importancia de tener herramientas computacionales que permitan auxiliar al proceso de investigación y que disminuyan los requerimientos de horas-hombre necesarios para realizar descubrimientos de interés.

A lo largo de este trabajo se logró aprender detalles muy importantes del proceso de implementar una herramienta que utiliza algoritmos y métodos generales a una solución específica, en un dominio del conocimiento muy teórico y de lenguaje muy técnico, como es el de la astronomía. Se pudo asimilar lo que implica hacer un proceso de investigación previo a la fase misma de implementación de un sistema, con el fin de que sus prestaciones se encuentren alineadas con sus requerimientos. Y esto se vuelve aun más crucial en aplicaciones científicas interdisciplinarias. La interacción con expertos de diversas ramas del conocimiento y la investigación fue, sin lugar a dudas, uno de los puntos más importantes en el proceso de aprendizaje llevado a cabo en el desarrollo de este trabajo.

Queda para el desarrollo a futuro el optimizar el flujo de trabajo de la herramienta, mediante hacer más compacta las interfaces entre módulos y hacer más general la aplicación del sistema de pre-procesamiento de datos; con el fin de que se vuelva parte íntegra del sistema, replicable y adaptable por el usuario a datos de características diversas.

Otro importante objetivo que queda para futuro es la implementación de una interfaz gráfica de usuario, que facilite la visualización, manejo de resultados y el evitar labores repetitivas por parte del usuario en su flujo de trabajo. Una alternativa a este punto sería hacer que el sistema sea parte de alguna herramienta ya existente de visualización y operación de datos astronómicos.

Relacionado con esto está otro importante objetivo a futuro, que es el realizar la implementación de la herramienta en ambientes de computación de alto rendimiento, y el hacer que se conforme a estándares de observatorios virtuales. Medidas como estas expandirán de forma considerable las posibles aplicaciones futuras y el impacto de la solución desarrollada a lo largo de este trabajo.

Bibliografía

[alm, 2014] (2014). Atacama large millimeter/submillimeter array (alma). `http://www.almaobservatory.org`. online, accesed July 2014.

[pyt, 2014] (2014). Welcome to python.org. `http://www.python.org`. online, accesed July 2014.

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE.

[Agrawal et al., 1994] Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

[Bodon, 2010] Bodon, F. (2010). A fast apriori implementation. In *Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations (FIMI'03)*, volume 90.

[Brin et al., 1997a] Brin, S., Motwani, R., and Silverstein, C. (1997a). Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD Record*, volume 26, pages 265–276. ACM.

[Brin et al., 1997b] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997b). Dynamic itemset counting and implication rules for market basket data. In *ACM SIGMOD Record*, volume 26, pages 255–264. ACM.

[Cai et al., 1998] Cai, C. H., Fu, A. W.-C., Cheng, C., and Kwong, W. (1998). Mining association rules with weighted items. In *Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International*, pages 68–77. IEEE.

[Carmona-Saez et al., 2007] Carmona-Saez, P., Chagoyen, M., Tirado, F., Carazo, J. M., and Pascual-Montano, A. (2007). Genecodis: a web-based tool for finding significant concurrent annotations in gene lists. *Genome biology*, 8(1):R3.

[Chandola et al., 2009] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detec-

tion: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15.

[Chaves et al., 2011] Chaves, R., Górriz, J., Ramírez, J., Illán, I., Salas-Gonzalez, D., and Gómez-Río, M. (2011). Efficient mining of association rules for the early diagnosis of alzheimer's disease. *Physics in medicine and biology*, 56(18):6047.

[Cohen et al., 2007] Cohen, E., Fiat, A., and Kaplan, H. (2007). Associative search in peer to peer networks: Harnessing latent semantics. *Computer Networks*, 51(8):1861–1881.

[Davidson et al., 2010] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al. (2010). The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM.

[Dehaspe et al., 1998] Dehaspe, L., Toivonen, H., and King, R. D. (1998). Finding frequent substructures in chemical compounds. In *KDD*, volume 98, page 1998.

[Eguchi, 2013] Eguchi, S. (2013). "Superluminal"FITS File Processing on Multiprocessors: Zero Time Endian Conversion Technique. *Publ.Astron.Soc.Pac.*, 125:565.

[Estan et al., 2003] Estan, C., Savage, S., and Varghese, G. (2003). Automatically inferring patterns of resource consumption in network traffic. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 137–148. ACM.

[Evfimievski et al., 2004] Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. (2004). Privacy preserving mining of association rules. *Information Systems*, 29(4):343–364.

[Ferragina and Gulli, 2008] Ferragina, P. and Gulli, A. (2008). A personalized search engine based on web-snippet hierarchical clustering. *Software: Practice and Experience*, 38(2):189–225.

[Ghinita et al., 2008] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., and Tan, K.-L. (2008). Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM.

[Han et al., 2007] Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86.

[Han et al., 2004] Han, J., Pei, J., Yin, Y., and Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87.

[Harrington, 2012] Harrington, P. (2012). *Machine Learning in Action*. Manning Publications Co., Greenwich, CT, USA.

[Iqbal et al., 2013] Iqbal, F., Binsalleeh, H., Fung, B., and Debbabi, M. (2013). A unified data mining solution for authorship analysis in anonymous textual communications. *In-*

formation Sciences, 231:98–112.

[Karabatak and Ince, 2009] Karabatak, M. and Ince, M. C. (2009). An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications*, 36(2):3465–3469.

[Kobilarov et al., 2009] Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., and Lee, R. (2009). Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In *The semantic web: research and applications*, pages 723–737. Springer.

[Kosters et al., 2003] Kosters, W. A., Pijls, W., and Popova, V. (2003). Complexity analysis of depth first and fp-growth implementations of apriori. In *Machine Learning and Data Mining in Pattern Recognition*, pages 284–292. Springer.

[Kramer et al., 2001] Kramer, S., De Raedt, L., and Helma, C. (2001). Molecular feature mining in hiv data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 136–143. ACM.

[Lee and Stolfo, 2000] Lee, W. and Stolfo, S. J. (2000). A framework for constructing features and models for intrusion detection systems. *ACM transactions on Information and system security (TiSSEC)*, 3(4):227–261.

[Li et al., 2008] Li, X., Guo, L., and Zhao, Y. E. (2008). Tag-based social interest discovery. In *Proceedings of the 17th international conference on World Wide Web*, pages 675–684. ACM.

[Park et al., 1995] Park, J. S., Chen, M.-S., and Yu, P. S. (1995). *An effective hash-based algorithm for mining association rules*, volume 24. ACM.

[Patcha and Park, 2007] Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470.

[Romero and Ventura, 2007] Romero, C. and Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications*, 33(1):135–146.

[Romero et al., 2008] Romero, C., Ventura, S., and García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384.

[Savasere et al., 1995] Savasere, A., Omiecinski, E. R., and Navathe, S. B. (1995). An efficient algorithm for mining association rules in large databases.

[Škoda and Vážnỳ, 2011] Škoda, P. and Vážnỳ, J. (2011). Searching of new emission-line stars using the astroinformatics approach. *arXiv preprint arXiv:1112.2775*.

[Srikant and Agrawal, 1995] Srikant, R. and Agrawal, R. (1995). Mining generalized association rules. In *VLDB*, volume 95, pages 407–419.

[Srikant and Agrawal, 1996] Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. In *ACM SIGMOD Record*, volume 25, pages 1–12. ACM.

[Tennyson, 2010] Tennyson, J. (2010). *Astronomical Spectroscopy: An Introduction to the Atomic and Molecular Physics of Astronomical Spectra*. World Scientific.

[Wang et al., 2000] Wang, W., Yang, J., and Yu, P. S. (2000). Efficient mining of weighted association rules (war). In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 270–274. ACM.

[Wootten and Thompson, 2009] Wootten, A. and Thompson, A. R. (2009). The atacama large millimeter/submillimeter array. *Proceedings of the IEEE*, 97(8):1463–1471.

[York et al., 2000] York, D. G., Adelman, J., Anderson Jr, J. E., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J., Barkhouser, R., Bastian, S., Berman, E., et al. (2000). The sloan digital sky survey: Technical summary. *The Astronomical Journal*, 120(3):1579.

[Zaki and Ogihara, 1998] Zaki, M. J. and Ogihara, M. (1998). Theoretical foundations of association rules. In *3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 71–78. Citeseer.

Apéndice A

Tabla *SpecLineNames*

Nombre	**Valor**	**Descripción**
AlIII_1857	1857	1857.40
CaII_8500	8500	8500.36
CaII_8544	8544	8544.44
CaII_8665	8665	8664.52
CII_1335	1335	1335.31
CII_2326	2326	2326.00
CIII_1909	1909	1908.73
CIV_1549	1549	1549.48
G_4306	4306	4305.61
H_3970	3970	3969.59
Ha_6565	6565	6564.61
Hb_4863	4863	4862.68
Hd_4103	4103	4102.89
He_3971	3971	3971.19
HeI_3889	3889	3889.00
HeII_1640	1640	1640.40
Hg_4342	4342	4341.68
Hh_3799	3799	3798.98
K_3935	3935	3934.78
Li_6708	6708	6707.89
Lya_1216	1216	1215.67
Mg_5177	5177	5176.70
MgII_2799	2799	2799.12
Na_5896	5896	5895.60
NeIV_2439	2439	2439.50
NeV_3347	3347	3346.79
NeV_3427	3427	3426.85
NI_6529	6529	6529.03
NII_6550	6550	6549.86
NII_6585	6585	6585.27
NV_1241	1241	1240.81

OI_1306	1306	1305.53
OI_6302	6302	6302.05
OI_6366	6366	6365.54
OII_3727	3727	3727.09
OII_3730	3730	3729.88
OIII_1666	1666	1665.85
OIII_4364	4364	4364.44
OIII_4933	4933	4932.60
OIII_4960	4960	4960.30
OIII_5008	5008	5008.24
OVI_1033	1033	1033.82
Oy_3836	3836	3836.47
SII_4072	4072	4072.30
SII_6718	6718	6718.29
SII_6733	6733	6732.67
SiIV_1398	1398	1397.61
SiIV_OIV_1400	1400	1399.80
UNKNOWN	0	0.00