PEDECIBA Informática

Instituto de Computación – Facultad de Ingeniería

Universidad de la República

Montevideo – Uruguay

Minería de Calidad de Datos: Aplicación de Técnicas de Minería de Datos para la Evaluación de la Calidad de los Datos

Ing. Sergio Pio Alvarez

sergiop@fing.edu.uy, sergiopio@gmail.com

Tesis de Maestría

Directora Académica:

Dra. Adriana Marotta

amarotta@fing.edu.uy

Directoras de Tesis:

Dra. Adriana Marotta

amarotta@fing.edu.uy

Dra. Libertad Tansini

libertad@fing.edu.uy

Tribunal:

Dr. Ismael Caballero Muñoz-Reja

Dra. Lorena Etcheverry

Dr. Pablo Rodríguez Bocca

2018

Resumen

El aseguramiento de la calidad de los datos con los cuales se trabaja es crucial para tomar decisiones acertadas, efectivas y a tiempo. Lograr una buena calidad de datos no solo implica trabajar con datos que no contengan errores, sino que también incluye características tales como la completitud (tener la mayor cantidad posible de datos), la actualidad (que los datos sean lo más actuales posibles), la usabilidad (que los datos sean adecuados y comprensibles), y la disponibilidad (que se pueda acceder a ellos cuando se los necesita), entre muchas otras.

La minería de datos, por otra parte, permite descubrir información oculta en los datos, utilizando un paradigma inverso al usual: mientras normalmente se comienza planteando una hipótesis para luego tratar de confirmarla, la minería de datos propone identificar en forma automatizada patrones que pueden resultar interesantes y que posiblemente no hayan sido imaginados por los analistas.

Si bien ambas áreas son altamente relevantes en el mundo académico e industrial de la actualidad, donde la informática brinda un soporte tecnológico apropiado, la literatura existente y algunas experiencias muestran que existe muy poca o nula integración entre la calidad de datos y la minería de datos. En general, los trabajos pertenecientes a un área suelen ser ajenos a los existentes en la otra.

En este trabajo se realiza un estudio en profundidad de las dos áreas introducidas para luego hacer un análisis de los mecanismos que permitirían vincularlas, y finalmente implementar técnicas que permitan abordar el análisis de la calidad de conjuntos de datos aprovechando las capacidades inherentes de la minería de datos.

El trabajo presenta dos propuestas nuevas para la aplicación de técnicas de minería de datos para la evaluación de la calidad de datos, que fueron presentadas en dos eventos internacionales especializados. Una de ellas se orienta a la determinación de si un conjunto de datos es suficientemente actualizado, y la otra se orienta al análisis de datos faltantes. Además, se presenta también una tercera propuesta, aún en etapa de formulación, para evaluar qué tan usable es un conjunto de datos en base a sus características.

Palabras clave: calidad de datos, minería de datos, minería de calidad de datos.

Índice general

1 - Introducción

1.1 - Contexto

El aseguramiento de la calidad de los datos con los que se trabaja es crucial para tomar decisiones eficientes con el fin de lograr los mejores resultados. Cuando se habla de calidad de datos no solo se trata de trabajar con datos correctos o que no contengan errores (concepto algo ambiguo como se verá más adelante), sino que también incluye características tales como la completitud (tener la mayor cantidad posible de datos y lo más completos posibles), la actualidad, frescura y vigencia (que los datos sean lo más actuales posibles y no estén obsoletos), la usabilidad (que los datos sean los necesarios y a la vez sean comprensibles), entre muchas otras. Desde hace bastante tiempo se han hecho esfuerzos para medir, evaluar y mejorar, la calidad de los datos. Para esto se han realizado clasificaciones de los atributos de los datos que afectan a su calidad, y se han propuesto técnicas y procedimientos enfocados en su mejoramiento.

La minería de datos, por otra parte, es un área que ha visto un gran crecimiento en los últimos tiempos debido a que permite descubrir información importante oculta entre los datos, utilizando un paradigma inverso al usual: mientras normalmente se comienza por plantear una hipótesis para luego tratar de confirmarla (para lo cual es necesario contar con un conocimiento relativamente profundo sobre el dominio), la minería de datos propone identificar patrones que pueden resultar interesantes y que posiblemente no hayan sido imaginados por los analistas, en forma automatizada.

1.2 - Motivación

El abordaje de la calidad de datos es una tarea que en general exige bastante trabajo y particularmente un gran conocimiento del dominio. Cuando se trata de tener datos sin deficiencias, es necesario saber detectar cuándo se está ante una deficiencia, y cómo corregirla; cuando se trata de contar con datos completos es necesario poder detectar si están todos los datos, y cuando se está ante un dato faltante se debe poder explicar porqué falta, si es que realmente falta o no corresponde; y cuando se pretende trabajar con datos actuales y no obsoletos es necesario saber detectar cuándo los datos lo son.

Por otra parte, la minería de datos es un área en constante evolución, emparentada con el aprendizaje automático y la inteligencia artificial. Todas ellas son áreas de gran interés y aplicabilidad. Con el gran volumen de datos que se generan, almacenan y utilizan hoy en día, y con la complejidad que tienden a presentar dichos datos, se hace cada vez más necesario contar con herramientas de análisis automatizado que permita descubrir información en ellos.

Surge entonces la pregunta de si es posible aplicar técnicas de minería de datos para abordar el análisis de la calidad de un conjunto de datos. Dicha integración, si bien no es novedosa, ha recibido poca atención por parte de los investigadores, aunque el potencial que podría alcanzar merece una revisión.

En este contexto se propone un estudio en profundidad de las dos áreas involucradas (calidad de datos y minería de datos) para luego hacer un análisis de los mecanismos que permitirían vincularlas, y finalmente proponer técnicas que permitan abordar el análisis de la calidad de conjuntos de datos aprovechando las capacidades inherentes de la minería de datos.

Se parte de la suposición de que si bien ambas áreas son altamente relevantes en el mundo informático, las personas involucradas en una suelen ser ajenas a la otra. Esta suposición toma como base la experiencia propia y la literatura conocida previamente.

1.3 - Objetivos

Este trabajo tiene los siguientes objetivos generales:

- Estudiar el estado del arte en el área calidad de datos, desde su definición y las múltiples concepciones existentes hasta las técnicas habituales para su tratamiento.
- Estudiar el estado del arte en el área minería de datos, desde su definición e incluyendo los diferentes problemas que aborda y las técnicas que provee para cada uno.
- Analizar las posibilidades de integración de las dos áreas.
- Proponer algunas técnicas concretas para el abordaje de la calidad de datos que apliquen técnicas de minería de datos.

1.4 - Organización del documento

El resto del documento se divide en cinco secciones como se describe a continuación.

El capítulo 2 aborda el estado del arte en lo que el área de la Calidad de datos, exponiendo diferentes visiones existentes, incluyendo un relevamiento de las principales dimensiones de calidad.

El capítulo 3 aborda el estado del arte del área de la Minería de datos, abarcando las diferentes técnicas que las componen.

El capítulo 4 se enfoca en la Minería de Calidad de Datos, un área poco abordada que pretende integrar la Minería de Datos en el abordaje de la Calidad de Datos, aplicando técnicas de la primera para resolver problemas de la segunda. En este capítulo se incluye un relevamiento de trabajos relacionados, y se describen las principales propuestas existentes.

El capítulo 5 presenta tres propuestas propias sobre la aplicación de técnicas de minería de datos para evaluar la calidad de un conjunto de datos.

El capítulo 6 presenta las conclusiones obtenidas al finalizar el trabajo, al tiempo que se describen los aportes realizados y las limitaciones del trabajo.

Finalmente, en el apéndice se incluye información adicional para ampliar conceptos manejados a lo largo del documento.

2 - Calidad de datos

Este capítulo se centra en el primero de los grandes temas abordados, la **calidad de datos** (***data quality***). Se realiza primero una introducción al tema, mostrando diferentes aproximaciones sobre qué es y qué implica, para luego describir las diferentes formas de analizar la calidad de los datos.

2.1 - Datos e información

El término "**dato**" (***data*** en inglés) ha sido usado por científicos y filósofos por siglos [001]. Sin embargo, la mayoría de los autores que han escrito sobre la noción de dato han evitado, o no han podido, dar una definición concreta sobre qué significa el término, usándolo informalmente y a menudo como sinónimo de "**información**" [002]. Aunque parecería más apropiado interpretar al término información (*information*) como datos que han sido procesados de alguna manera [003] [004], en general son quienes provienen del área de la administración los que suelen diferenciar más ambos conceptos, mientras que en términos informáticos suelen usarse en forma intercambiable [005]. Según la Organización Internacional para la Estandarización (ISO, *International Standard Organization*) los datos son la representación de la información de una manera adecuada para la comunicación, interpretación y procesamiento (ISO 25012:2008 (E)), por lo que si bien a veces son utilizados como sinónimos, no deberían serlo [006]. A nivel general, puede asumirse que el término dato es usualmente utilizado para referirse a la información en etapas tempranas del tratamiento de un conjunto de datos con un fin específico, mientras que se utiliza el término información para cualquier etapa, por lo que también incluye a los datos [007]. En otras palabras, toda información es un dato en sí mismo, pero no todos los datos constituyen información.

2.2 - Concepto de calidad de datos

Intuitivamente parece fácil llegar a una noción sobre qué es la calidad de datos: consiste en lograr datos adecuados y sin deficiencias, en el lugar y el momento apropiado para realizar la tarea que se pretende [008][009]. Aunque la noción intuitiva parece simple es bastante más complejo alcanzar una definición formal, y de hecho existen múltiples aproximaciones diferentes, cada una abordando el tema desde un punto de vista diferente [010]. De entre todas ellas se destacan tres que por separado consideran todas las características relevantes sobre la calidad de datos:

- "Adecuación al uso" [009][011][012][013][014][015][016],
- "Conformidad con los requerimientos" [010][017][018],
- "Consistencia con el mundo real" [016][019][020].

La "*adecuación al uso*" implica que un conjunto de datos es de buena calidad (o *tiene* buena calidad), si se ajusta al uso que se le pretende dar. Por ejemplo, si una persona debe decidir si es buen momento para invertir en acciones de cierta compañía pero solo cuenta con los resultados deportivos del día anterior, éstos para su objetivo particular no son útiles, y por tanto no son de buena calidad. La "*conformidad con los requerimientos*" indica que un conjunto de datos es de buena calidad si se ajusta a cierta especificación realizada previamente a su uso. Por ejemplo, un requerimiento podría ser que para cierto atributo los datos deben cumplir con una determinada regla sintáctica; si los datos disponibles se ajustan a dicha especificación se dice que son de buena

calidad. Finalmente, por "*consistencia con el mundo real*" se entiende que si los datos constituyen una abstracción del mundo real, cuanto más cercana es la representación que hacen de dicho mundo entonces mejor es su calidad [021]. Considerando las tres definiciones brindadas en conjunto se puede concluir que **la calidad de datos es un concepto subjetivo y dependiente del contexto** [003].

Es importante destacar que son muchas las características que resultan importantes al momento de evaluar la calidad de los datos, tales como que estén disponibles (que se pueda contar con ellos cuando se los necesita), que sean correctos (que no contengan errores o que no estén incompletos), que sean actuales (que sean suficientemente recientes o que no estén obsoletos), que sean comprensibles (que puedan ser entendidos por las personas que deben usarlos), entre muchas otras. Estas características son las que permiten evaluar la calidad de los datos, y por eso es de suma importancia establecer cuáles son, qué significan, y cómo se relacionan entre sí. Fallar en establecer un entendimiento apropiado de esas características hará que sea más difícil la tarea de lograr datos de buena calidad [022].

Existe una tendencia a relacionar a la calidad de los datos casi exclusivamente con la ausencia de errores en ellos [014][023]. Sin embargo es posible que los datos no contengan errores y aún así no sean de buena calidad. Por ejemplo, una persona podría estar buscando un producto particular en un catálogo de una tienda comercial. Existe alguna posibilidad de que el catálogo haya sido impreso con errores, pero aunque no fuese el caso bajo el criterio del observador los datos representados en él podrían ser de baja calidad, por ejemplo si nota que el catálogo es muy viejo (seguramente el precio ya no sea el mismo o el producto ya no esté disponible), lo que constituye un problema de *actualidad* (los datos no están actualizados). También podría suceder que el catálogo no tenga errores y sea perfectamente actual pero esté incompleto (el producto buscado no figura en él o no están expresadas todas sus características), en cuyo caso hay un problema de *completitud*. Por otra parte, podría ser que el producto deseado sí figure en el catálogo con todas sus características relevantes y el catálogo sea perfectamente actual y sin errores, pero el texto esté redactado en un idioma extranjero desconocido (no se entiende qué dice), lo que significa un problema de *comprensibilidad*, o que el dato buscado se encuentra en un trozo del catálogo dañado o faltante que impide su lectura, lo que constituye un problema de *accesibilidad*. Se puede ver así que son muchas las características que afectan de alguna manera a la calidad de los datos [014][016][023], por lo que se dice que la calidad de datos es multidimensional (abarca a muchas dimensiones a la vez) [005][024][003]. Todas esas características que hacen a la calidad de los datos se suelen clasificar en categorías denominadas **dimensiones de calidad**. No hay una caracterización universal de las dimensiones de la calidad de datos, y diferentes autores establecen sus propias dimensiones, a veces solapándose entre sí. En la página 14, la Figura 1 muestra una clasificación particular (propia) de las dimensiones de calidad de datos adoptada para este documento.

Cualquiera que sea la definición que se considere sobre calidad de datos se asume que los datos que la cumplen cabalmente tienen calidad perfecta o ideal. Sin embargo, debido a múltiples factores (económicos, técnicos, operativos, entre otros), en el mundo real es difícil, incluso imposible, lograr una calidad de datos perfecta. Cuando no se logra la calidad perfecta se dice que existe algún tipo de **deficiencia** en los datos, y que éstos son **datos sucios** (*dirty data*). Una **deficiencia de calidad** es cualquier problema encontrado en los datos analizados bajo una o más dimensiones de calidad que hace que su calidad no concuerde con la definición elegida. Algunos autores utilizan el término "**error**" aunque dicho término puede usarse de muchas maneras, siendo el significado más amplio el que indica que los datos no cumplen las expectativas de los usuarios y no necesariamente que son incorrectos o están mal formados [025]. Por esta razón es preferible evitar el uso de dicho término en favor de otros menos categóricos como deficiencia o problema.

Las deficiencias de los datos pueden variar según se trate de sistemas monofuentes (los datos proceden de una única fuente de datos) o sistemas multifuentes (los datos proceden de la integración de dos o más fuentes) [026]. La calidad de los datos en un sistema monofuente está relacionada con las restricciones definidas en ella. En las fuentes sin esquema fijo (como los sistemas de archivos y las bases de datos no-sql como las documentales) hay pocas restricciones que se puedan imponer sobre los valores admisibles, dando lugar a una alta probabilidad de encontrar deficiencias en los datos, mientras que en las fuentes con esquema fijo (como las bases de datos relacionales) se pueden imponer restricciones estrictas. Los problemas que se presentan en los sistemas monofuentes se agravan al integrar múltiples fuentes, donde cada una puede tener sus propios problemas, y además presentar distintos tipos de heterogeneidades, como semántica, de representación, etc..

Como se mencionó anteriormente la calidad de datos es un concepto relativo que debe evaluarse dentro de un contexto determinado y en el marco de una tarea particular. Así como no existe una definición estandarizada de calidad de datos tampoco existen niveles de calidad aplicables en todos los contextos y para todas las tareas. Que los datos sean de buena calidad no implica que sean perfectos para cualquier uso, ni siquiera para una tarea particular, sino que al menos reúnen un conjunto de características que los hacen adecuados para el uso que se pretende darles.

Es muy importante recordar que los datos son generados, usados y mantenidos mediante sistemas informáticos. Los sistemas informáticos influyen directamente en la calidad de los datos y recíprocamente la calidad de los datos también afecta a los sistemas informáticos [027]. De hecho los sistemas informáticos suelen ser responsables de las deficiencias de los datos, principalmente porque ellos mismos presentan deficiencias, tanto de diseño como funcionales. Las deficiencias de diseño están vinculadas a la concepción del sistema informático, y tienen como efecto que éste se vea limitado para representar fielmente al mundo real (no todo el mundo real puede ser representado), o que permita representar algo diferente al mundo real (cosas que realmente no ocurren en el mundo real). Estas deficiencias se pueden agrupar, a su vez, en tres categorías [021]:

- *Representación incompleta*: algunos estados del mundo real no pueden ser representados en el sistema informático. Por ejemplo, si para cada persona se permite registrar solo un número telefónico el sistema informático no podrá representar los casos en los que una persona tiene más de un teléfono.
- *Representación ambigua*: dos o más estados del mundo real se representan de igual manera en el sistema informático, no permitiendo determinar cuál de ellos está representado. Por ejemplo, cuando el sistema informático sólo admite ingresar el número de teléfono de las personas pero no especificar su tipo (domicilio, trabajo , móvil).
- *Representación inválida*: se permite representar estados que no pueden existir nunca. Por ejemplo, si el sistema informático no controla la edad de las personas o permite registrar fechas tales como “32/13/20018”.

Habitualmente cuando se habla de evaluar la calidad de un sistema de información el enfoque se pone únicamente en el software y se aparta a los datos. La validación y verificación son técnicas en las que se asume que los datos son provistos por agentes externos. Históricamente la comunidad del software ha subestimado la calidad de los datos asumiendo que los problemas de la baja calidad de los datos no es un problema de los profesionales que realizan el software sino de los usuarios que lo utilizan [027].

2.3 - Efectos de la calidad de los datos

Es bastante repetida la observación de que los datos son en la actualidad un activo muy importante de cualquier organización [003][004][014][028][029]. Cada vez se recolectan más datos y se publican más datos; al mismo tiempo es cada vez más claro que la mayor parte de esos datos son de baja calidad [008]. Algunos datos son simplemente incorrectos o están desactualizados, otros están pobremente definidos o no pueden ser comprendidos adecuadamente por los usuarios, y otros datos no son relevantes para las tareas que se realizan. Así, los datos de baja calidad son la norma no la excepción [008]. Según una publicación de The Wall Street Journal citada en [021] "g*racias a las computadoras, bases de datos enormes y repletas de información están a nuestro alcance, a la espera de ser explotadas. Pueden ser utilizadas para encontrar prospectos de ventas entre los clientes existentes, analizadas para descubrir hábitos corporativos costosos, manipuladas para predecir tendencias. Sólo hay un problema: esas bases de datos enormes pueden estar llenas de basura... En un mundo donde la gente se encamina hacia la gestión de la calidad total, una de las áreas críticas es la de los datos*". Los datos no deberían recolectarse y almacenarse solo porque se tiene la capacidad de hacerlo, sino para usarlos, y el primer paso para permitir su utilización es velar por su calidad [009][012]; desafortunadamente, muchas personas asumen que cuando mayor es el volumen de datos recolectados mayores serán los beneficios que pueden ser obtenidos, aún cuando no tienen en consideración que no saben o no tienen las herramientas apropiadas para extraer el conocimiento apropiado de esos datos [030].

Los efectos que pueden tener los datos de baja calidad son variados: el incremento de los costos operativos, la insatisfacción de los clientes, la dificultad para la toma de decisiones y para la ejecución efectiva de estrategias, el daño en la imagen corporativa, entre otros [008][015][022] [028]. En general, los costos ocasionados por la utilización de datos de baja calidad son difíciles de determinar: algunos costos como los correspondientes a las tareas de detección y corrección de deficiencias se pueden medir (según el costo de las herramientas, las personas involucradas, el tiempo requerido, etc.) pero otros como la insatisfacción de los clientes son más difíciles de hacerlo.

Naturalmente los detalles particulares sobre los problemas gerenciales y operativos de las organizaciones varían de caso a caso, pero probablemente la causa de muchos de los problemas que enfrentan las organizaciones tiene a la baja calidad de datos como una de sus raíces. Por esta razón la calidad de datos debería ser considerada un área funcional transversal a las organizaciones, abarcando a todas las áreas funcionales [031].

Adicionalmente, la consideración de los actores involucrados es importante porque son éstos los encargados de determinar cuáles datos se recolectan, cómo se recolectan, qué se hace con ellos, cómo se los almacena, y fundamentalmente, para qué y cómo se los emplea. Se identifican tres tipos de actores, o roles, en el proceso de manufacturación de información [007][013][015]: los **productores** de los datos, incluyendo a las personas responsables de hacer mediciones, observar lecturas y solicitar datos a los clientes, y las encargadas de ingresar los datos en los sistemas informáticos; los **custodios**, quienes proveen y administran los sistemas informáticos utilizados para almacenar, procesar y asegurar los datos, incluyendo a los administradores de bases de datos y de sistemas; y los **consumidores**, quienes utilizan los datos para su provecho y que, como resultado, pueden también convertirse en productores. En la práctica suele haber conflictos en las visiones que tienen sobre los datos y su calidad los diferentes actores. No es poco usual que los custodios consideren que sus datos son de alta calidad mientras que los consumidores los vean difíciles de utilizar para sus propósitos y por tanto los consideren de baja calidad.

2.4 - Evolución del estudio de la calidad de datos

La preocupación por la calidad de los datos no es reciente, sino que a lo largo de la historia las personas se han beneficiado o han sufrido por la calidad de los datos con los que cuentan (un ejemplo sencillo es la información disponible para los comandantes militares durante los enfrentamientos, los cuales han existido desde siempre). Lo que sí es nuevo es la explosión de la cantidad de datos disponibles y la creciente dependencia de las personas respecto a ellos [032]. También es nueva la intención de utilizar los datos para una amplia gama de tareas, incluso diferentes para las cuales fueron recolectados [033]. Adicionalmente, hasta hace relativamente poco tiempo no había mucha conciencia entre los directivos de las organizaciones sobre la importancia de la calidad de los datos y su impacto en los costos financieros y operativos [007]. En la actualidad es cada vez más frecuente encontrar en la literatura técnica, y hasta en las noticias, referencias a la calidad de los datos y su impacto en el mundo real [034].

La calidad de datos ha sido abordada desde diferentes áreas de investigación, principalmente desde la estadística, la administración y la computación [016]. Los estadísticos fueron los primeros en abordar algunos de los problemas relacionados con la calidad de datos sobre finales de la década de 1960 [012][016], aunque también se puede remontar a los finales de la década de 1940 con los estudios de la teoría de la información de C. Shannon [003]. Posteriormente, en la década de 1980 los gerenciadores, a partir del reconocimiento de que la calidad de los datos no es más que un caso particular de la calidad de los productos y procesos, comenzaron a enfocarse en cómo controlar los sistemas de tratamiento de datos para detectar y eliminar los problemas en la calidad de los datos [012][016]. Recién a principios de la década de 1990 los informáticos se introdujeron formalmente en al área, enfocándose específicamente en cómo definir, medir y mejorar la calidad de los datos almacenados en bases de datos y repositorios de información electrónicos [012][016]. Una de las razones para justificar la demora de la introducción de los informáticos en el área es la creencia difundida por entonces de que el tiempo de vida de los sistemas de información sería de apenas unos pocos años, por lo que no tendría sentido realizar grandes esfuerzos en asegurar la calidad de los datos [020]; en la práctica la mayoría de los sistemas de información han vivido mucho más tiempo de lo anticipado. Otra razón radica en el hecho de que el software y los datos no eran considerados bienes organizacionales con un valor concreto sino meras herramientas para la operativa de la organización; en la actualidad el software es considerado como un activo importante en el balance de las organizaciones, considerando el costo de crearlo y mantenerlo, así como los beneficios en que redunda su utilización, y los datos van siguiendo el mismo proceso [027][032] [035]). Adicionalmente, es cada vez más frecuente que la utilización que se hace de los datos exceda la aplicación para los cuales fueron recolectados originalmente, siendo procesados en conjunto con datos provenientes de otras fuentes, y son usados por usuarios que no están familiarizados con ellos [017].

Para abordar los problemas de calidad de datos en 1992 el MIT (*Massachussets Institute of Technology*) lanzó el programa TDQM (*Total Data Quality Management*) con el objetivo de definir procesos y estándares para realizar el tratamiento de la calidad de los datos [022][036], lo que es considerado como el inicio de la preocupación formal por la calidad de datos que luego llevó a que muchos investigadores se enfocaran en este campo dando lugar a la revista *Journal of Data and Information Quality* (JDIQ) de la ACM (*Association for Computing Machinery*). Recientemente los investigadores han estado estudiando los problemas de la calidad de datos desde la perspectiva de su generación y ya no solo desde el punto de vista de los propios datos [027]. La idea subyacente es que a través de la mejora de los procesos de generación de los datos se pueden mejorar a los propios datos, de la misma forma que mejorando los procesos de producción de software se puede lograr

software de mejor calidad [027][035].

Las investigaciones realizadas en las dos o tres últimas décadas han producido mucho conocimiento y han expandido la habilidad para resolver muchos de los problemas relacionados a la calidad de datos y a la información [036]. En [003] los autores hacen un relevamiento de diferentes esfuerzos orientados en resaltar el problema y proponer soluciones en dicho sentido. Sin embargo, al parecer no se ha avanzado lo suficiente aún como para definir métodos estándar para medir las deficiencias y solo unas pocas organizaciones evalúan la calidad de sus datos en forma rutinaria [034], aunque un gran número considera que es uno de los puntos más importantes respecto de la tecnología de la información [037]. Aún en la actualidad las evaluaciones de la calidad de los datos son generalmente hechas de una manera informal (*ad-hoc*) para resolver problemas específicos, faltando en la práctica la aplicación de principios fundamentales necesarios para desarrollar métodos reutilizables [005]. Si bien existe una cantidad importante de herramientas orientadas a la gestión, evaluación, mejora y control de la calidad de datos, en general cada una se centra en unos pocos aspectos del proceso de evaluación y mejora de la calidad de datos y tienen aplicabilidad limitada [038].

2.5 - Modelos, dimensiones, factores y métricas de calidad de datos

Los datos tienen muchas características de calidad, algunas de las cuales pueden resultar más importantes o interesantes que otras en cada contexto específico. Así como es difícil gestionar la calidad de un producto particular sin comprender los atributos que lo caracterizan es también difícil gestionar la calidad de los datos sin conocer las características que los definen [024]. Estas características de calidad son denominadas **dimensiones de calidad** [016]. Una dimensión de calidad es una característica de los datos que puede ser evaluada según estándares establecidos para determinar la calidad de dichos datos [039]. Si bien la expresión dimensión de calidad ha sido usada por años para encapsular distintas características de la calidad de los datos, el conjunto de dimensiones de calidad consideradas aún hoy no está universalmente acordado [021][039]. De todas formas las dimensiones de calidad deben ser consideradas solo como indicadores para medir y comunicar la calidad de los datos, por lo que mientras exista algún tipo de acuerdo que permita el entendimiento es admisible utilizar diferentes definiciones [039].

Probablemente los primeros en abordar el tema de las dimensiones de calidad de datos fueron Donald Ballou y Harald Pazer quienes en 1985 publicaron un artículo ([011]) en el cual realizaron una propuesta sobre un conjunto de dimensiones de calidad de datos que incluía correctitud (*accuracy*), completitud (*completeness*), consistencia (*consistency*) y oportunidad (*timeliness*) [014] [033]. Si bien esta propuesta es bastante antigua aún hoy en día es la base para otras más complejas [009][012][033][040][041]. Muchas fueron las propuestas de otros autores que le siguieron ([042], [043], [044], entre otras). Si bien cada una de esas propuestas considera conjuntos de dimensiones de calidad diferentes, algunas de ellas son consideradas casi universalmente, aunque no existe acuerdo total sobre su nombre o significado: en diferentes propuestas el mismo nombre es dado a conceptos distintos (homonimia) al tiempo que diferentes nombres han sido dados a elementos similares (sinonimia) [016]. En general la correctitud y la completitud son las únicas que son consideradas en la casi totalidad de las propuestas, encontrándose en segundo lugar las relacionadas a la consistencia, la temporalidad y la comprensibilidad [016].

Las dimensiones de calidad de datos se clasifican en 4 tipos [013][028][045]:

- **Intrínsecas**: implican características que son propias de los datos y que no dependen del contexto dentro del cual son consideradas. Algunos ejemplos de dimensiones de calidad de

datos intrínsecas son la correctitud (*accuracy*), la objetividad (*objetivity*), la credibilidad (*believability*), y la reputación (*reputation*).

- **Contextuales**: implican características de la calidad de datos que resaltan la dependencia del contexto bajo el cual se consideran los datos por lo que deben ser evaluadas en un contexto particular. Algunos ejemplos de dimensiones de calidad de datos contextuales son la relevancia (*relevancy*), la oportunidad (*timeliness*), la completitud (*completeness*) y el volumen (*amount*).
- **Representacionales**: implican características de la calidad de los datos que afectan a la capacidad de aprovechamiento de los datos. Algunos ejemplos de dimensiones de calidad representacionales son la interpretabilidad (*interpretability*), la comprensibilidad (*understability*), la brevedad (*conciseness*) y la consistencia (*consistency*).
- **Relacionadas con la accesibilidad**: implican características de la calidad que atañen a la posibilidad de los usuarios de acceder a los datos y utilizarlos. Algunos ejemplos de dimensiones de calidad son la accesibilidad (*accesibility*), la seguridad (*security*) y la privacidad (*privacy*).

De todas maneras, así como no hay uniformidad en la concepción de las dimensiones de calidad tampoco lo hay en su clasificación. Por ejemplo, la completitud es una dimensión que Ballou y Pazer clasifican como intrínseca mientras que Wang y Strong, otros reconocidos autores, la clasifican como contextual [045]. De igual manera pueden existir dudas sobre como clasificar la consistencia, la privacidad o la credibilidad.

Es importante considerar que la calidad de datos, aunque es un concepto multidimensional, es un fenómeno único, y como resultado las dimensiones de calidad no son independientes entre sí sino que algunas afectan a otras [025][045][046]. Por ejemplo, cuanto más actuales sean los datos mayor será la probabilidad de que sean correctos, lo que implica que la actualidad y la correctitud mantendrían una relación directa, mientras que la actualidad y la oportunidad podrían mantener una relación inversa ya que al buscar datos más actuales éstos podrían lograrse demasiado tarde. El conocimiento de las dependencias entre las dimensiones de calidad es importante porque permite diagnosticar cuál es la causa más probable para los problemas de calidad, permite seleccionar la estrategia más efectiva para mejorar la calidad de datos, y permite actuar en forma proactiva para evitar el degradamiento de la calidad de datos [046].

Las diferentes visiones que tienen los distintos autores al momento de definir las dimensiones de calidad conduce a que se haga de forma demasiado amplia, agrupando bajo un mismo nombre a características relativamente relacionadas aunque diferentes. Por ello se suele desmenuzar más el concepto de dimensiones de calidad en conjuntos de **factores de calidad**, donde cada factor de calidad es una característica definida y concreta de los datos. La Figura 1 muestra una posible clasificación de un conjunto de dimensiones de calidad de datos, con sus respectivos factores de calidad; esta clasificación no pretende abarcar a todas las posibles dimensiones y factores de calidad sino que solo ilustra los conceptos de dimensión y factor de calidad.

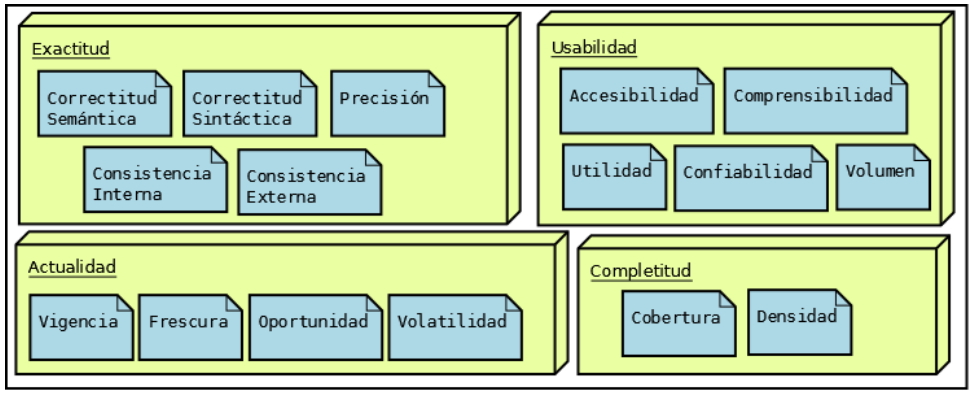

Figura 1: Propuesta personal de dimensiones y factores de calidad

Una vez escogido el conjunto de dimensiones de calidad y sus correspondientes factores de calidad, para poder evaluar la calidad de los datos en vista de ellos es necesario medirlos de alguna manera, para lo cual es preciso definir métricas. Una **métrica** es un instrumento para asignar valores (preferentemente numéricos) a los datos según cada factor de calidad, y se define en base a la semántica (qué y cómo se mide, por ejemplo "contar la cantidad de valores nulos"), la unidad de medida y la granularidad (nivel de detalle, por ejemplo atributo, entidad o conjunto de datos). También es necesario definir procedimientos de **agregación de las métricas** que permita determinar una medida de calidad para una entidad dadas las medidas de calidad de cada uno de sus atributos, y luego determinar la medida de calidad de todo el conjunto de datos dada la medida de calidad de cada una de las entidades. Para esto, lo usual es utilizar promedios simples, ponderados o sensibles.

El conjunto de dimensiones de calidad seleccionadas, con sus respectivos factores de calidad, y las métricas de éstos, constituyen lo que se denomina un **modelo de calidad**. De esta manera, el análisis de un conjunto de datos siempre se realiza en un contexto particular utilizando un modelo de calidad.

En las siguientes secciones se describen las dimensiones, factores y métricas de calidad de datos representadas en la Figura 1.

2.5.1 - Exactitud

La **exactitud** (***accuracy***) hace referencia a los atributos de los datos que permiten evaluar en forma objetiva qué tan bien representan al mundo real, permitiendo indicar si los datos se corresponden con el mundo real, si presentan o no errores, si tienen suficiente detalle, entre otras cualidades [025]. Es una dimensión que está incluida, con variantes, en todas las propuestas [021] [040]. Dentro de esta dimensión se suele definir tres factores de calidad: la correctitud semántica, la correctitud sintáctica, y la precisión, a los cuales usualmente se adicionan otros dos llamados consistencia interna y consistencia externa (aunque algunos autores los asignan a una dimensión de calidad separada llamada consistencia).

La **correctitud semántica (*semantic correctness*)** permite evaluar qué tan cercana es la representación lograda del mundo real dada por los datos disponibles. Por ejemplo, se podría evaluar si un registro corresponde a la persona correcta, si el dato no está equivocado, etc. Cuanto mayor sea la correctitud semántica de un conjunto de datos más cercano estará del mundo real la representación hecha con ellos [025]. La medición de la correctitud semántica suele realizarse utilizando valores binarios, donde el valor 1 indica que el dato es correcto y el valor 0 lo contrario

(podría también utilizarse una escala entre dichos valores), y con una granularidad de atributo (cada atributo se evalúa en forma independiente del resto). Para poder evaluar a un conjunto de datos bajo esta dimensión es necesario poder contrastarlos directamente contra el mundo real; debido a que no suele ser posible hacerlo (puede tratarse de datos sobre eventos pasados o correspondientes a locaciones geográficas lejanas, entre otras causas), muchas veces el contraste se realiza contra otro conjunto de datos de referencia, considerado suficientemente bueno [012][025]. Para esto es además necesario definir funciones de distancia, que permiten determinar qué tan distintos son un valor *v* considerado exacto (tal cual se presenta en el mundo real o en el conjunto de datos de referencia) y un valor *v'* que forma parte del conjunto de datos bajo análisis [012][040].

La **correctitud sintáctica (*syntactic correctness*)**, denominada por algunos autores **validez**, permite evaluar qué tan bien expresados están los datos respecto a una sintaxis predefinida [025]. Esta sintaxis puede incluir reglas relacionadas con el dominio (tipos de datos, valores válidos, vocabularios) y formatos (por ejemplo que las fechas estén expresadas en forma "día/mes/año", que los valores decimales tengan siempre dos cifras significativas, que los campos de texto empleen la codificación ISO-8859-1, etc.). En el análisis de la exactitud sintáctica no interesa determinar si un dato representa fielmente al mundo real sino que sólo importa si dicho dato es un valor aceptable [040]. Por ejemplo, para una persona tanto "Juan", "Jorge", "Jose" y "Manuel" son valores sintácticamente correctos como nombre, aunque su nombre real sea "José". Al igual que en el caso de la correctitud semántica es usual hacer la medición con granularidad de atributo, y utilizar una métrica binaria, aunque también es posible determinar un grado de desviación respecto del valor "correcto" más cercano.

La **precisión (*precision*)** permite evaluar qué tanto detalle tienen los datos [025]. Por ejemplo, para el caso de una dirección podría considerarse si el dato incluye el nombre de la calle o el código que la identifica, el número de puerta y/o apartamento, etc.; en el caso de una medida de peso puede ser presentada en gramos o en kilogramos enteros [025]. También es relevante considerar si esa información está en un formato estructurado o en cambio se encuentra inmerso en un texto sin formato: dadas dos representaciones conteniendo exactamente la misma información aquella que permita explotar los detalles con mayor facilidad puede considerarse más precisa. Por ejemplo, el texto "Av. Julio Herrera y Reissig 565, Montevideo, Uruguay, C.P. 11300" es una representación bastante completa de la dirección de la Facultad de Ingeniería de la Universidad de la República, y sin embargo los datos relevantes están embebidos en un texto sin formato que hace difícil su procesamiento automático (la misma información podría expresarse de otra forma cambiando el orden de los elementos). La medición puede hacerse con granularidad de atributo (o grupo de atributos en algunos casos), y lo más usual es utilizar una escala (posiblemente entre 0 y 1) siendo el valor asignado manualmente por expertos en el dominio.

La **consistencia interna** consta de la definición de un conjunto reglas semánticas que los datos deberían cumplir para ser considerados consistentes [012][040]. Dentro de estas reglas semánticas se pueden encontrar rangos de valores admisibles para cada uno de los atributos (por ejemplo, "la edad debe ser mayor a 18 años"). Aquellos datos que no cumplan dichas reglas presentan un problema de inconsistencia interna. En el mundo de las bases de datos relacionales a la consistencia interna se le denomina **consistencia intrarrelación** debido a que la verificación de la consistencia de los datos (tuplas) se realiza analizando cada tupla de la relación por sí misma verificando que se cumplan todas las reglas semánticas establecidas. Lo más usual es realizar la medición con granularidad de atributo, usando valores binarios donde 1 indica que el valor cumple con las reglas establecidas.

La **consistencia externa** permite determinar qué tan consistentes son los datos comparados con

otros datos. Se dice que dos observaciones son inconsistentes entre sí si en el mundo real no pueden ser ambas correctas al mismo tiempo. Por ejemplo, si por un lado se tienen datos de clientes y por otro se tienen datos sobre empleados, cuando se toma una misma persona que está en ambos conjuntos a la vez (es cliente y empleado al mismo tiempo) entonces aquellos datos en común no deben presentar contradicciones (el nombre y la fecha de nacimiento del cliente deberían ser iguales al nombre y la fecha de nacimiento del empleado). En el mundo de las bases de datos relacionales a la consistencia externa se le suele llamar **consistencia interrelación** porque generalmente verifica la consistencia entre tuplas de dos relaciones diferentes [040] (aunque podría tratarse de la misma). Al igual que en el caso de la consistencia externa lo usual es realizar la medición con granularidad de atributo, usando valores binarios donde 1 indica que el valor cumple con las reglas establecidas.

Normalmente es más difícil evaluar la correctitud semántica de un conjunto de datos que cualquiera de los otros factores de la dimensión [025]. Esto es así porque las deficiencias semánticas no son errores en sí mismos sino que son deficiencias en la comparación con la realidad (o con la fuente de referencia). Por ejemplo, si un dato indica que el nombre del socio número 12345 es “José” cuando en la realidad es “Jorge” esta discrepancia no puede detectarse simplemente observando los datos pues tanto “José” como “Jorge” son valores válidos para el nombre de una persona y solo una contrastación con la realidad puede exponer la deficiencia.

2.5.2 - Completitud

La **completitud** (***completeness***) hace referencia a qué tan completo es el modelo que se puede construir con los datos disponibles respecto de la realidad que pretende representar. Distintos autores tienen diferentes visiones sobre qué significa completitud; dependiendo de la definición de calidad de datos que se considere puede referirse al grado según el cual los datos requeridos son recabados o al grado en el que todos los casos posibles están representados [009][025]. En [040] se hace una comparación de algunas concepciones sobre la completitud. Aunque lo ideal sería conocer todo sobre todo, rara vez (o nunca) puede lograrse un juego de datos absolutamente completo [047]. Entre las causas más usuales para ello es que los datos son producidos usando juzgamientos subjetivos (las personas determinan qué es importante) lo que lleva a cometer omisiones al momento de determinar cuáles datos se recolectan, a que el acceso a ciertos datos puede estar en conflicto con los requerimientos de seguridad, privacidad o confidencialidad del entorno, y a que la falta de recursos puede limitar el acceso a los datos o la posibilidad de registrarlos en su totalidad. Se suelen considerar dos tipos de completitud o factores de calidad: la cobertura y la densidad [025].

La **cobertura (*coverage*)**, también llamada **completitud horizontal** [012] o completitud extensional, establece cuántos de todos los posibles casos observables del mundo real están representados por los datos disponibles [040]. Por ejemplo, si el universo son todas las personas que habitan una ciudad la completitud evalúa cuántas de ellas están representadas, aunque apenas sea con algunos datos mínimos de cada una. Al igual que sucede con la correctitud semántica, el análisis de cobertura implica una comparación con el mundo real, y en el caso de no ser posible se puede recurrir a un conjunto de datos de referencia, el cual en este caso podría ser construido combinando conjuntos de datos obtenidos desde diferentes fuentes. La granularidad es necesariamente a nivel de conjunto de datos, y el valor suele ser un porcentaje (a veces normalizado a la escala de 0 a 1).

La **densidad (*density*)**, también llamada **completitud vertical** [012] o completitud intensional, es una medida de cuántos datos se tiene de cada caso representado en el conjunto de datos. Por ejemplo, si de cada persona interesa su documento de identificación, nombre, dirección y teléfono,

la densidad evalúa cuántos de dichos datos se conoce de cada una de las personas representadas por los datos, independientemente de que no todas las personas lo estén. En el mundo de las bases de datos relacionales la densidad suele estar asociada a la presencia o no de **valores nulos** [040], donde un valor nulo tiene el significado de un valor faltante o inexistente; sin embargo, para caracterizar apropiadamente la densidad es importante entender por qué falta dicho valor [012][040]: puede suceder que exista pero que sea desconocido (por ejemplo, se sabe que la persona tiene un nombre pero se desconoce), que no exista (por ejemplo, si la persona está viva no tiene fecha de defunción aún) o que no se sepa si existe o no (por ejemplo, no se sabe si la persona tiene dirección de correo electrónico). La evaluación de la densidad se suele realizar a nivel de entidad, utilizando un porcentaje (a veces normalizado a la escala de 0 a 1) indicando cuántos atributos tienen valores no faltantes (es decir, contando valores no nulos). De todas formas, se debe tener mucho cuidado sobre cómo se consideran los valores faltantes ya que no siempre se trata de deficiencias (como es el caso de un valor nulo en la fecha de defunción de una persona viva), al tiempo que la ausencia de valores nulos (la existencia de valores concretos) sí podrían representar un problema de densidad (por ejemplo, si una persona viva tiene un valor no nulo como fecha de defunción) [048] (estos casos son analizados en el capítulo 5.1). Esto trae aparejada la dificultad de que no siempre se puede determinar si se está o no ante una deficiencia solamente analizando los datos disponibles. Por ejemplo, dada la escolaridad de un alumno donde para cada examen al cual se inscribió se registra la fecha de realización del examen y el resultado obtenido, se pueden tener cuatro casos:

- Se tiene la fecha y el resultado: en principio no habría ninguna deficiencia de completitud observable, salvo que el caso real fuese que el estudiante aún no ha rendido el examen y por tanto no debiera tener registrado ningún dato, pero esto no puede saberse con los datos disponibles.
- Se tiene la fecha pero no el resultado: probablemente el alumno haya rendido el examen pero aún no fue corregido, en cuyo caso no se está ante una deficiencia. Sin embargo, podría tratarse de alguna deficiencia y la calificación no haber sido registrada aunque el examen sí fue corregido, o el estudiante puede no haber rendido el examen aún.
- Se tiene el resultado pero no la fecha: seguramente se trata de una deficiencia, aunque no queda claro cuál: si el estudiante rindió el examen el atributo deficiente es la fecha, mientras que si el estudiante no lo rindió aún entonces el atributo deficiente es el resultado.
- No se tiene ni el resultado ni la fecha: lo más probable es que el alumno aún no haya rendido el examen. Podría tratarse también de algún tipo de deficiencia, aunque parece menos probable. Tampoco es posible saber cuál es el caso mediante observación de los datos.

Resulta interesante notar que es posible lograr un máximo de cobertura al mismo tiempo que se tiene un mínimo de densidad (por ejemplo, pueden tenerse registradas todas las personas pero de ellas conocer solo su identificación y su nombre) y que también puede darse el caso inverso, logrando máxima densidad con un mínimo de cobertura (por ejemplo, se tienen todos los datos requeridos pero de apenas un conjunto reducido de personas). Por otra parte, parece muy difícil lograr una buena cobertura al tiempo que se logra una buena densidad.

2.5.3 – Actualidad

La **actualidad (*currency*)** hace referencia a qué tan actuales son los datos al momento de usarlos. Una característica importante sobre los datos es que cambian con el tiempo [040], y cualquier evaluación que se haga de la calidad de los datos con seguridad variará en el futuro. Hay datos que son estables y no necesitan ser actualizados frecuentemente (por ejemplo, las fechas de nacimiento o los nombres de las personas), pero hay otros que son más inestables (como las

direcciones de las personas o los precios de los productos) [012][025]. Incluso la frecuencia de variación es relativa: para algunos datos un período determinado es corto mientras que para otros el mismo período es largo (por ejemplo, un mes es demasiado para un pronóstico meteorológico mientras que es poco para la población de una ciudad). Cuando se trata de características temporales se debe tener en cuenta que la actualidad de los datos depende tanto de la actualización de la fuente como de los retrasos en el procesamiento y la entrega [025]. Estos retrasos pueden parecer mínimos pero una demora en el procesamiento de los datos podría ser mayor que el período de actualización (por ejemplo, en un medidor automático que recoge muestras cada 1 segundo, una demora de procesamiento de medio segundo se corresponde al 50% de la vida útil de los datos). Los factores que usualmente se incluyen en esta dimensión son la frescura, la vigencia, la volatilidad y la oportunidad. En [040] los autores hacen un compendio de diferentes dimensiones o factores relacionados con la cualidad temporal de los datos.

La **frescura** (***freshness***) determina qué tan recientes son los datos: ¿cuánto tiempo hace que fueron recolectados? ¿son tan recientes como para cumplir las expectativas? ¿hay datos más recientes? [025]. La frescura ha sido identificada como uno de los atributos de calidad más importantes para los consumidores de datos, al punto que para algunos autores es el factor determinante de la actualidad [012][025].

La **vigencia** (***currency***) determina qué tan vigentes están los datos. Como hay datos que cambian muy seguido y otros que cambian más lentamente, el concepto de frecuencia es relativo. Por ejemplo, si en promedio las personas se mudan de domicilio cada diez años un registro de población creado hace menos de dos años puede considerarse relativamente vigente (serán pocas las personas cuyo domicilio haya cambiado en este tiempo) mientras que si el registro fue hecho hace cincuenta años los datos seguramente no sean nada vigentes (es altamente probable que pocas personas continúen residiendo en la misma dirección).

La **oportunidad** (***timeliness***) establece qué tan oportunos son los datos cuando se los necesita, es decir, si están disponibles en el momento en que deben ser utilizados o con cuánto retraso llegan como para ser considerados de utilidad [005][012]. Muchas veces los datos, para ser útiles, deben estar disponibles en el momento preciso en que son necesitados; por ejemplo, para tomar una decisión de inversión se debe tener los datos del mercado antes de tomarla, y retrasarla puede significar una pérdida de beneficios, mientras que conseguirlos después seguramente represente una pérdida.

La **volatilidad** (***volatility***) determina qué tan frecuentes se producen cambios en los datos [012]. Esta es una característica de los datos que se relaciona estrechamente con todas las otras temporales ya que es la que marca qué tan frescos, vigentes u oportunos pueden ser los datos: cuanto más volátiles sean (cuanto más rápido cambien) mayor será la pérdida de frescura, de vigencia y menor será la probabilidad de que sean oportunos.

La medición de la frescura, vigencia y oportunidad suele ser hecha con granularidad de entidad (a menos que se logre asociar a cada atributo individual su instante de registro, lo que es poco común), utilizando una escala porcentual (normalizada al intervalo de 0 a 1) donde 100% indica que el dato es extremadamente reciente, vigente u oportuno. La medición de la frescura y de la vigencia usualmente se determina con respecto de su instante de generación o registro (aunque la velocidad de decrecimiento puede depender de factores externos usualmente establecidos por un experto en el dominio), mientras que la medición de la oportunidad suele ser realizada por un experto en el dominio, que puede ser el usuario mismo. Por su parte la volatilidad suele medirse de acuerdo a una escala predefinida, siendo establecida por un experto o calculada en base a la frecuencia de cambios observados en los datos, pudiendo ser con cualquier granularidad, desde atributo hasta el conjunto

de datos completo.

Se puede considerar que la frescura y la oportunidad son opuestas: aunque se podría pensar que los datos más recientes (mayor frescura) son mejores también se debe considerar que si se espera a tener los últimos datos posiblemente éstos no estén disponibles en el momento en que se los necesita (menor oportunidad), en cuyo caso tal vez sea preferible utilizar datos algo más antiguos (datos con menor frescura pero con mayor oportunidad). Por su parte la vigencia y la frescura están directamente relacionadas en un único sentido ya que cuanto más frescos sean los datos mayor será la probabilidad de que sean vigentes, aunque, por el contrario, que los datos sean vigentes no necesariamente significa que sean frescos, si su volatilidad es escasa.

2.5.4 - Usabilidad

La **usabilidad** (***usability***) hace referencia a qué tan aprovechables, o utilizables, son los datos. Es la dimensión más subjetiva porque la propia idea de usabilidad implica que debe ser evaluado bajo el contexto de adecuación al uso. Está compuesta principalmente por los siguientes factores de calidad: accesibilidad, comprensibilidad, confiabilidad, utilidad y volumen.

La **accesibilidad** (***accesibility***) evalúa si los datos están disponibles para ser usados cuando se los necesita. Si los datos pudiesen reunir una hipotética excelencia de calidad según cualquier otro factor pero no se puede acceder a ellos cuando se los necesita hay una notoria deficiencia de calidad. Existen muchas razones por las cuales los datos podrían no estar accesibles: porque no existen (no se recabaron mientras se podía), porque no están almacenados (se recabaron pero no se almacenaron), porque están protegidos (no se tiene permiso para utilizarlos), por razones operativas (no se sabe cómo acceder a los datos o la tecnología disponible no es la apropiada), etc. Los enfoques convencionales tratan a la accesibilidad como un tema técnico de los sistemas informáticos y no como algo relacionado con la calidad de datos [013]. Por otra parte, la accesibilidad a menudo está en conflicto con la seguridad: los datos de acceso restringido son vistos por los usuarios como datos no accesibles o con accesibilidad limitada y las medidas que se toman para garantizar su seguridad o integridad son percibidas como barreras para el acceso y por lo tanto como problemas de calidad [007][013][020]. La medición de la accesibilidad suele ser realizada con granularidad de conjunto de datos, en una escala de 0 a 1 donde 1 indica que los datos son perfectamente accesibles; la evaluación debe ser realizada por los usuarios finales de los datos.

La **comprensibilidad** (***understability***) evalúa qué tan fáciles de entender son los datos para que puedan ser aprovechados. Un dato del cual no se pueda entender su significado (porque no es suficientemente descriptivo o que no está adecuadamente documentado) es poco aprovechable [017]. Por ejemplo, de nada sirve saber que para el registro cuyo identificador es '3' el valor del atributo 'A' es '5.1' si no se sabe qué representan el atributo y el valor. Nuevamente este es un factor dependiente del contexto ya que algunos usuarios expertos podrían tener bien claro el significado de los datos pudiendo comprenderlos, mientras que para otro sería información críptica. La medición de la comprensibilidad suele ser realizada con granularidad de atributo, o conjunto de atributos, en una escala de 0 a 1 donde 1 indica que el dato es perfectamente comprensible; la evaluación debe ser realizada por los usuarios finales de los datos.

La **confiabilidad** o credibilidad (confiability, ***reliability***, believability) evalúa qué tan confiables son los datos, es decir, que tan verdaderos y creíbles resultan para el usuario [005]. No todos los datos son igualmente confiables, dependiendo de la fuente y otras características. Por ejemplo, probablemente resulte más confiable un registro sobre personas creado por un organismo oficial que uno realizado a través de encuestas telefónicas, de igual manera que puede resultar más confiable un registro geológico actualizado en 2014 que uno hecho en 1966. La confiabilidad de los datos es

también extremadamente subjetiva ya que la evaluación la deben realizar los usuarios, muchas veces en base a su experiencia. La medición de la confiabilidad suele ser realizada con granularidad de conjunto de datos, en una escala de 0 a 1 donde 1 indica que los datos son completamente confiables; la evaluación debe ser realizada por un experto en el dominio.

La **utilidad** (***usefulness***) evalúa qué tan útiles son los datos disponibles para desarrollar la tarea que se pretende realizar. Este es, junto con la confiabilidad, el más subjetivo de todos los factores de calidad de datos ya que solo la persona que deba usar los datos para una tarea puntual puede juzgar qué tan útiles le resultan: podrían ser muy útiles para una tarea particular pero poco útiles para otra, incluso podrían resultar con diferentes grados de utilidad para la misma tarea a personas diferentes según los conocimientos y habilidades de cada una. La medición de la utilidad suele ser realizada con granularidad de conjunto de datos, en una escala de 0 a 1 donde 1 indica que los datos son completamente útiles; la evaluación debe ser realizada por los usuarios finales de los datos.

El **volumen** (***amount***) evalúa la cantidad de datos disponibles. No debe confundirse volumen con completitud: mientras la completitud evalúa cuántos de todos los datos posibles están representados, el volumen solo considera la cantidad por sí misma. Tener en cuenta el volumen es importante porque permite realizar consideraciones sobre lo adecuado que pueden resultar los datos: si bien podría pensarse que cuantos más datos se tengan disponibles es mejor, la realidad puede mostrar lo contrario, ya que grandes volúmenes de datos pueden hacer que el acceso a ellos sea difícil en un tiempo razonable [007], requiriendo más espacio de almacenamiento y mayor poder de procesamiento. Además, cuanto más datos se tengan, mayor es la probabilidad de encontrar otros tipos de deficiencias, como duplicados, inconsistencias, etc. La medición del volumen de datos suele ser realizada con granularidad de conjunto de datos, en una escala de 0 a 1 donde 1 indica que el conjunto de datos tiene un tamaño ideal para su aprovechamiento; la evaluación debe ser realizada en conjunto por los usuarios finales de los datos, expertos en el dominio y el personal técnico encargado de su cuidado, ya que todos ellos pueden tener visiones diferentes respecto del factor.

En la actualidad es una práctica común recolectar grandes volúmenes de datos que no se necesitan esperando que algún día alguien pueda usarlos para algo, asumiendo que es más barato hacerlo desde el principio que incorporarlos después. Sin embargo, se puede comprender que si los datos no son usados entonces con el tiempo los errores en la captura de los cambios del mundo real serán ignorados, lo que los científicos en el área biológica denominan atrofiamiento ("*úsalo o piérdelo*") [020]. Si nadie usa los datos entonces los sistemas se vuelven insensibles a ellos, incluyendo a los usuarios: en la década de 1970 Ken Orr estableció, conjuntamente con otros colegas, que los sistemas que recolectaban datos innecesarios tendrían problemas de calidad en el futuro y sin embargo la práctica ha perdurado en el tiempo [020].

2.6 - Tratamiento de la calidad de los datos

El tratamiento de la calidad de los datos debería ser considerada una tarea específica dentro de un proceso mayor. Este proceso, que es conocido bajo nombres diferentes tales como **aseguramiento de la calidad** de datos, **control de la calidad** de datos, o **gestión de la calidad** de datos, debería ser considerado una operación propia de cualquier organización que utilice datos (todas las organizaciones actuales). El tratamiento de la calidad de datos no debe perseguir únicamente el objetivo de mejorar los datos sino que debe estar enfocado en construir un perfil de los mismos, descubriendo y documentando sus fortalezas y debilidades, de forma tal de que cuando se pretenda hacer uso de ellos se pueda establecer una medida de confianza a su respecto. Esto significa que el interés real del tratamiento de la calidad de datos no radica en lograr datos de

calidad perfecta sino en lograr datos de calidad adecuada según el contexto, conociendo también sus debilidades [020].

Antes de intentar medir la calidad de los datos utilizando algún conjunto de dimensiones es necesario establecer las reglas de calidad contra las cuales los datos serán evaluados [039]. Es imposible hacer mediciones sobre la calidad de algo si previamente no se establece qué se debe medir y cómo se debe hacerlo [027]. Además, por ser algo subjetivo, no puede tratarse la calidad de los datos en forma independiente de los usuarios [015]. Una tarea importante para el tratamiento de la calidad de los datos es determinar cuáles son los datos que merecen el esfuerzo de ser medidos y mejorados [033]: ¿son preferibles los datos que son usados para mayor cantidad de tareas o los datos que son usados para tareas más importantes? Además, debe distinguirse entre lo que es vital y lo que es meramente deseable.

La calidad de un producto manufacturado normalmente depende tanto de las materias primas utilizadas como del proceso por el cual dicho producto es diseñado y fabricado. De la misma manera la calidad de los datos depende de los procesos de diseño y producción involucrados en su generación [021][037]. Existen interesantes analogías entre los problemas de calidad en los procesos de manufacturación y en los de un sistema de información: los primeros pueden verse como un proceso que toma materias primas como entrada y producen un producto como salida, mientras que un sistema de información toma datos de entrada y produce más datos como salida, a veces llamada información [022][042]. El control y la gestión de la calidad es algo necesario para cualquier negocio actual, y hay una rica experiencia de décadas que podría ser aprovechada en el tratamiento de la calidad de los datos [022].

Uno de los grandes problemas que enfrenta el tratamiento de la calidad de datos es que las técnicas tradicionales son consumidoras de tiempo y recursos, con un cubrimiento limitado de datos, y pueden no detectar ciertas deficiencias. La forma de solventar eso es desarrollando metodologías que permitan establecer, documentar y automatizar el tratamiento de la calidad de los datos [049]. Una metodología para el tratamiento de la calidad de datos es un conjunto de lineamientos y técnicas que apuntan a evaluar y mejorar la calidad de los datos [040]. Estas metodologías se pueden dividir en dos categorías: las orientadas a los procesos y las orientadas a los datos [015][040].

- Las **metodologías orientadas a los datos** (***data-driven***) se proponen mejorar la calidad de los datos alterándolos, ya sea remplazándolos por otros de mejor calidad, corrigiendo valores incorrectos, completando valores faltantes, o eliminando los datos deficientes. De esta manera permiten atacar diferentes problemas como la falta de completitud, las inexactitudes y las inconsistencias [015]. Para ello utilizan técnicas tales como la adquisición de datos nuevos (actualización), la normalización o estandarización (remplazar los valores no estandarizados por otros estandarizados), el reconocimiento de entidades (identificar cuándo dos o más observaciones son en realidad la misma con variantes), la integración de esquemas (unificar las vistas de los datos provenientes de diferentes fuentes heterogéneas), etc. [040].
- Las **metodologías orientadas a los procesos** (***process-driven***) se proponen mejorar la calidad de los datos alterando los procesos que los crean y modifican, permitiendo la identificación de las causas de las deficiencias en los datos a través de todo el proceso de adquisición y tratamiento [015]. Muchos autores ponen énfasis en el valor de la mejora de los procesos antes que tratar de solucionar los problemas alterando a los datos, ya que en el segundo caso es probable que las deficiencias vuelvan a presentarse nuevamente una vez solucionadas [015]. Sin embargo, no es eficiente solo centrarse en los procesos por varias

razones: por un lado, es difícil de hacer y bastante costoso en tiempo y recursos, requiriendo verificaciones y cambios frecuentes de todos los procesos y subprocesos de la organización, y por otro, es difícil alcanzar un consenso entre todos los usuarios de datos sobre qué es lo que se debe mejorar, para qué y cómo [015]. Este tipo de metodologías se caracterizan por hacer uso de dos estrategias: el control de los procesos (inserción de procedimientos de verificación y control cuando los datos son creados, modificados y accedidos) para evitar la degradación de la calidad con el tiempo, y el rediseño de los procesos (análisis y eliminación de las causas que degradan la calidad) para quitar o redefinir aquellos pasos que son más propensos a permitir la introducción de deficiencias [040].

Los dos tipos de metodologías deben complementarse para realizar un tratamiento de la calidad de datos apropiado. Las orientadas a los datos tienen grandes ventajas respecto a las orientadas a los procesos porque pueden ser automatizadas y enfocadas en las necesidades puntuales de los usuarios, además de que las orientadas a procesos dependen mucho del esfuerzo humano, lo que aumenta los costos en tiempo y dinero. Sin embargo, las orientadas a los datos no permiten capturar las causas de los problemas por lo que no permite erradicarlas. A largo plazo, las orientadas a procesos son mejores debido a que permiten eliminar las causas de los problemas de calidad aunque a corto plazo el rediseño de los procesos puede ser muy costoso, lento y encontrar oposición por parte de los involucrados. Por el contrario, las orientadas a datos suelen ser más efectivas a corto plazo pero más costosas a largo plazo ya que deben ejecutarse continuamente [040].

En los últimos años se han propuesto muchas metodologías para evaluar y mejorar la calidad de los datos, principalmente en base a métodos tales como encuestas y entrevistas [015]. Desde que la recolección automatizada de datos se ha vuelto algo común el volumen de datos producido se ha elevado exponencialmente. Para que estos datos puedan ser reutilizados y compartidos es crucial desarrollar técnicas automatizadas para evaluar su calidad [050]. El aumento de la automatización mejora la productividad [051].

Entre las múltiples metodologías de aseguramiento de la calidad de datos se destacan TDQM, AIMQ, DQA, DaQuinCIS y CDQ [040]**. TDQM** ([052]) fue la primera metodología general, propuesta inicialmente por Richard Wang en 1998 y en la actualidad es ampliamente adoptada como una guía, con el objetivo de extender los principios de *Total Quality Management* (TQM) al abordaje de la calidad de los datos. **AIMQ** ([045]) es una metodología basada en mediciones y comparaciones, fundamentada en una tabla que permite clasificar las dimensiones desde la perspectiva de los administradores y de los usuarios finales de los datos, distinguiendo cuatro clases de dimensiones: probable (*sound*), confiable (*dependable*), útil (*useful*) y usable (*usable*); para conformar la tablas se utilizan formularios en fases iterativas, primero detectando las dimensiones relevantes y luego midiendo la calidad de los datos bajo ellas según la opinión de los usuarios. **DQA** ([040]) hace una distinción entre métricas de calidad subjetivas y objetivas, y a su vez las objetivas son clasificadas en dependientes o independientes de las tareas. **DaQuinCIS** ([053]) apunta a evaluar la calidad de los datos en ambientes colaborativos donde se presentan otros problemas como la confiabilidad interinstitucional; esta metodología introduce el concepto de certificación de los datos, mediante el cual se asocia a ellos sus correspondientes medidas de calidad, que pueden ser intercambiadas y comprendidas por las diferentes organizaciones. **CDQ** ([054]) está compuesto por tres etapas: reconstrucción del estado, evaluación y proceso de mejora; en la primera se reconstruyen las relaciones entre las unidades organizacionales, los procesos, los servicios y los datos y se modelan con una matriz que describe cuál unidad utiliza cuáles datos y qué rol tiene en los procesos; en la segunda se establecen los objetivos de calidad y se evalúan costos y beneficios; y en la tercera se propone la secuencia de actividades que tienen la mejor relación costo/beneficio.

La calidad de los datos no se puede medir, mejorar una vez y mantenerla para siempre. Por el contrario, se va degradando con el tiempo si tanto los procesos como los datos ingresados no son controlados (y mejorados) apropiadamente [031]. Por esta razón el tratamiento continuo de la calidad de los datos es fundamental.

El tratamiento continuo de la calidad de los datos es un proceso iterativo que está compuesto típicamente por siete actividades [039]:

1. Identificar cuáles datos necesitan ser evaluados: determinar cuáles son los datos que tienen mayor impacto en las operaciones de negocio o en la toma de decisiones.
2. Seleccionar cuáles dimensiones de calidad de datos son relevantes, y en qué medida.
3. Para cada dimensión de calidad de datos seleccionada establecer los valores mínimos aceptables que marquen la frontera entre datos de buena y mala calidad.
4. Aplicar los criterios de evaluación de calidad que correspondan a cada dimensión y determinar el valor correspondiente a cada una.
5. Evaluar los resultados y determinar si la calidad de datos es aceptable o no.
6. Tomar acciones correctivas, tales como limpiar los datos, incorporar otras fuentes, y/o mejorar el proceso de recolección y tratamiento de datos.
7. Repetir los pasos anteriores en forma periódica para monitorear la evolución de la calidad de los datos, teniendo en cuenta que con el tiempo los requerimientos pueden cambiar.

Los beneficios del monitoreo continuo de la calidad de los datos son numerosos. Debido a que los datos están en constante cambio, las deficiencias pueden reaparecer luego de haber logrado una calidad de datos aceptable [009]. El agregado de nuevos datos, las actualizaciones de software, los cambios en el personal, entre otros factores, pueden conducir a una degradación de la calidad de los datos. Realizar la evaluación continua de la calidad de los datos permite detectar tendencias en la introducción de deficiencias, permitiendo mejorar el proceso de recolección de datos o ajustar los sistemas con los cuales se realiza, previniendo los errores antes de que sucedan en lugar de corregirlos después. La decisión sobre qué tan seguido realizar el control depende del uso que se le de a los datos y de su variabilidad [009], siendo probable que algunos datos deban ser controlados más frecuentemente que otros.

Uno de los principales problemas relacionados a la calidad de datos es lograr que los usuarios (los que los ingresan, los que los mantienen y los que los utilizan) comprendan los fundamentos de la gestión de la calidad de datos. Para que un programa de calidad de datos tenga éxito a largo plazo es necesario dedicar una cantidad significativa de tiempo a entrenar a los usuarios [020].

Todo lo mencionado en los párrafos anteriores lleva a concluir que el tratamiento de la calidad de los datos es una tarea que debe ser planificada cuidadosamente, considerando los objetivos organizacionales y los recursos disponibles, y algo no menor, debe ser abordado por personas con las capacidades y habilidades apropiadas, con las herramientas adecuadas [035].

3 - Minería de datos

Este capítulo se centra en el segundo de los grandes temas abordados: la minería de datos. Se realiza primero una introducción sobre qué es y para qué es útil, para luego mostrar aplicaciones, describir los objetivos y analizar algunas de las técnicas más comúnmente utilizadas. Debe tenerse en cuenta que este capítulo aborda la minería de datos como disciplina en sí misma, en forma independiente a la calidad de datos. El capítulo 4 aborda la integración de la minería de datos como herramienta para realizar la gestión de la calidad de los datos.

3.1 - Qué es la minería de datos

La **minería de datos** (***data mining***) es un conjunto de técnicas orientadas al descubrimiento de información en grandes volúmenes de datos o en conjuntos de datos complejos que pueden ser difíciles de analizar con las técnicas usuales [055][056][057], en lo posible con la menor intervención humana, en especial de expertos [056]. A grandes rasgos, es una combinación de la estadística, la lógica, la inteligencia artificial, el aprendizaje automático y los sistemas de gestión de información, entre otras áreas de conocimiento [058]. Se diferencia del análisis de datos clásico en que por el gran volumen de datos involucrado, y la complejidad de los datos, la observación directa y las técnicas estadísticas tradicionales no son aplicables [059][060]; de hecho, a veces la minería de datos es considerada como la capa intermedia entre la computación y la estadística [061].

El término minería de datos fue utilizado por primera vez por los estadísticos para referirse al hecho de extraer información que no estaba explicitada directamente por los datos [062]. A pesar de que sus raíces se remontan a finales de la década de 1980, en la década de 1990 aún estaba pobremente desarrollada: durante una conferencia llevada a acabo en 1993, en Viena, Austria ("*The Vienna Update*") [063], se les pidió a 5 reconocidos investigadores que indicaran cuáles áreas de investigación les parecían más interesantes. Entre los primeros lugares figuró la minería de datos (junto a las interfaces de usuario y las aplicaciones multimedia), siendo la principal motivación su aplicabilidad en el campo de las transacciones comerciales; apenas cuatro años antes se había hecho algo similar en Laguna Beach, Estados Unidos, y la minería de datos no había figurado en absoluto [063].

Aunque la idea de extraer información valiosa a partir de los datos no es nueva, la innovación radica en las oportunidades que ofrecen los avances tecnológicos proporcionados por diversas disciplinas como la informática y la inteligencia artificial [064], incluyendo el aumento del poder computacional de los equipos en varios de órdenes de magnitud.

Normalmente en todas las áreas de investigación, ya sean científicas, económicas o sociales, los investigadores suelen implementar alguna técnica que se emparenta con el método científico clásico compuesto de cuatro etapas: observación, hipótesis, experimentación y conclusión. El punto crítico de esta metodología para obtener conocimiento nuevo es la formulación de la hipótesis, la cual se genera usualmente a partir de la observación; esto depende mucho de las personas, ya que requiere que éstas estén atentas a todos los detalles, además de que las interacciones existentes han de ser tantas y tan complejas que puede resultar imposible detectar todos los posibles estados o características de la realidad estudiada. Es en estos casos en que la minería de datos es útil: las técnicas de minería de datos aplicadas pueden *sugerir* las hipótesis a partir de los datos en forma automatizada. Luego los investigadores deben formalizar las hipótesis y tratar de corroborarlas

mediante experimentos.

A menudo sucede que se cuenta con muchos datos sobre un dominio particular pero no se tiene muy en claro qué significan ni cómo utilizarlos. Por ejemplo, una base de datos donde se registran todas las ventas realizadas por una tienda comercial puede esconder información útil, especialmente cuando no se la busca en forma explícita: correlaciones entre productos vendidos (del tipo "*cuando los clientes compran el producto A es usual que también compren el producto B*"), o entre productos y eventos o períodos particulares (como "*los fines de semana el producto C es más vendido que durante la semana*"). Esta información no suele ser proporcionada por los sistemas informáticos tradicionales, los cuales están diseñados para realizar tareas operativas como registrar las ventas y el permitir consultar el *stock* de productos; también es muy difícil deducirla mediante observación directa de los datos debido a su volumen y la forma en que están estructurados. Las bases de datos transaccionales (que son del tipo OLTP, o*n line transaction processing*, procesamiento de transacciones en línea) están diseñadas para soportar muchas lecturas y escrituras concurrentes, y manejar grandes volúmenes de datos, pero debido a su diseño no son eficientes para analizarlos. Para esto se han diseñado otros tipos de bases de datos (del tipo OLAP, o*n line analytical processing*, procesamiento analítico en línea), conocidas como ***data-warehouses*** (almacenes de datos), optimizadas para realizar análisis de datos más complejos. Estas bases de datos reducen el número de correlaciones (*joins*) necesarias al costo de aumentar aún más el volumen de datos almacenados y admitir redundancia (controlada). Los sistemas OLTP y los OLAP tienen objetivos diferentes y por lo tanto no parece factible concebir un sistema que se adecue a todos los propósitos [058]. Así como los sistemas transaccionales son eficientes para almacenar y recuperar datos en forma repetida pero no son eficientes para realizar el análisis de dichos datos, los *data-warehouses* son ideales para realizar análisis de datos aunque hacen más difícil la incorporación, el mantenimiento y la recuperación de datos. Los *data-warehouses* son ideales para la aplicación de la minería de datos.

La minería de datos busca patrones o modelos ocultos en grandes volúmenes de datos [062]. Un patrón es una expresión genérica de un subconjunto de datos [065]. Un modelo es una representación de las observaciones hechas del mundo real como resultado de aplicar algoritmos para identificar patrones en los datos [058]. Los patrones encontrados deben ser interpretados antes de poder ser llamados conocimiento [056]. Los modelos pueden ser expresados como conjuntos de reglas del tipo *{x → y}* (que se interpreta como "*si ocurre el evento x es probable que también ocurra el evento y*"), árboles de decisión, cuadros de doble entrada, etc.

Son requerimientos importantes de la minería de datos que los patrones descubiertos sean válidos, comprensibles e interesantes [065][066]. Pero mientras hay concordancia en la literatura sobre el significado de válido (que los patrones representen adecuadamente a la realidad) y de comprensible (que la forma de representarlos pueda ser entendida por las personas) no lo hay para el significado de interesante, porque lo que es interesante para unas personas en un determinado contexto podría no serlo para otras en otro contexto diferente [067]. Esto indica que son los expertos en el dominio y no los expertos en las técnicas de minería de datos quienes deben determinar si los patrones encontrados son interesantes o no [060].

Los métodos de minería de datos constan principalmente de tres componentes [068]:

- **Un modelo**, el cual está determinado por el objetivo del método (qué se pretende modelar) y su representación (cómo se pretende modelarlo: árboles de decisión, redes neuronales, grafos, etc.). Generalmente está dado por una forma básica conocida pero parametrizada, debiéndose determinar los valores apropiados de los parámetros a partir de los datos en forma automática.

- **Un criterio de evaluación**, que permita dar preferencia a un conjunto de parámetros por sobre otros. Muchos métodos funcionan encontrando múltiples juegos de parámetros para luego aplicar un criterio que seleccione el más apropiado.
- **Un algoritmo de búsqueda**, encargado de encontrar el modelo y/o los valores para los parámetros a partir de los datos, dada la forma del modelo buscado y el criterio de preferencia.

3.2 - Descubrimiento de conocimiento en bases de datos

La minería de datos es en realidad solo un paso en un proceso más grande denominado descubrimiento de conocimiento en bases de datos (**KDD**, k*nowledge discovery in databases*), que consiste en aplicar análisis de datos y algoritmos para encontrar patrones o modelos interesantes y novedosos sobre un conjunto de datos [057][060][065][069]. El principal problema al que apunta KDD es transformar los datos crudos y voluminosos para ser comprendidos por sí mismos en alguna otra forma de representación más compacta (resumida), más abstracta (con menos detalle), y más útil (aplicable) [065]. El método tradicional de transformar datos en conocimiento se ha basado en el análisis y la interpretación asistida por computadoras, incluyendo la visualización de los datos, pero a medida que el volumen de datos crece el procedimiento manual se vuelve impracticable. KDD intenta atacar el mayor problema de la era de la información digital: la sobrecarga de datos [065]; para ello en el núcleo del proceso se encuentra la minería de datos. Pero para que las técnicas de minería de datos sean capaces de brindar resultados útiles los pasos preliminares de preparación de la información y los posteriores de verificación son imprescindibles.

Desde fines de la década de 1990 se han desarrollado varias metodologías de KDD, siendo la principal diferencia entre ellos el número de pasos y el alcance de cada uno. Los dos primeros procesos formales fueron propuestos en 1996 por Usama Fayyad, Gregory Piatetsky-Shapiro y Padhraic Smyth ([070]), compuesta por nueve pasos, y por Ronald Brachman y Tej Anand ([071]), compuesta por ocho ([069]), ambos iterativos. Con el tiempo, múltiples autores han participado en la pseudo-estandarización del proceso que en la forma más general está compuesto por ocho pasos [015][057][060][065][068]:

1. Comprender el dominio, recabar el conocimiento existente e identificar el objetivo del proceso de KDD.
2. Seleccionar los datos y las variables con las cuales se va a trabajar.
3. Limpiar los datos, eliminando el ruido, determinando cómo tratar los valores faltantes, etc.
4. Reducir y proyectar los datos, quitando los atributos y las muestras que no resultan de interés.
5. Seleccionar el método de minería de datos más apropiado según el objetivo establecido y los datos disponibles.
6. Ejecutar el algoritmo de minería de datos seleccionado, ajustando los parámetros en la forma más conveniente.
7. Interpretar los resultados. Si es necesario volver a alguno de los pasos anteriores. No debe perderse de vista que determinar si un modelo encontrado al ejecutar el algoritmo de minería de datos se ajusta o no al mundo real es parte del proceso de KDD, y por lo tanto requiere un juzgamiento humano que es inherentemente subjetivo [068].
8. Representar, verificar y usar los resultados.

Aunque la mayor parte de la literatura se enfoca en el paso de la ejecución del algoritmo de minería de datos, en general ocurre que los pasos anteriores son los que más tiempo consumen [072]. Por otra parte, se debe tener en cuenta que no existe un método de minería de datos universal que pueda ser aplicado en cualquier caso, con cualquier objetivo, y con cualquier juego de datos. En ocasiones elegir el mejor método es un arte y de hecho una gran parte del esfuerzo suele ser dedicado a plantear correctamente el problema, y no a aplicar un algoritmo de minería de datos [065].

Si bien el proceso KDD surgido en el ámbito académico parece bastante bien definido, en realidad son solo lineamientos básicos sobre como implementar un proyecto de descubrimiento de información, casi siempre llevado a la práctica con esfuerzos individuales y de forma artesanal. Desde la industria también existía el interés por el descubrimiento de información (en contextos concretos) y fue así como varias organizaciones, entre ellas Teradata, ISL (actualmente SPSS), Daimler-Chrysler y OHRA, se unieron para definir una metodología estandarizada, dando como resultado CRISP-DM (c*ross industry standard process for data mining*) [073]. Esta metodología se compone de una serie de tareas descritas con cuatro niveles de abstracción (Figura 2): en el nivel superior el proceso se descompone en fases, cada una de ellas compuesta por un conjunto de tareas genéricas (segundo nivel), las cuales a su vez se descomponen en tareas especializadas (tercer nivel); el cuarto nivel incluye las acciones, decisiones y resultados [057].

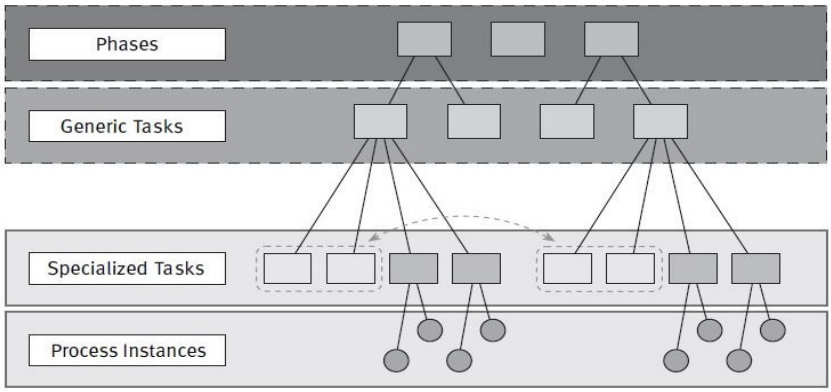

Figura 2: Metodología de CRISP-DM (tomada de [073])

CRISP-DM consta de 6 fases, aunque la secuencia no es estricta, sino que depende del resultado de la ejecución de cada fase [057][069][073]:

- **Entendimiento del negocio**: comprender los objetivos del proyecto y los requerimientos desde el punto de vista del negocio. Definir claramente el proyecto de minería de datos y construir un plan general de ejecución.
- **Familiarización con los datos**: recolectar datos, estudiarlos, identificar sus características y determinar sus problemas de calidad; formular algunas hipótesis sobre posible conocimiento escondido.
- **Preparación de los datos**: incluye a todas las actividades necesarias para construir el conjunto de datos con el cual se trabajará, tales como seleccionar cuáles datos serán usados, cuáles atributos son importantes, y realizar tareas de limpieza y transformación.
- **Construcción del modelo**: seleccionar uno o varios modelos, y determinar sus correspondientes parámetros óptimos. A menudo esto implica utilizar diferentes técnicas

para construir múltiples modelos diferentes. Esto comprende también la selección de una técnica de minería de datos y su ejecución.

- **Evaluación del modelo**: evaluar el modelo construido; puede incluir también revisar las tareas ejecutadas anteriormente para determinar si se cumplieron los objetivos. El objetivo final de esta fase es determinar si los resultados alcanzados pueden ser utilizados o no.
- **Puesta en producción**: la construcción del modelo nunca debe ser el objetivo del proceso, sino que el objetivo final debe ser la puesta en producción, es decir, utilizar el modelo para cumplir los objetivos planteados. Esto incluye presentar el conocimiento descubierto en una forma que sea comprensible por los usuarios finales. También puede incluir una definición de un proceso que pueda aplicarse en forma repetitiva.

La Tabla 1 (tomada de [057]) muestra una comparativa entre KDD y CRISP-DM.

KDD	**CRISP-DM**
Comprensión del dominio	Entendimiento del negocio
Selección los datos y las variables	Familiarización con los datos
Limpieza de los datos	Preparación de los datos
Reducción de datos	
Selección de la técnica de minería de datos	Construcción del modelo
Ejecución de la minería de datos	
Interpretación de los resultados	Evaluación del modelo
Uso de los de los resultados	Puesta en producción

Tabla 1: Comparación entre KDD y CRISP-DM, tomada de [057]

Además de CRISP-DM existen otras propuestas para estandarizar el proceso de descubrimiento de conocimiento, tales como *Two Crows* (1998), Cios (2000), RAMSYS (2002), DMIE (2002) y SEMMA (2005). Sin embargo, en la actualidad CRISP-DM es la más utilizada e incluso se considera el estándar de hecho [057]. El sitio web KDNuggets publicó en 2014 los resultados de una encuesta ([074]) sobre cuál es la metodología utilizada para llevar adelante proyectos de análisis y minería de datos en las organizaciones, encontrando que un 43% de los consultados dijo utilizar CRISP-DM, un 8.5% dijo usar SEMMA y un 7.5% dijo seguir la metodología básica del proceso KDD, mientras que el restante 41% se distribuyó entre quienes afirmaron utilizar una metodología propia, otras metodologías, o ninguna. En 2007 se publicó una nueva versión de CRISPM-DM, llamada CRISP-DM 2.0, que incorpora los nuevos requerimientos que han ido surgiendo con el avance de las tecnologías, como el uso de fuentes de datos no estructuradas (o semi estructuradas, entre ellas texto, web, XML), la integración con otros sistemas operacionales, y la adaptación a los requerimientos de tiempos y recursos, entre otras [057].

3.3 - Fundamentos de la minería de datos

Las técnicas de minería de datos se dividen en dos grupos según el objetivo que persiguen: técnicas descriptivas y técnicas predictivas [058][064]. Las **técnicas descriptivas** son aquellas que tienen como objetivo construir un modelo a partir de un conjunto de datos para tratar de describir el mundo real al cual corresponden dichos datos. Por ejemplo, se puede buscar agrupar a los clientes en grupos de afinidad según sus características para luego preparar ofertas especialmente llamativas a cada uno de esos grupos (sin tener bien claro de antemano cuáles son las características relevantes). Las **técnicas predictivas** son aquellas que tienen como objetivo construir un modelo a partir de un conjunto de datos para tratar de predecir cómo se comportará el mundo real bajo determinadas condiciones. Por ejemplo, en meteorología se tienen registros sobre qué ocurrió cuando se observaron ciertas combinaciones de factores meteorológicos y se pretende anticipar qué pasará en un futuro cercano cuando en el presente se observan determinados comportamientos.

Otra forma de clasificar las técnicas de minería de datos es según su mecanismo de operación, dividiéndose en técnicas supervisadas y técnicas no supervisadas [059]. Las **técnicas supervisadas** son aquellas que para cumplir con su cometido deben ser aplicadas en dos etapas: en la primer etapa, llamada **entrenamiento**, se ejecuta un algoritmo utilizando un conjunto de datos especialmente preparado, llamado **datos de entrenamiento**, para construir un modelo, y que en la segunda etapa se utiliza dicho modelo para analizar datos no conocidos previamente, llamados **datos objetivo**. Las **técnicas no supervisadas** son aquellas que no requieren un entrenamiento previo sino que actúan desde el principio con un conjunto de datos objetivo. Existen también algunas técnicas denominadas **semi-supervisadas** que combinan las anteriores, utilizando una técnica no supervisada para descubrir determinada información que luego es alimentada a una técnica supervisada para completar el descubrimiento.

Si se consideran las dos formas de clasificación se puede observar que las técnicas predictivas requieren un entrenamiento para construir el modelo que luego será utilizado para predecir nuevos casos. Esto implica que en general las técnicas predictivas son también supervisadas. Por el contrario, se observa que las técnicas descriptivas no requieren entrenamiento (puesto que no suele conocerse mucho sobre los datos antes de comenzar) por lo que en general son también técnicas no supervisadas. De todas formas, los límites no siempre están bien definidos y frecuentemente se encontrarán casos que comienzan utilizando un tipo de técnica para luego migrar a otra. Por ejemplo, luego de utilizar una técnica descriptiva para comprender mejor el mundo real, y luego de validar el modelo (determinar si lo que el modelo describe efectivamente se cumple en los datos conocidos), puede pasarse a una técnica predictiva, asumiendo que el modelo construido anteriormente fue el fruto de un “entrenamiento”.

La minería de datos puede ser aplicada con diferentes objetivos, según el interés que se tenga respecto de cómo utilizar los datos disponibles. Entre esos múltiples objetivos, son tres los que se destacan: la **clasificación de datos**, la **agrupación de datos** y la **búsqueda de reglas de asociación**. A continuación se describen cada uno de estos tres objetivos principales, aunque es importante aclarar que no son las únicas.

3.3.1 - Clasificación

La **clasificación**, también llamada **aprendizaje supervisado** (*supervised learning*), deriva del aprendizaje automático. Es una técnica supervisada porque usa datos de entrenamiento para construir un modelo, llamado **modelo predictivo**, que permite clasificar a las observaciones (datos) en un conjunto de categorías preestablecidas llamadas **clases** [015][059][075]; dicho modelo puede

ser usado luego para hacer predicciones sobre nuevos datos cuya clase se desconoce, por lo que también es una técnica predictiva. Generalmente se llama **clasificación** cuando la clase es un valor discreto y **regresión** cuando la clase es un valor numérico continuo, aunque se considera al término clasificación en forma genérica [064][065][076].

Las técnicas de clasificación toman como entrada un conjunto de datos, llamado **conjunto de entrenamiento** (*training set*), compuestos por múltiples atributos (o características) llamados **clasificadores** (*classifiers*) y anotados con una etiqueta llamada **clase** (*class*). El objetivo es analizar dicho conjunto de entrenamiento y producir una descripción precisa de cada una de las clases conocidas en base a los clasificadores (por ejemplo, "*si el atributo A tiene el valor a_1 y el atributo B el valor b_1, entonces la clase C debería ser c_1*"). Luego dicha descripción puede ser usada para clasificar nuevos ejemplos para los cuales se desconoce la clase correspondiente [077]. A modo de ejemplo, dado un conjunto de pacientes que presentan ciertos síntomas y son diagnosticados con determinadas enfermedades, un objetivo de la minería de datos sería construir un modelo que permita asociar la presencia o ausencia de los síntomas con cada una de las enfermedades; de esta manera, cada vez que un nuevo paciente se presenta se aplicaría dicho modelo para realizar un diagnóstico primario automático en base a los síntomas que declara.

Existen muchos algoritmos de clasificación que se diferencian principalmente en la forma de construir el modelo y en la representación del mismo. Entre ellos se destacan los árboles de decisión, las redes bayesianas, las redes neuronales, los algoritmos evolutivos, el *boosting* adaptativo y las máquinas de soporte vectorial [075][076]. Todos ellos tienen en común que constan de tres etapas: **aprendizaje**, usando el conjunto de datos de entrenamiento (construcción del modelo), **validación** del modelo, y **clasificación** de nuevos datos no conocidos previamente. Al final del proceso de entrenamiento el modelo construido debería producir las salidas correctas para los datos de entrenamiento pero también debería ser capaz de generalizar adecuadamente para otros datos no vistos aún. Una pobre generalización puede ser caracterizada por el sobreajuste a los datos de entrenamiento, donde el modelo solo "recuerda" los ejemplos proporcionados y no es capaz de producir salidas correctas para datos nuevos [078].

La validación del modelo es una parte muy importante del proceso de construcción. Esta validación consiste en aplicarlo sobre un conjunto de datos del cual se conoce su clase, y determinar en cuántos casos la clasificación realizada por el modelo coincide con la clase correcta.

Por lo general es deseable que el conjunto de datos de validación sea diferente al utilizado para el entrenamiento, pero esto no siempre es posible ya que el hecho de requerir que las observaciones se encuentren previamente clasificadas exige un trabajo importante (muchas veces la cantidad de observaciones clasificadas previamente y con certeza correctas son escasas). Para los casos en los cuales no se tiene un conjunto de datos de validación se puede recurrir a algunas técnicas conocidas:

- Utilizar como conjunto de validación el mismo conjunto de datos de entrenamiento. Podría pensarse que al utilizar el mismo conjunto de datos de entrenamiento para la validación se debería tener 100% de coincidencia, pero es fácil ver que no es así tomando como ejemplo el algoritmo ZeroR (descrito en el apéndice), el cual se limita a asignar a todas las observaciones la clase que aparezca en mayor número en el conjunto de entrenamiento; al realizar la validación aquellas observaciones que no tengan esa clase fallarán. El principal problema de esta técnica es que en el caso de que el modelo esté sobreajustado a los datos de entrenamiento el proceso de validación no es capaz de detectarlo.
- Utilizar la técnica denominada **hold-out** [078], donde el conjunto de datos de entrenamiento se divide en dos partes mutuamente excluyentes: una parte se utiliza para realizar el

entrenamiento propiamente dicho, y la otra se utiliza para realizar la validación. Usualmente se utiliza 2/3 de la muestra para el entrenamiento y 1/3 restante para la validación, aunque no hay una regla específica para esto. Esta técnica es sensible a cómo se realice la partición ya que las observaciones de entrada no necesariamente deben estar uniformemente distribuidas. Además, para su aplicación se requiere un conjunto de datos lo suficientemente grande como para que tanto el conjunto de entrenamiento y el de validación sean suficientemente representativos.

- Utilizar la técnica denominada **k-folds-validation** [078] (también llamada **leave-one-out** o **cross-validation**), la cual consiste en dividir el conjunto de entrenamiento en *k* subconjuntos de igual tamaño mutuamente excluyentes. Luego, se quita uno de esos *k* subconjuntos y se realiza el entrenamiento del modelo utilizando los *k-1* restantes combinados, utilizando el apartado para realizar la validación. Se repite el procedimiento quitando un conjunto a la vez y luego se agregan los resultados para obtener un resultado global (por ejemplo, promediando los resultados parciales). El problema de esta técnica es el tiempo requerido para ejecutar las *k* repeticiones del proceso.

3.3.1.1 - Evaluación de la calidad de los modelos de clasificación

Naturalmente, no todos los algoritmos de clasificación se comportan de la misma manera en todas las situaciones. Esto implica que habiendo muchas formas de clasificar un conjunto de observaciones, los resultados en unos casos serán mejores que en otros. Existen diferentes criterios para evaluar la calidad de los modelos de clasificación, entre ellos la exactitud, las medidas precision y recall, las curvas ROC, el valor kappa, y la matriz de confusión, que se describen a continuación.

En primer lugar, la **exactitud** es simplemente la cantidad de observaciones clasificadas correctamente respecto de la cantidad total de observaciones que componen el conjunto de validación. Es un valor numérico que permite la comparación fácil entre diferentes algoritmos de clasificación aplicados sobre el mismo conjunto de observaciones, aunque poco indica de la calidad misma del modelo. En otras palabras, solo permite seleccionar un algoritmo por sobre otro en cada caso puntual, pero que no permite generalizar.

En segundo lugar, el **valor estadístico kappa, κ,** es el indicador más usualmente utilizado para determinar la calidad de un modelo de clasificación. Básicamente, permite establecer qué tan buena resulta la clasificación realizada con el modelo construido si se la compara con el resultado que se obtendría si la clasificación se hiciera en forma aleatoria [079]. Matemáticamente se calcula como $\kappa=(TA-RA)/(1-RA)$, donde TA (*total accuracy*) indica la relación de observaciones correctamente clasificadas respecto de la cantidad total de observaciones del conjunto de validación, y RA (*random accuracy*) indica la probabilidad de que cada una de las clases sea asignada a cada observación si se realizara en forma aleatoria (en el caso de que todas las clases sean equiprobables entonces RA es equivalente a *1/#clase*s). Un valor de κ cercano a cero indica que el modelo obtenido no es mejor que la clasificación aleatoria, mientras que un valor cercano a 1 indica que el modelo de clasificación tiende a ser adecuado. Si bien el valor kappa puede ser considerado una medida apropiada para evaluar la calidad de un modelo no siempre queda claro el verdadero significado; por ejemplo, un valor 0.57 podría ser adecuado para algunas personas, pero no tanto para otras [079]. En [080] los autores proponen una clasificación por rangos de valores, la cual si bien ellos mismos aclaran que es arbitraria aún hoy es una referencia comúnmente citada [079]:

Valor de Kappa	Categoría
< 0.00	Pobre
0.0 – 0.20	Débil
0.21 – 0.40	Justa
0.41 – 0.60	Moderada
0.61 – 0.80	Sustancial
0.81 – 1.00	Casi perfecta

Tabla 2: Categorización del valor estadístico Kappa [079]

En tercer lugar, la **matriz de confusión** (*confussion matrix*) es un cuadro de doble entrada que permite observar la distribución de los errores cometidos por el clasificador aplicado al conjunto de datos de validación. Una matriz de confusión tiene la siguiente forma, donde el valor v_{ij}, indica la cantidad de observaciones a las cuales les correspondía la clase i y el modelo de clasificación les asignó la clase j:

	Clase 1	**Clase 2**	**…**	**Clase J**	**…**	**Clase N**
Clase 1	$\boldsymbol{v_{11}}$	v_{12}	…	v_{1J}	…	v_{1N}
Clase 2	v_{21}	$\boldsymbol{v_{22}}$	…	v_{2J}	…	v_{2N}
…	…	…	…	…	…	…
Clase I	v_{I1}	v_{I2}	…	$\boldsymbol{v_{IJ}}$	…	v_{IN}
…	…	…	…	…	…	…
Clase N	v_{N1}	v_{N2}	…	v_{NJ}	…	$\boldsymbol{v_{NN}}$

Tabla 3: Matriz de confusión de un modelo de clasificación

La suma de los valores en la diagonal principal corresponde a la cantidad total de observaciones correctamente clasificadas, mientras que la suma de los valores de todas las restantes entradas corresponde a la cantidad total de observaciones clasificadas en forma incorrecta [081].

Cuando el modelo de clasificación es binario, es decir solo existen dos clases, son frecuentes también el uso de las medidas denominadas *precision* y *recall*, y las curvas *ROC*.

Para definir precision y recall es necesario seleccionar previamente una de las clases objetivo como indicadora de un valor buscado; a dicha clase se le denomina S y a la otra clase se le denomina N. En base a dicha consideración se pueden definir cuatro métricas asociadas a la clasificación:

- Positivos correctos o TP (*true positives*), igual a la cantidad de observaciones que fueron clasificadas como S cuando realmente eran S,
- Positivos incorrectos o FP (*false positives*), igual a la cantidad de observaciones que fueron clasificadas como S cuando en realidad eran N,
- Negativos correctos o TN (*true negatives*), igual a la cantidad de observaciones que fueron clasificadas como N cuando realmente eran N,

- Negativos incorrectos o FN (*false negatives*), igual a la cantidad de observaciones que fueron clasificadas como N cuando en realidad eran S.

Habiendo seleccionado la clase indicadora y aplicado un modelo de clasificación a un conjunto de datos se definen precision y recall de la siguiente manera [082][083][084]:

- **Precision** (precisión): es la relación entre las observaciones clasificadas correctamente como S (true positives) respecto de todas las observaciones que fueron clasificadas como S (ya sea correctamente o incorrectamente). Matemáticamente, P=TP/(TP+FP).
- **Recall** (recuperación): es la relación entre las observaciones clasificadas correctamente como S (true positives) respecto de todas las observaciones que son realmente S (notar que éstas incluyen a las anteriores y también a las que siendo S fueron clasificadas como N, es decir a las false negatives). Matemáticamente, R=TP/(TP+FN).

Es importante tener en cuenta que la elección de la clase indicadora afecta a los valores precision y recall del modelo. Normalmente esta elección dependerá del contexto de aplicación del proceso de clasificación, y de igual forma dependerá del contexto el hecho de otorgarle mayor relevancia a una u otra medida [084]. Para eliminar este factor de incertidumbre se ha estandarizado una medida adicional que tiende a equilibrar la importancia de ambas medidas, denominada **medida-F** (*F-measure*) [083][084] definida como la media armónica de las medidas precision y recall: $F=(2\times R\times P)/(P+R)$. En cualquier caso los valores concretos de las diferentes medidas suelen no ser útiles por sí mismos sino que toman relevancia en la comparación con el valor de la misma medida en otro contexto, por ejemplo para comparar diferentes algoritmos de clasificación aplicados al mismo conjunto de datos.

También en el caso de la clasificación binaria es bastante difundido en análisis de la calidad de modelos de clasificación en base a las llamadas curvas ROC (*receiver operation characteristics*), originalmente ideadas para el análisis de señales captadas por radares y receptores de radio con el objetivo de identificar la presencia o no de determinadas amenazas [082][084]. El objetivo original de las curvas ROC era permitir ajustar los receptores a un nivel adecuado de receptividad de forma de maximizar los verdaderos positivos y/o minimizar los falsos negativos, La técnica fue tomada prestada por el área de la minería de datos para permitir ajustar los modelos de clasificación tratando de equilibrar la tasa de verdaderos positivos (TPR, *true positive rate*, coincidente por definición con la medida recall, es decir TPR=TP/(TP+FN)) con la tasa de falsos positivos (FPR, *false possitive rate*, definida como la cantidad de falsos positivos respecto de la cantidad total de negativos, es decir, FPR=FP/(TN+FN)) [084]. De esta manera, la curva ROC consiste de una gráfica bidimensional en la cual la abscisa representa la tasa de falsos positivos y la ordenada la tasa de verdaderos positivos (ambas en el rango [0, 1]). Idealmente, el mejor modelo (si se comparan varios modelos), o los mejores parámetros (si se comparan diferentes configuraciones del mismo modelo) se logra cuando el punto en la gráfica se encuentra más cercano a la esquina superior izquierda [085] (ningún falso positivo y ningún falso negativo).

3.3.2 – Agrupamiento (*clusterización* o *clustering*)

El **agrupamiento**, también conocido como clusterización (derivado de ***clustering*)**, es similar a la clasificación con la salvedad de que al comenzar se desconoce cuáles son las clases asignables, y en cambio éstas deben ser deducidas también a partir de los datos. La idea es dividir un conjunto de observaciones en grupos llamados **clústeres** (*clusters*) de tal manera que todos los miembros de un clúster sean similares entre sí y diferentes a los miembros de otros clústeres [059][064][065][086]. Si bien es deseable que los clústeres encontrados sean mutuamente excluyentes (no existan datos que sean miembros de más de un clúster a la vez) algunas propuestas admiten el solapamiento

debido a que asignan a cada clasificación una probabilidad o indicador de confianza (DoB, *degree of belief*) sobre la pertenencia de una observación a un clúster [066]. Es importante notar que el agrupamiento requiere de una noción de distancia, de forma tal que dadas dos observaciones cualesquiera se pueda calcular la distancia entre ellas; cuanto menor es la distancia entre dos observaciones más similares son entre sí. Es usual que para el mismo tipo de observaciones exista más de una función de distancia aplicable; según la función de distancia adoptada en cada caso el agrupamiento final puede variar. La agrupación es una técnica descriptiva puesto que su objetivo es tomar observaciones de una cierta realidad y agruparlas en un conjunto de clústeres desconocidos previamente; al mismo tiempo es una técnica no supervisada ya que no requiere de un entrenamiento previo para deducir los clústeres [066].

Entre los ejemplos más típicos de la agrupación se encuentra la selección de clientes en grupos de afinidad (de forma de armar promociones o planes específicos) y de los productos de una tienda en canastas (de forma de ofrecerlos juntos).

Los algoritmos de agrupamiento se suelen catalogar en dos tipos: los jerárquicos y los basados en particionamiento [062][076]. Los **algoritmos jerárquicos** a su vez se dividen en dos tipos: los llamados aglomerantes, o *bottom-up*, y los decisivos, o *top-down*. Los **algoritmos jerárquicos aglomerantes** comienzan por definir un clúster para cada observación y luego en forma iterativa van combinándolos en base a su "cercanía" hasta alcanzar alguna condición de parada [062][076]. Los **algoritmos jerárquicos decisivos** hacen lo opuesto, comenzando con un único clúster que contiene a todas las observaciones para luego ir desglosándolo hasta alcanzar una condición de parada [076]. Los **algoritmos basados en particionamiento**, por otra parte, tienen una primera fase en la que se estiman los clústeres (se define el conjunto de clústeres y se conforma de alguna manera la cantidad de clústeres definidos) y luego una segunda fase en la cual se asigna cada una de las observaciones a uno de ellos [062][076].

3.3.3 - Análisis asociativo

El **análisis asociativo**, o minería de reglas de asociación (ARM, *association rule mining*), tiene por objetivo encontrar correlaciones entre los atributos característicos de los datos [059][064]. Estas correlaciones se representan como reglas del tipo $A \rightarrow B$ donde la parte a la izquierda de la flecha se denomina antecedente y a la derecha consecuencia, y que se interpretan como "*cada vez que ocurre A probablemente también ocurra B*" [062][087]. Desde su introducción en 1993 por R. Agrawal, T. Imielinski y A. Swami ([089][088]) la tarea de descubrir reglas de asociación ha acaparado mucha atención, siendo en la actualidad uno de los métodos más comunes en KDD [087]. Si bien surgió con el objetivo de analizar los hábitos de compra de las personas para determinar cuáles productos suelen comprarse juntos [088] de forma de situarlos estratégicamente en las góndolas y planificar las promociones (de ahí que el análisis asociativo también sea conocido como **análisis de la cesta de las compras**) ha comprobado ser una técnica muy útil en múltiples áreas, como la detección de actividades fraudulentas [062][087] y el pronóstico del tiempo.

No se debe confundir las asociaciones entre atributos, donde el valor de un atributo está relacionado de alguna manera con el valor de otro, con las relaciones causales (o de dependencia) entre atributos, donde uno afecta explícitamente al otro. Que exista una asociación entre dos atributos no implica que deba existir una relación causal entre ellos [059] (por ejemplo, que las personas usualmente compren pan y leche al mismo tiempo no implica que necesariamente la compra de uno conduce a la compra del otro).

No es poco usual que luego de aplicar un método de análisis asociativo se termine teniendo decenas, cientos o incluso miles de reglas de asociación, la mayoría de las cuales no aportan mucha

información debido a que se cumplen en tan pocos casos que es poco probable que se vuelvan a utilizar o a que se cumplen en tantos casos que seguramente son reglas generales ya conocidas (triviales). Para poder seleccionar cuáles de todas reglas de asociación encontradas pueden ser realmente útiles se definen usualmente dos métricas de confiabilidad, llamadas respaldo (o soporte) y confianza (o confiabilidad). El **respaldo** (*support*) de una regla es la relación entre la cantidad de observaciones que satisfacen dicha regla respecto de la cantidad total de observaciones, mientras que la **confianza** (*confidence*) es la relación entre la cantidad de observaciones que cumplen la regla respecto de la cantidad de observaciones que solo cumplen con el antecedente (es decir, el complemento de la cantidad de observaciones en los cuales se cumple el antecedente de la regla pero no el consecuente, y por tanto la regla falla). Tanto el respaldo como la confianza son valores que están en el rango real [0,1]. El respaldo provee un indicador sobre la significancia de la regla, mientras que la confianza es un indicador sobre su fortaleza [089]. El problema de descubrir reglas de asociación "interesantes" en un conjunto de datos consiste en encontrar todas las reglas de asociación que superan un umbral mínimo de respaldo y confianza preestablecido [015][077][087] [089].

3.4 - Limitaciones de la minería de datos

Es importante tener en cuenta que la minería de datos no puede responder a todas las preguntas [058] ya que está limitado a los datos disponibles, los cuales no necesariamente son representativos del grupo de individuos que se estudia, pueden estar incompletos, pueden ser incorrectos, pueden estar desactualizados, entre muchas otras deficiencias. En general, la calidad de los resultados obtenidos al aplicar minería de datos depende directamente de la calidad de los datos así como de la correcta elección de los atributos a considerar [090]. Adicionalmente, no se debe confiar excesivamente en la minería de datos debido a que es posible que se consideren como patrones reales casos que se dan por azar, especialmente cuando los datos con los que se trabaja no son completamente representativos de la realidad modelada [062]. Un principio bien conocido por los estadísticos, llamado **Principio de Bonferroni**, establece que si en un conjunto de datos completamente aleatorios se obtiene una cantidad de observaciones de un evento particular mayor a la cantidad de observaciones que se obtienen en datos reales, entonces las observaciones reales no son relevantes y pueden ser descartadas [091]. Este principio brinda un límite superior sobre la calidad de los resultados que produce un algoritmo o técnica.

3.5 - Técnicas de minería de datos

Como se describió anteriormente, los algoritmos de minería de datos tienen tres componentes fundamentales: un modelo de representación, un criterio de evaluación, y un mecanismo de búsqueda. El modelo de representación es usado para describir los patrones presentes en los datos. El criterio de evaluación está conformado por un conjunto de funciones de ajuste que indican qué tan bien un modelo y sus parámetros son útiles para las metas trazadas en el proceso de minería de datos. El mecanismo de búsqueda consiste en seleccionar, o definir, una familia de modelos de representación (parametrizado) y luego resolver un problema de optimización consistente en determinar los valores de los parámetros más apropiados para los datos disponibles [065].

La elección de la técnica de minería de datos más apropiada depende fuertemente de los datos disponibles y del uso que se pretenda dar al conocimiento encontrado: ¿se busca predecir o describir? ¿es más importante la comprensibilidad o la confiabilidad? [060]. De todas maneras, salvo ciertos problemas específicos y de alta dificultad, la mayoría de problemas abordados por la

minería de datos dan resultados comparables, cualquiera que sea la técnica utilizada.

A continuación se describen algunas de las técnicas más comunes utilizadas en minería de datos.

3.5.1 - Árboles de decisión

Los **árboles de decisión** son principalmente una técnica predictiva, siendo ampliamente utilizados para tareas de clasificación y regresión, aunque también pueden ser usados para tareas descriptivas [065][092]. Son una de las técnicas más antiguas ya que han sido utilizadas desde hace mucho tiempo para múltiples aplicaciones en la industria y el comercio, incluso antes de que se emplease la expresión “minería de datos”.

Un árbol de decisión está conformado por tres componentes: un **nodo raíz** (*root node*) que constituye el punto de partida para construir y recorrer el árbol; un conjunto de nodos intermedios llamados **nodos de decisión** (*test nodes*) que presentan condiciones que conducen a tomar un camino u otro; y **nodos de hoja** (*leaf nodes*) que se corresponden con las clases objetivo [059]. Las secuencias de nodos que van desde la raíz hasta cada uno de los nodos hoja se denominan **ramas** del árbol. Normalmente los árboles de decisión son binarios, lo que significa que en cada nodo de decisión solo hay dos opciones, aunque no necesariamente deben serlo. Las observaciones son clasificadas navegando el árbol desde la raíz hacia las hojas, de acuerdo a los resultados de las pruebas realizadas en cada nodo de decisión [093]. La Figura 3 muestra un árbol de decisión que permite determinar el resultado de aplicar la operación XOR a dos entradas binarias *x* e *y*.

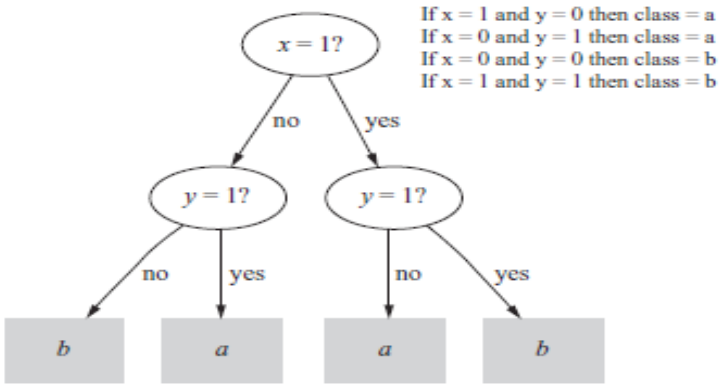

Figura 3: Árbol de decisión para calcular x XOR y, tomada de [094]

Se llaman **inductores** a los algoritmos que son capaces de construir un árbol de decisión a partir de un conjunto de datos de entrenamiento [093], con el objetivo de que sea mínimo (en profundidad y/o en anchura) y óptimo (que minimice la posibilidad de error en la clasificación). Construir el árbol mínimo y óptimo es posible en muy pocos casos, por lo que usualmente se utilizan técnicas heurísticas basadas en hacer crecer el árbol y podarlo para lograr una aproximación razonable [095]. Casi todos los algoritmos se fundamentan en la metodología de dividir y conquistar (*divide and conquer*) en forma iterativa: en cada iteración dividen la muestra de entrenamiento según algún criterio generando cada división un nodo de decisión, para luego repetir el procedimiento en cada porción resultante hasta alcanzar un determinado criterio de parada, como por ejemplo que todas las instancias de una rama pertenezcan a la misma clase (no hay división posible) o que el árbol alcance una profundidad máxima estipulada [093]. En cada iteración se busca encontrar la división que sea más selectiva [064] (es decir, que permita separar las observaciones lo más uniformemente posible). La elección de los criterios de parada no es sencilla ya que uno demasiado estricto puede crear árboles pequeños pero mal ajustados (hay grandes posibilidades de que la clasificación sea

incorrecta), mientras que uno demasiado permisivo puede crear árboles muy grandes y sobreajustados a los datos de entrenamiento (clasifican con exactitud a los datos de entrenamiento pero fallan en otros casos) [093].

Los árboles de decisión tienen ventajas y desventajas cuando se los compara con otras técnicas [064][092][093]. Entre las ventajas se destacan el hecho de que son fáciles de comprender incluso por personas no entrenadas, pueden ser utilizados con atributos nominales o numéricos (pueden ser utilizados también para regresión), y son robustos ante valores faltantes y datos incompletos. Por otro lado, la mayoría de los algoritmos de inducción sólo funcionan con clases discretas, su rendimiento decrece cuando son muchos los atributos a considerar, pueden ser demasiado sensibles a los datos de entrenamiento, a atributos irrelevantes y al ruido en los datos, y las particiones se realizan sobre clasificadores individuales sin considerar combinaciones de ellos por lo que no son es capaces de capturar relaciones entre atributos.

3.5.2 - Redes bayesianas

Las **redes bayesianas**, también llamadas redes de credibilidad (*belief networks*) o **clasificadores bayesianos**, son una de las técnicas de clasificación más ampliamente utilizadas debido a su relativa simplicidad de uso (aunque no así su construcción) [064]. Pertenecen a una familia de modelos llamados probabilistas basados en grafos en los cuales se utilizan grafos acíclicos dirigidos (DAGs, *directed acyclic graphs*) para representar el conocimiento que se tiene sobre un dominio [096]. En particular cada nodo del grafo representa una variable aleatoria mientras que las aristas representan las dependencias probabilistas entre ellas. En adición al DAG, que es la parte cualitativa del modelo, las redes bayesianas poseen también una parte cuantitativa compuesta por los parámetros del modelo, los cuales describen de alguna manera la distribución de probabilidad condicional (CPD, *conditional probability distribution*) de cada uno de los nodos, que depende solo de sus ancestros. Para el caso de variables aleatorias discretas las probabilidades condicionales se suelen representar como una tabla, llamada **tabla de probabilidades condicionales** (CPT, *conditional probability table*), en la cual se lista la probabilidad de que un nodo particular tome cada uno de sus posibles valores en base a los valores de sus padres [096], como se muestra en la Figura 4.

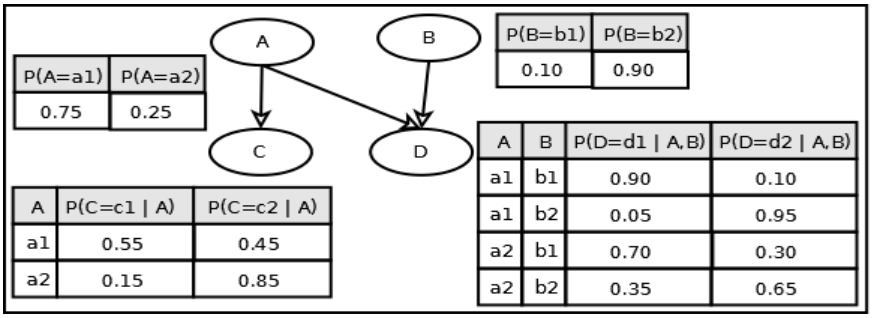

Figura 4: Ejemplo de una red bayesiana

Al comenzar, la red bayesiana es desconocida y debe ser construida a partir de los datos disponibles. Esto es conocido como **aprendizaje** y consiste en estimar la topología de la red y sus correspondientes parámetros (probabilidades condicionales) a partir de un conjunto de datos de entrenamiento. En general ocurre que estimar la topología es más difícil que estimar los parámetros y se hace más complejo aun cuando existen nodos para los cuales no hay observaciones disponibles (nodos ocultos) o los datos son insuficientes o incompletos (observabilidad parcial). De hecho, en la práctica solo para el caso en que la topología es conocida de antemano y hay observabilidad total existen técnicas aplicables; si la topología no es conocida o hay observabilidad parcial por lo

general se trata de problemas computacionalmente intratables [096].

Dada una red bayesiana, con su correspondiente topología y parámetros, pueden realizarse dos tipos de actividades: **inferencia predictiva** (*predictive support*), también llamado razonamiento descendente (*top-down*), consistente en determinar el valor de un nodo particular en base a los valores de sus padres, e **inferencia diagnóstica** (*diagnostic support*), también llamado razonamiento ascendente (*botton-up*), consistente en determinar el valor de un nodo basándose en los valores de sus hijos [096]. Esto significa que las redes bayesianas pueden ser usadas como método predictivo (clasificación) y como método descriptivo (clusterización). La atracción principal de las redes bayesianas reside en su simplicidad de interpretación, eficiencia computacional y rigurosidad matemática, aunque también presentan algunos importantes inconvenientes tales como la necesidad de un conjunto de datos de entrenamiento muy grande, y el hecho de que si una categoría de predicción (clase) no está presente en los datos de entrenamiento la técnica asume que dicha categoría tiene probabilidad cero [064].

3.5.3 – Redes neuronales

Las **redes neuronales artificiales** (ARN, *artificial neural networks*), o solo redes neuronales, son una abstracción del sistema nervioso de los animales [097][098] propuesta inicialmente en 1943 por Warren McCulloch y Walter Pitts ([099]). Son mecanismos numéricos compuestos por tres tipos de capas: una **capa de entrada**, que acepta datos, una o más capas de procesamiento, llamadas **capas ocultas**, y una **capa de salida**, que entrega los resultados. Cada capa puede tener cualquier número de nodos de procesamiento, llamados **neuronas**. Se dice que una red neuronal está **completamente conectada** si todas las neuronas de una capa se conectan con todas las neuronas de la capa siguiente [097][098] (ver la Figura 6 en la página 40). La cantidad de capas, la cantidad de neuronas en cada una de ellas y las conexiones efectivas entre neuronas constituyen la **topología de la red neuronal**.

Una red neuronal queda caracterizada por su topología y por dos funciones concretas llamadas función base y función de activación. La **función base**, también llamada **función de red**, es una función que toma como entrada todos los datos proporcionados a cada neurona y los combina de alguna manera para producir un único valor (el tipo de combinación realizada determina el tipo de neurona). Esta función base suele tener un conjunto de parámetros que se establecen inicialmente en forma manual por un experto, o incluso aleatorio, para luego ir ajustándolos a medida que la red neuronal procesa los datos de entrada. La **función de activación** toma como argumento el resultado de la función base y produce una salida que es considerada el resultado de procesamiento de la neurona. Si bien tanto la función base como la de activación son propias de la red (son las mismas para todas las neuronas) los parámetros de la función base varían para cada par de neuronas, de forma que dadas dos neuronas de capas consecutivas si hay flujo entre ellas entonces existe un conjunto de valores de los parámetros específicos de ellas.

Existen diferentes tipos de neuronas, los cuales se caracterizan principalmente por su función base. El tipo original de neurona, llamado **perceptrón**, fue desarrollado entre 1950 y 1960 por Frank Rosenblatt inspirado por el trabajo de McCulloch y Pitts. Hoy en día se utilizan otros tipos de neuronas más complejas, como las neuronas **sigmoideas**. Un perceptrón es un tipo de neurona que toma varias entradas $x=\{x_1,x_2,..,x_n\}$ binarias y produce una única salida también binaria como resultado de la combinación lineal de las entradas con un **vector de pesos** (*weight vector*) $w=\{w_1,w_2,...,w_n\}$ [100]; si el resultado de la combinación lineal es menor o igual a un cierto umbral entonces la salida de la neurona es 0, mientras que en caso contrario es 1. En ocasiones, en lugar de utilizar un umbral se emplea un valor llamado **sesgo** (*bias*) cuyo valor es el opuesto del umbral y es

representado con la letra *b* (*b*=0-*umbral*). De esta manera, la salida de la neurona se puede representar como se muestra en la fórmula 1 [100]. Tanto el vector de pesos como el umbral son establecidos inicialmente por el analista para luego ser ajustados automáticamente a medida que se procesan los datos de entrenamiento. El valor del sesgo, que no está restringido a ningún rango especial pudiendo ser tan grande o chico como se desee o incluso negativo, puede ser interpretado como la facilidad que tiene la neurona para producir una salida igual a 1 [100].

$$salida: \begin{cases} 0 \; si \; w \cdot x + b \leq 0 \\ 1 \; si \; w \cdot x + b > 0 \end{cases}$$

Fórmula 1: Salida de un perceptrón

La cantidad de neuronas de la capa de entrada está determinado por el número de atributos de los datos (una neurona para cada atributo), mientras que la cantidad de neuronas de la capa de salida está determinado por la cantidad de clases posibles (si hay N clases entonces puede utilizarse N o $\lceil \log_2(N) \rceil$ neuronas según se desee). El número de neuronas ocultas determina la capacidad de aprendizaje de la red neuronal; para que el comportamiento de la red neuronal sea correcto es importante determinar apropiadamente el número de neuronas de las capas ocultas, que debe ser la cantidad mínima con las cuales la red rinda de forma adecuada, lo cual se consigue evaluando el rendimiento de diferentes arquitecturas. En cualquier caso, para la mayoría de problemas prácticos basta con utilizar una sola capa oculta [097].

Si bien una red neuronal es útil para procesar un conjunto de datos de entrada difícilmente pueda ser construida manualmente, es decir, que los analistas determinen los mejores valores para el vector de pesos y el sesgo. En su lugar se han desarrollado algoritmos de aprendizaje automático que permiten determinar dichos valores a partir de un conjunto de datos de entrenamiento. El supuesto sobre el cual se basan estos algoritmos es que un pequeño cambio en los valores de los pesos o del sesgo produce un pequeño cambio en la salida ($x[w+\partial w]+[b+\partial b] \rightarrow [salida+\partial salida]$). Lamentablemente dicha suposición no se cumple cuando las neuronas son de tipo perceptrón porque su salida solo puede ser 0 o 1: o bien las modificaciones no producen alteraciones en la salida, o bien producen un cambio radical. Por esto se dice que los perceptrones son “duros para aprender” [100]. Para solventar este problema se han desarrollado otros tipos de neuronas, entre los cuales se destaca la **neurona sigmoidea**. Este nuevo tipo de neurona admite entradas que abarquen todo el rango [0,1], y en lugar de producir la salida mediante una combinación lineal de ellas utiliza la función sigmoidea ($\sigma(x)=1/[1+e^{-x}]$). Dado que el símbolo σ es también llamado función logística a las neuronas sigmoideas también se le llama a veces **neuronas logísticas**.

$$salida = \frac{1}{1 + e^{-\sum_j w_j x_j - b}}$$

Fórmula 2: salida de una neurona sigmoidea o logística

De esta manera la salida de una neurona sigmoidea se calcula como se muestra en la fórmula 2. En la Figura 5, tomada de [100], se puede observar que la función sigmoidea es una versión “más suave” de la función escalera, propia de los perceptrones, y es justamente esta suavidad la que permite aplicar la suposición de que pequeños cambios en el vector de pesos o en el sesgo producen pequeños cambios en la salida de la neurona. Así, mientras las neuronas sigmoideas presentan casi

el mismo comportamiento cualitativo que los perceptrones hacen más fácil determinar cómo los cambios en el vector de pesos y en el sesgo afectan a la salida [100].

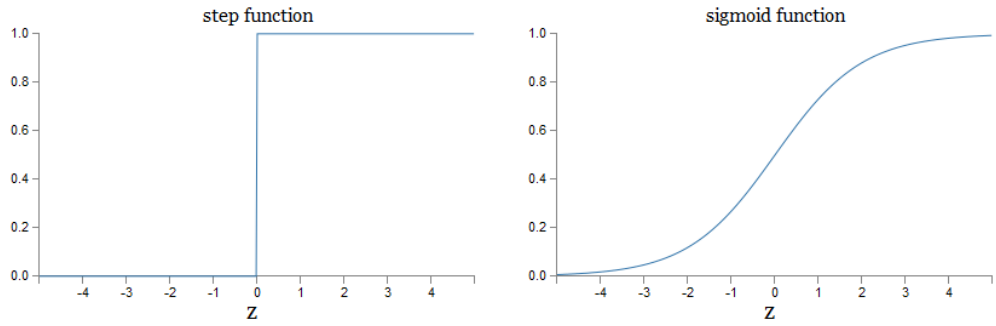

Figura 5: Comparación gráfica entre las funciones escalera y sigmoidea, tomada de[90]

De esta manera, la red neuronal comienza casi completamente ignorante, con los valores de los parámetros de la función base y el sesgo fijados de forma relativamente arbitraria (incluso aleatoria), y va ajustando dichos valores a partir de las observaciones hasta conformar un modelo de comportamiento de la realidad [059][101]. El entrenamiento consiste en descubrir las ponderaciones y sesgo que ajustan mejor con los datos existentes bajo la premisa de que las entradas similares deben producir salidas similares. Existen muchas formas de hacerlo pero la técnica más usual es la denominada **propagación inversa** (*back-propagation*) que consiste en generar un conjunto de valores de parámetros y sesgo, computar la salida de la red neuronal para una muestra dada, determinar el error total, compararlo con el anterior y ajustar el vector de pesos y el sesgo hasta alcanzar una condición de parada preestablecida [097][101][102]. La Figura 6 muestra un ejemplo de una red neuronal con una arquitectura 3-4-2 completamente conectada. Las cuatro neuronas de la capa oculta tienen un valor de sesgo y todas las conexiones tienen un parámetro (solo se muestran cuatro de los veinte existentes).

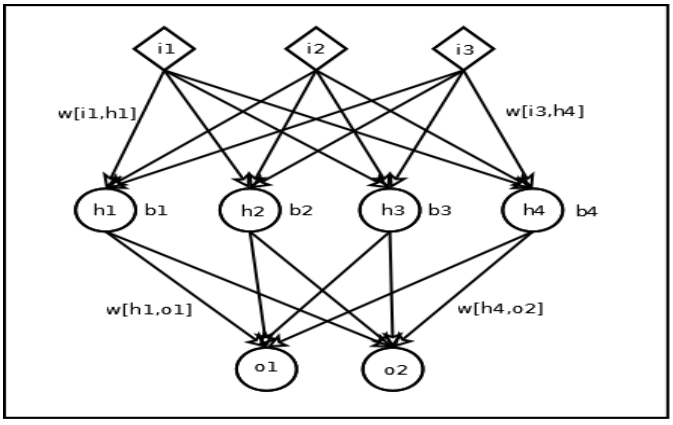

Figura 6: Red neuronal 3-4-2 completamente conectada

Si bien las redes neuronales son muy populares entre la comunidad del aprendizaje automático son menos apropiadas para la minería de datos debido a que funcionan como una caja negra y no explican cómo llegaron al modelo presentado [060]. Además, aunque tienen una alta tolerancia al ruido en los datos y poseen la habilidad de capturar relaciones muy complicadas entre los clasificadores y la clase, para su correcto funcionamiento deben disponer de datos de entrenamiento suficientes y requieren más tiempo para su ejecución que otras técnicas [064].

3.5.4 - Algoritmos genéticos

Los **algoritmos genéticos** son una técnica supervisada que se suele utilizar en problemas para los cuales no hay un método general conocido, o cuya solución es demasiado compleja para ser computada [059]. Fueron propuestos originalmente en la década de 1960 por John Holland y desarrollados en las décadas siguientes por el propio Holland y sus estudiantes [103]. Se caracterizan por tener un enfoque de fuerza bruta (realizar muchas pruebas con poca o nula planificación hasta encontrar una solución), lo que requiere de un gran poder de procesamiento. En general tienden a producir soluciones no muy lejanas de la óptima, aunque muy lentamente en comparación con otros métodos [059].

Los algoritmos genéticos parten de un conjunto de posibles soluciones remotamente aceptables (en ocasiones creado en forma completamente aleatoria) para luego combinarlas entre sí de forma iterativa conformando soluciones híbridas que pueden acercarse a la solución óptima (aunque también podrían alejarse). Para determinar qué tan bien una solución híbrida se aproxima a la solución real se la aplica al conjunto de datos de entrenamiento y se determina cuántos de los datos son clasificados correctamente. La idea fundamental es favorecer a las soluciones más fuertes en detrimento de las más débiles, de manera similar al proceso de selección natural propuesto por Charles Darwin (supervivencia del más apto) [059][067][103]. El conjunto de soluciones candidatas es llamado **población**, y cada una de las soluciones que lo componen es llamada **individuo**. Los individuos se representan como cadenas de símbolos usualmente binarios llamados **cromosomas**, de forma tal que todos los individuos tienen el mismo número de cromosomas [059][076][103]. Por ejemplo, si se tienen dos variables binarias, *a* y *b*, y una clase también binaria, *c*, entonces la regla "*if a and not b then c"* podría representarse como *101* ("*si a=1 y b=0 entonces c=1"*); si los atributos o las clases no son binarios pueden usarse más cromosomas para representar a cada uno [076]. Bajo esta representación todos los cromosomas que corresponden a las variables independientes conforman el **antecedente** del individuo mientras que los que corresponden a la clase conforman el **consecuente**; en el ejemplo, el antecedente es 10 y el consecuente es 1. En cada iteración se construye una nueva **generación** de individuos, y se dice que la población **evoluciona** hacia su objetivo. La evolución puede ser por **cruzamiento**, donde un par de individuos se intercambian los antecedentes y consecuentes o por **mutación**, donde algunos cromosomas de un individuo se alteran al azar [076][103]. Algunos autores llaman cruzamiento a cualquier combinación que se haga de los individuos, sin tener en cuenta si forman parte del antecedente o del consecuente [067].

Existen dos enfoques comunes para la representación de los individuos, llamadas Michigan y Pittsburgh. Cada enfoque presenta sus ventajas y desventajas para la aplicación en los algoritmos. El enfoque Pittsburgh produce individuos sintácticamente más largos (requiere más símbolos que Michigan para representar a cada individuo), lo que suele hacer las computaciones más complicadas y lentas. Por su parte el enfoque Michigan produce individuos más cortos, reduciendo el tiempo y el esfuerzo de procesamiento, aunque solo permite evaluar la adecuación de cada individuo en forma aislada, no permitiendo evaluar el conjunto de individuos en su globalidad [067].

3.5.5 - Sistemas difusos

Los llamados **sistemas difusos** (*fuzzy systems*) constituyen una técnica basada en la **teoría de los conjuntos difusos** (*fuzzy sets theory*, FST), también llamada teoría de posibilidades (*possibility theory*), propuesta en 1965 por Lotfi A. Zadeh como evolución de la teoría clásica de conjuntos, que permite trabajar con datos vagos o inexactos que no pueden encuadrarse en categorías con límites estrictos [076][104]. Por ejemplo, cuando se trata de definir el conjunto de las personas "altas",

surgen dudas como ¿qué se considera una persona alta? o ¿a partir de qué umbral se considera que una persona es alta? [105]. Incluso pueden considerarse factores externos al problema, como la procedencia de las personas (las personas nórdicas son más altas que las asiáticas) y el contexto de evaluación (lo que puede considerarse alto para jugar al básquetbol es muy diferente para ser corredor de caballos).

Bajo la teoría clásica de conjuntos, dado un elemento y varios conjuntos, el elemento o bien pertenece a cada conjunto o no pertenece, no habiendo una tercera opción (principio del tercero excluido), aunque podría pertenecer a más de un conjunto a la vez (intersección de conjuntos). En 1964 Zadeh propuso una nueva teoría, que llamó teoría de conjuntos difusos, según la cual el elemento pertenece a todos los conjuntos a la vez, pero con diferentes grados de pertenencia en cada uno [068][076][086][105][106]. Así, para un elemento específico existe un valor asociado para cada conjunto, entre 0 y 1, que indica su **grado de pertenencia** a dicho conjunto. El grado de pertenencia de un elemento a un conjunto está dado por una función, llamada función característica del conjunto difuso, o **función de pertenencia**, que es propia del conjunto [086][105][106]. El artículo [106] brinda una explicación psicológica de la teoría de conjuntos difusos que resulta interesante para comprender porqué la teoría es apropiada para extraer conocimiento en base a las similitudes entre la teoría y la forma de razonamiento humano.

A partir de la teoría de conjuntos difusos se han construido múltiples técnicas y algoritmos que en el área de la minería de datos son utilizados principalmente para agrupación, aunque también pueden ser usados para clasificación, descubrimiento de reglas de asociación, y otras tareas usuales. Por ejemplo, la mayoría de los algoritmos convencionales de agrupamiento producen modelos en los cuales todas y cada una de las observaciones son asignadas a un y solo un clúster, clústeres que tienen fronteras estrictamente determinadas. Sin embargo, en el mundo real esas fronteras no son tan naturales y hasta pueden ser poco intuitivas. La teoría de conjuntos difusos es aplicable en este caso en la llamada agrupación difusa (*fuzzy clustering*) que ha demostrado ser útil en muchas áreas, por ejemplo en la bioinformática [107].

Además de su aplicación en minería de datos, las herramientas construidas en base a la teoría de los conjuntos difusos tienen el potencial de soportar todas las etapas de KDD, y en particular puede ser utilizada en la selección y preparación de los datos, modelando datos vagos como conjuntos difusos, o condensando varias observaciones borrosas como un único conjunto difuso [107]. Aunque los algoritmos genéticos y las redes neuronales son tan buenos como los sistemas difusos, en la mayoría de los casos estos últimos tienen la ventaja de que las soluciones encontradas pueden representarse en términos que los operadores humanos pueden comprender mejor [068].

3.5.6 – Sistemas aproximados

Los **sistemas aproximados** están basados en la teoría de conjuntos aproximados (RST, *rough set theory*) introducida por Zdzislaw Pawlack en 1982 y que se considera uno de los primeros enfoques no estadísticos para el análisis de datos [015][076][108][109][110]. La **teoría de conjuntos aproximados** se basa en el concepto de clases de equivalencia, bajo la asunción de que los datos que forman parte de una misma clase de equivalencia son indistinguibles entre sí, por lo que para lograr una correcta clasificación de las observaciones basta con encontrar las relaciones de equivalencia que las definen, llamadas **relaciones de indiscernibilidad**. Parte con el mismo objetivo que la teoría de conjuntos difusos de permitir modelar el conocimiento vago o impreciso, solo que en lugar de hacerlo mediante la membresía ponderada (donde cada elemento tiene una cierta probabilidad de pertenecer a una clase) lo hace mediante aproximaciones a los conjuntos a los que puede pertenecer, haciendo borrosos los límites de esos conjuntos de clasificación (clases) para

dar lugar a que cada observación pueda pertenecer a más de una a la vez [108]. Para lograrlo, cada una de las clases se define en base a una **aproximación inferior**, compuesta por todas las observaciones que es seguro que forman parte de ella, y a una **aproximación superior**, compuesta por todas las observaciones que no se puede afirmar que no forman parte de ella. La diferencia entre ambas aproximaciones se llama **región frontera** y está compuesta por las observaciones que no puede asegurarse que formen parte del conjunto. Si para algún conjunto ocurre que las aproximaciones inferior y superior son iguales se dice que el conjunto es **definible**, mientras que en otro caso se dice que el conjunto es **indefinible**.

Los sistemas aproximados representan el conocimiento o la información disponible como una tabla, llamada tabla de decisión, en la cual las filas corresponden a las observaciones y las columnas a los atributos que las describen (como en la Figura 7). Uno de los atributos es llamado decisión o clase. Cada fila de la tabla determina una regla de decisión (que puede repetirse), ya que en base a los valores de cada atributo se puede determinar el valor de la clase. En ocasiones puede ocurrir que para dos filas diferentes todos los atributos correspondientes tengan el mismo valor pero que la decisión sea diferente, en cuyo caso se dice que las reglas son inconsistentes. El factor de consistencia de una tabla de decisión dados un conjunto de atributos y una decisión se calcula como la cantidad de reglas consistentes respecto de la cantidad total de reglas y es un indicador de qué tantas reglas inconsistentes hay en la muestra [108].

U	a_1	a_2	a_3	d
x_1	2	1	3	1
x_2	3	2	1	2
x_3	2	1	3	1
x_4	2	2	3	2
x_5	1	1	4	3
x_6	1	1	2	3
x_7	3	2	1	2
x_8	1	1	4	3
x_9	2	1	3	1
x_{10}	3	2	1	2

Figura 7: Ejemplo de tabla de decisión, tomada de [110]

En la teoría de conjuntos aproximados es importante el concepto de dependencia entre atributos: un subconjunto de los atributos es dependiente de otro subconjunto de atributos si los valores de los atributos del primero pueden determinarse en base a los valores del segundo. La dependencia entre atributos es importante porque permite eliminar algunos atributos, llamados superfluos, y aún así seguir teniendo el mismo sistema sin pérdida de información; los conjuntos de atributos minimales (a los que ya no se puede quitar más atributos sin perder información) se denominan **reducciones** (*reducts*), y los atributos que están presentes en todos los reductos constituyen el **núcleo** (*kernel*) del sistema. Notar que los atributos que no forman parte de ningún reducto son de por sí superfluos y pueden obviarse definitivamente. Puede suceder que para el mismo antecedente se tengan dos consecuentes diferentes en cuyo caso se dice que la regla es **no determinista**; si esto no sucede, se dice que la regla es **determinista**.

El artículo [110] describe el proceso de clasificación con mayor detalle, paso a paso y con ejemplos de aplicación concretos; en [108] se muestra la aplicación de la teoría al caso real de la detección de pacientes con dengue.

Una vez entrenado el sistema de clasificación por conjuntos aproximados puede utilizarse para clasificar nuevas observaciones, donde para cada una de ellas puede presentarse cuatro casos [110]: si la observación coincide con una de las reglas deterministas entonces la clasificación es directa; si la observación coincide con una regla no determinista se presenta al analista las opciones y éste es responsable de tomar una decisión (puede ser simplemente elegir la de mayor soporte); si la observación coincide con más de una regla (la diferencia con el caso anterior radica en que ahora los antecedentes no son iguales) pero todas ellas llevan a la misma decisión entonces la clasificación también es directa, sino se procede como en el caso anterior; y si la observación no coincide con ninguna de las reglas posiblemente la mejor opción sea ofrecer al analista un listado de las reglas más parecidas bajo alguna cierta función de distancia en base a los atributos participantes.

Los sistemas de clasificación basados en la teoría de conjuntos aproximados solo son aplicables con atributos discretos (cualitativos), por lo que en el caso de atributos continuos (cuantitativos) es necesario realizar una discretización de ellos [110].

3.5.7 – Boosting adaptativo

El **boosting adaptativo** (*adaptive boosting*) es una técnica que en lugar de intentar determinar un conjunto reducido de reglas de predicción complejas pretende generar una colección grande de reglas generales rudimentarias ponderadas que pueden ser combinadas de alguna manera para obtener un modelo más complejo [075][111]. La idea original, presentada en 1989 por Robert Shapire, parte de la proposición de que un algoritmo "débil", apenas mejor que una simulación aleatoria, puede ser mejorado (*boost*) hasta convertirse en un algoritmo más fuerte [111]. Existen muchas variantes de algoritmos de *boosting* adaptativo así como existen muchas herramientas independientes que los implementan. El *boosting* adaptativo puede ser usado para clasificaciones binarias (donde la clase solo puede tomar uno de dos dos valores posibles) o multinomiales (donde la clase puede tomar uno de múltiples valores diferentes). En el caso de la clasificación binaria el *boosting* adaptativo casi siempre usa un esquema {-1,1} porque esa codificación simplifica la implementación de los algoritmos respecto del esquema clásico *{0,1}* utilizado en la mayoría de las otras técnicas de clasificación.

Las reglas generales construidas mediante los algoritmos de *boosting* adaptativo se denominan **clasificadores débiles** y son de la forma *"si x entonces y"*. A cada uno de dichos clasificadores débiles se le asigna un factor de ponderación, denotado con el símbolo α (de ahí que se le llame factor alfa o valor alfa). El conjunto de clasificadores débiles y sus respectivos valores alfa conforman el modelo de clasificación. Una vez construido un modelo de clasificación, para clasificar un nuevo elemento se consideran todas las reglas que aplican y se calcula la suma ponderada del valor de clasificación de dicha regla con el valor alfa [075]. Determinar el conjunto de clasificadores débiles es bastante sencillo ya que basta con determinar todas las reglas de la forma *"si x entonces y"* con un solo antecedente a partir de las observaciones; luego la determinación de los valores α es la parte clave de los algoritmos de *boosting* adaptativo.

3.5.8 – Máquinas de soporte vectorial

Las **máquinas de soporte vectorial** (*support vector machines*) más conocidas por su abreviatura en inglés, SVM, son principalmente una técnica de clasificación binaria (donde solo hay dos clases posibles) aunque también puede usarse para análisis regresivo e incluso existen propuestas para permitir la clasificación con más de dos clases. La primera referencia a las SVM pueden encontrarse en el trabajo de Vladimir Vapnik y sus colegas en 1979, aunque se suele considerar que el artículo principal, también correspondiente a Vapnik, es de 2005 [112]. Entre las principales ventajas de las

SVM se encuentran el hecho de que se adaptan a conjuntos de datos de entrenamiento pequeños y que usualmente tienen pocos parámetros ajustables [041], además de que su funcionamiento es demostrable matemáticamente.

En la teoría de las SVM los datos a clasificar son representados como vectores de números reales en un espacio *d*-dimensional (donde *d* es la cantidad de atributos de cada vector); así cada dato representa un punto en dicho espacio. El objetivo de las SVM es determinar un **hiperplano**[1] dentro de ese espacio que permita separar linealmente a los puntos de la muestra en dos clases. En el caso de que exista más de un hiperplano (por lo general así ocurre) se elige el que resulte equidistante de las dos clases (considerando los puntos más cercanos de cada clase al hiperplano), siendo llamado **hiperplano óptimo** [113][114].

Dado un conjunto de datos de entrenamiento, si los datos fuesen linealmente separables entonces existiría un hiperplano como el buscado. Sin embargo muy rara vez sucede que los datos puedan ser divididos linealmente por lo que usualmente no es posible encontrar ningún hiperplano que permita separarlos fácilmente. Para solventar esta dificultad las SVM introducen el concepto de **espacio transformado inducido por un núcleo** (*kernel induced feature space*) que consiste en transformar los datos trasladándolos a un espacio de mayor dimensión donde sí sean separables linealmente (Figura 8[2]). Se denomina **espacio de entrada** al inicial y **espacio transformado** o espacio caracterizado al final [112][113][115]. Para realizar esta transformación se debe definir una función que permita transformar cada vector de entrada [112] a su correspondiente vector en el espacio transformado. En la práctica se observa que cuando la dimensión del plano caracterizado es mucho mayor a la del plano de entrada entonces es probable que los datos puedan separarse linealmente; solo hay que tener cuidado de no embarcarse en un problema que requiera demasiada computación y en no sobreajustarse a la muestra [112].

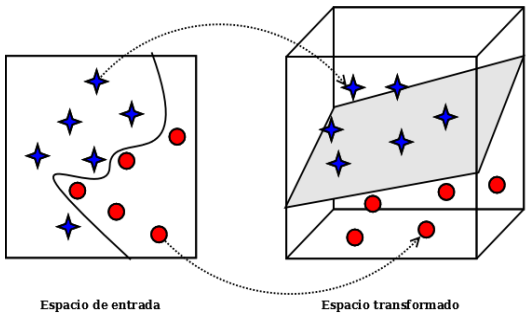

Figura 8: Transformación de vectores (2 dimensional a 3 dimensional) [2]

Durante el desarrollo de la teoría de SVM los investigadores notaron que siempre que se aplica la función de transformación a un vector es en el marco de un producto interno con algún otro vector, por lo que siempre será necesario poder aplicar dicha operación en el vector transformado; sin embargo, dependiendo de la dimensión del espacio transformado el cálculo del producto interno en dicho espacio podría ser demasiado costoso o incluso imposible si se desconoce su expresión. Para lidiar con este problemas las SVM recurren al teorema de Mercer, la cual se puede resumir en

1 Dado un espacio d-dimensional, un hiperplano, o plano de decisión, es una construcción de dimensión d-1 que permite dividir linealmente al espacio en dos partes. Que la división sea lineal implica que es hecha mediante una figura de forma no-irregular (por ejemplo, en un espacio bidimensional una división lineal está dada por una línea recta mientras que en un espacio tridimensional está dada por un plano). Para todo espacio siempre existen infinitos hiperplanos.

2 Esta imagen es una adaptación de la que puede verse en https://tr.m.wikipedia.org/wiki/Dosya:Kernel_yontemi_ile_veriyi_daha_fazla_dimensiyonlu_uzaya_tasima_islemi.png, pero de la cual no se tiene seguridad de la fuente ni de los derechos de autor.

que si una función de al menos dos variables es simétrica y definida positiva entonces dicha función es un producto interno en algún espacio (aunque no siempre se puede anticipar la dimensión del mismo). El teorema de Mercer permite aplicar lo que algunos autores denominan **truco del núcleo** (*kernel trick*) [113][114] que implica sustituir el producto interno complejo de calcular por una función especialmente elegida, que cumpla las condiciones mencionadas, llamada **función de núcleo** (*kernel function*), denotada $K(v_1,v_2)$, más sencilla de calcular. De esta manera no es necesario elegir de antemano la dimensionalidad del espacio (está dada por la función elegida) y no es necesario calcular ni la transformación de cada vector de un espacio a otro, ni el producto interno de los vectores transformados. Para obtener una función de núcleo existen dos caminos: tomar la función de mapeo del espacio de entrada al espacio caracterizado y luego derivar la expresión de la función (lo que requiere elegir dicho mapeo y luego determinar la transformación de los vectores y calcular el producto interno de todos ellos) o proponer una función (cualquiera) y verificar si es válida aplicando la condición de Mercer. Afortunadamente existen diferentes funciones de núcleo estándar que ya fueron probadas, entre las cuales se destacan la polinómica (donde un parámetro ajustable permite establecer la dimensionalidad del espacio transformado; cuanto mayor sea d mayor será la facilidad para separar los elementos pero también mayor será la necesidad de computación), RBF (*radial based function*), RBF gaussiano y el perceptrón multicapa [112][113] (en [113] se describen otras).

Un problema común a la hora de utilizar SVM es cómo elegir la función de núcleo más apropiada para el caso. Una opción sería probar con varias de ellas y luego elegir la que produzca mejores resultados, pero eso probablemente llevaría a un sobreajuste a los datos de entrenamiento. En general todo indica que no hay un mecanismo para elegir la mejor función y que usualmente se termina aplicando la experiencia: dado un problema específico debe buscarse en la literatura otros problemas similares para tratar de identificar semejanzas y así inclinarse por una [112].

3.6 - Minería de datos activa

La minería de datos activa (*active data mining*) es un paradigma bajo el cual el proceso de minería de datos se realiza periódicamente, y las reglas que se van aprendiendo en cada oportunidad se añaden a una base de datos conjuntamente con una serie de estadísticas tales como la confianza (*confidence*) y el respaldo (*support*) [116]. La idea detrás de esto es evitar aplicar un método de minería de datos a un conjunto enorme de datos cada vez (donde en cada ejecución se consideran todos los datos utilizados en la ejecución anterior más los datos nuevos) con los correspondientes requerimientos de esfuerzo y tiempo. Para hacerlo, los datos se dividen en ventanas de tiempo y el método de minería de datos elegido se aplica a cada una de las ventanas en forma incremental, habiendo siempre una ventana actual. Las reglas descubiertas en cada ejecución se almacenan en una base de datos de reglas, llamada ***rulebase***, en la cual los parámetros estadísticos (confianza y respaldo) de cada una de las reglas tiene asociado una secuencia de valores, llamado **historial** del parámetro de la regla. La base de datos de reglas puede entonces consultarse utilizando predicados que permitan seleccionar las reglas en base a la **forma** (*shape*) del historial de los parámetros; la forma del historial de un parámetro está dado por la secuencia de cambios (aumentos o disminuciones) de los valores. A medida que los datos van llegando para la ventana actual, el método de minería de datos se aplica a esos nuevos datos y la *rulebase* es actualizada, añadiendo las nuevas reglas encontradas o actualizando las estadísticas de las reglas conocidas. Una forma en el historial de una regla puede ser descrita considerando la secuencia de transiciones de valores desde el comienzo de un período hasta el final del mismo (por ejemplo, tres aumentos consecutivos, un aumento seguido de un descenso, etc.). Un conjunto estándar de formas propuesto en [117], llamado

formas básicas, es {*up*, *Up*, *down*, *Down*, *appears*, *disappears*, *stable*, y *zero*} [116], donde la forma "*up*" representa un incremento pequeño mientras que "*Up*" indica un incremento mayor; "*appears*" indica una transición de 0 a algo mayor que 0, mientras que "*disappears*" indica lo contrario; "*stable*" indica que no hay una variación apreciable. Además, pueden crearse formas complejas combinando las formas básicas usando operadores tales como *any*(p_1,p_2,...), *concat* (p_1,p_2,...), *exact*(*n*,*p*), *atleast*(*n*,*p*), *atmost*(*n*,*p*), entre otros [116]. En base a estas formas se definen consultas con la sintaxis *(query((f),(h)))*, donde *f* es la forma buscada, y *h* es el conjunto de valores que deben ser considerados de la forma "parámetro inicio fin"; también pueden combinarse diferentes formas en la misma consulta con los operadores *AND* y *OR*. El siguiente ejemplo define una forma compleja y luego la utiliza para resolver una consulta [116]:

- (*shape* ramp() (*concat Up Up*)): define una forma compleja llamada ramp compuesta por dos incrementos grandes consecutivos.
- (*query* ((ramp) (*confidence start end*))): busca todas las reglas que tuvieron dos incrementos consecutivos en su confianza (confidence) desde el primer dato (*start*) hasta el último conocido (*end*).

Además, se puede declarar gatillos (triggers) sobre la base de datos de reglas de tal manera que cuando el historial de las reglas comience a mostrar tendencias se ejecuten acciones programadas. Este mecanismo de la minería de datos activa puede ser usado para detectar una tendencia a datos de mala calidad y producir advertencias tempraneras [116].

3.7 - Algoritmos de minería de datos e implementaciones

Existen cientos de algoritmos diferentes para realizar minería de datos, muchos de ellos con distintas variantes. Por ejemplo, para realizar análisis asociativo se puede nombrar a los algoritmos A-Priori y FP-Growth; para la clasificación se pueden contar ZeroR, OneR, PARTR, J48, Random Tree y Random Forest; y para el agrupamiento se pueden mencionar K-Means, EM y Canopy. Para cada uno de ellos existen múltiples implementaciones en los más variados lenguajes de programación. También existen múltiples herramientas que implementan uno o más algoritmos con el objetivo de permitir a los usuarios aplicarlos sobre datos concretos, ya sea de forma académica o para uso industrial o comercial.

En el apéndice se proporciona mayor información sobre diferentes algoritmos, se describe con mayor detalle el software Weka, el cual fue utilizado para desarrollar los experimentos que se exponen en el capítulo 5, y se describen varios de los algoritmos implementados por dicho software.

4 - Minería de datos aplicada a la calidad de datos

En los capítulos previos se expusieron, por separado, dos temas de mucha importancia en la actualidad: la calidad de datos y la minería de datos. En este capítulo se analiza cómo se puede integrar ambas áreas de investigación con el objetivo de lograr una mejora en el proceso de gestión de la calidad de datos, tanto a nivel de resultados como de rendimiento, aplicando técnicas de minería de datos.

La gestión de la calidad de datos tradicionalmente se ha fundamentado en la visualización de los datos (análisis visual utilizando diferentes formas de presentación) soportado por análisis estadísticos clásicos [118]. Los datos que se detectan como incorrectos son corregidos si se puede ("limpiados") o marcados en otro caso para alertar a quien los utiliza de que no son plenamente confiables. Sin embargo, la cantidad de datos a evaluar se ha incrementado vertiginosamente, a la vez que las fuentes se han multiplicado y la representación de los datos se ha vuelto cada vez más heterogénea, por lo que las técnicas usuales son cada vez menos aplicables [118][029]. Al mismo tiempo los requerimientos de contar con datos de buena calidad se han incrementado, entre otras cosas porque en la actualidad muchas decisiones son tomadas automáticamente por sistemas computarizados a partir de los datos disponibles. El incremento significativo en el volumen de datos que pueden ser recolectados y almacenados, así como la mejora de los sistemas de información, ha conducido a recurrir a la minería de datos en varias áreas relativas al procesamiento de datos, entre ellas la gestión de calidad de los datos. Las razones para esto están asociadas con la capacidad de las técnicas de minería de datos para escalar bien en grandes volúmenes y con datos complejos, permitiendo extraer patrones frecuentes, encontrar reglas de asociación, clasificar información y predecir el futuro (determinar cómo se comportará una cierta variable en base a observaciones previas) [015][119].

4.1 - Qué es la minería de calidad de datos

La **minería de calidad de datos** (DQM, *data quality mining*) se define como la aplicación de técnicas de minería de datos con la intención de medir y mejorar la calidad de los datos que se recolectan, almacenan y procesan. El objetivo es detectar, cuantificar, explicar y corregir las deficiencias en la calidad de los datos en bases de datos [041][067][120]. La expresión "minería de calidad de datos" fue introducida por Jochen Hipp en 2001 ([121]) para abarcar técnicas de gestión de la calidad de datos utilizando minería de datos de forma similar a como se busca conocimiento. La idea detrás de esto es que hay solo diferencias marginales entre aplicar un algoritmo de minería de datos para descubrir nuevos patrones a partir de los datos y aplicar el mismo algoritmo para detectar deficiencias en los datos, y que las diferencias radican solo en cómo se aplican los patrones y como se interpretan los resultados [120]. La misma idea es manejada por otros autores bajo nombres diferentes, como "auditoría de datos" (*data auditing*) [122] o "minería de calidad" (*quality mining*) [023]. La aplicación de métodos de minería de datos para mejorar la calidad de datos es relativamente nueva y prometedora [123], aunque de escasa atención.

A grades rasgos, la minería de calidad de datos consta de dos etapas principales: una primera etapa de inducción de un modelo que permita representar las características de los datos, y una segunda en la cual se realiza la detección de desviaciones de los datos respecto de dicho modelo. Durante la inducción del modelo puede ser necesaria la participación de un experto en el dominio

capaz de ajustar los parámetros de forma adecuada [122].

Cuando se compara a la minería de calidad de datos con las técnicas usuales de análisis se observa que existen múltiples ventajas. La primera de ellas, que resulta obvia de la descripción de la minería de calidad de datos, es que escala bien para grandes volúmenes de datos y para datos complejos, algo que es fundamental en la actualidad. La segunda de las ventajas es que los modelos suelen ser más sencillos de construir e interpretar, principalmente porque son hechos de forma automatizada y con poca intervención de los analistas. Otra de las ventajas es que por las características propias de la minería de datos no es necesario que el analista determine completamente de antemano qué es lo que desea encontrar, sino que son las propias técnicas de minería de datos las que destacarán los patrones encontrados, quedando para el analista la determinación de cuáles de ellos resultan realmente interesantes. Finalmente, sabiendo cuáles son los patrones no conformados la herramienta de minería de calidad de datos es también capaz de sugerir correcciones a los datos o incluso prevenir la incorporación de datos que tengan el potencial de afectar negativamente la calidad del conjunto total de datos.

Un punto importante que debe ser tenido en cuenta es el de los tipos de datos de los atributos involucrados. Los tipos de datos pueden ser clasificados fundamentalmente en dos grupos: **discretos** y **continuos**. Los discretos, o categóricos, son aquellos que pueden enumerarse de alguna manera (lo que implica que pueden tener una cantidad finita de valores), mientras que los continuos son los que no pueden enumerarse, abarcando a los numéricos (enteros y reales), las fechas y textos en general; los textos que sí pueden enumerarse son usualmente denominados nominales y pertenecen a la categoría de discretos. Aunque es deseable que los algoritmos de minería de datos tengan la habilidad de trabajar con atributos tanto discretos como continuos la mayoría de ellos solo lo hacen con atributos discretos. Por lo general los atributos continuos pueden ser convertidos en discretos mediante la técnica llamada **encajonamiento** (*bin*), que consiste en definir un conjunto limitado de valores, llamados cajones, cada uno correspondiente a un subconjunto de los valores continuos originales, y remplazar el valor real de cada observación por el cajón correspondiente. Sin embargo pueden existir atributos que no admitan ser convertidos de tal manera ya que corren riesgo de pérdida de información, como es el caso de los datos científicos que requieren muy alta precisión [118].

La minería de calidad de datos no solo sirve para detectar problemas de calidad y colaborar en su solución (**mantenimiento reactivo** de la calidad), sino también para prevenir la degradación del conjunto de datos una vez que se alcanzó una cierta calidad (**mantenimiento preventivo**). Para esto, al momento de pretender incorporar nuevos registros a un conjunto de datos se puede mantener la calidad de dicho conjunto solo aceptando aquellos que conformen con los patrones, o tal vez corrigiendo a los que no lo hagan previo a su incorporación [118][121].

4.1.1 - Calidad de datos aplicada a la minería de datos

Si bien en este capítulo se hace un enfoque en la aplicación de técnicas de minería de datos para lograr mejorar la calidad de los datos también existe la relación inversa entre ambos conceptos: la calidad de datos tiene un efecto directo sobre los resultados de la minería de datos. Aunque la minería de datos ha demostrado ser efectiva para el descubrimiento de conocimiento en bases de datos, la experiencia marca que la calidad de los datos con los que se trabaja es esencial para determinar la confiabilidad y utilidad del conocimiento encontrado [023]. En particular, en la mayoría de los casos el descubrimiento de información en bases de datos se aplica en presencia de deficiencias en los datos, por lo que tratar la calidad de datos debe ser parte de cualquier proyecto de KDD, aunque usualmente se hace de una manera *ad-hoc* [121]. Esto implica que se está en

presencia de una circularidad: se pretende aplicar técnicas de minería de datos para abordar la calidad de los datos, pero la calidad de los datos a los cuales se pretende aplicar dichas técnicas también debe ser analizada previamente. En otras palabras, para analizar la calidad de los datos utilizando minería de datos es necesario contar con datos de buena calidad. Esto es un detalle no menor, ya que al momento de elegir métodos y técnicas de minería de datos debe tenerse en cuenta la forma en que son afectadas por la calidad de los datos.

4.2 - Trabajos relacionados

Si bien la expresión "minería de datos" fue propuesta en 2001, desde pocos años antes algunos investigadores habían comenzado a trabajar en propuestas para la aplicación de técnicas de minería de datos en el tratamiento de la calidad de los datos. A continuación se resume brevemente un relevamiento sobre trabajos existentes en el área, o relacionados de alguna manera con ésta; deliberadamente se presentan en orden cronológico, con el doble objetivo de mostrar cuán escasos son los trabajos existentes, y cómo ha evolucionado el área. Como se verá, varios de los nombres se repiten, reduciendo aún más el espacio de búsqueda.

Los comienzos del área se pueden ubicar en el año 1999, cuando Tamraparni Dasu y Theodore Johnson en un trabajo titulado "**Hunting of the snark: finding data glitches using data mining methods**" [119] describen algunos problemas para la detección de irregularidades en los datos (a las que llaman "data glitches"), y muestran cómo puede ser útil la minería de datos para su descubrimiento y resolución. Consideran una irregularidad cualquier cambio introducido en los datos por causas externas al proceso que los genera y que es diferente al ruido normal; esto incluye la reversión a valores por defecto, valores de atributos faltantes, registros omitidos, agregaciones con menor precisión que la de los datos originales, etc. En base a esto proponen un algoritmo llamado DataSphere que consiste en particionar los datos en secciones homogéneas y usar resúmenes representativos para analizarlos. Siguiendo la misma línea, en el año 2000 Jonathan Maletic y Andrian Marcus presentan dos trabajos, "**Data cleansing: beyond integrity analysis**" [124] y "**Utilizing association rules for the identification of errors in data**" [125], en los cuales evalúan diferentes métodos para realizar la limpieza de datos y detección de errores (estadísticos, agrupación, reconocimiento de patrones y descubrimiento de reglas de asociación), llegando a la conclusión de que lo que ellos llaman reglas de asociación ordinales, que son una extensión propia de las reglas de asociación tradicionales, son más eficientes que otros métodos, aunque solo pueden ser aplicadas a campos numéricos o fechas.

En 2001 Jochen Hipp, Ulrich Güntzer y Udo Grimmer presentaron su reconocido trabajo "**Data quality mining, making a virtue of necessity**" [121], en el cual introducen la expresión "minería de calidad de datos" (*data quality mining*). Este trabajo es actualmente considerado el punto de partida de la minería de calidad de datos. En él, los autores muestran cómo las técnicas de minería de datos pueden ser utilizadas para mejorar la calidad de los datos, y lo ejemplifican utilizando reglas de asociación: la idea general es utilizar un algoritmo de minería de datos para extraer reglas de asociación a partir de los datos y luego identificar aquellos registros que no las cumplen, permitiendo además asignar un puntaje (ranking) de desviación tomando en cuenta la confianza de las reglas violadas y no solo la cantidad. Como se verá más adelante en el documento este principio básico es el que ha fundamentado la gran mayoría de las técnicas de minería de calidad de datos.

En los años siguientes a ese trabajo el interés por la aplicación de la minería de datos en el abordaje de la calidad de los datos tuvo un crecimiento que parecía indicar que la "nueva" área se convertiría en una tendencia. En 2002 Satoshi Hori, Hirokazu Taki, Takashi Wahio y Hiroshi Motoda presentaron, en un artículo titulado "**Applying data mining to a field quality watchdog**

task" [126], un programa de vigilancia de datos automatizado (watchdog) que permitiría monitorear una base de datos y anticipar problemas en la operativa del negocio en base a la detección y aplicación de reglas de asociación. Al año siguiente (2003) Dominik Luebbers, Udo Grimmer y Matthias Jarke presentaron el artículo "**Systematic development of data mining-based data quality tools**" [122] donde explican porqué es necesario un mecanismo automático de auditoría y corrección de datos y cómo puede construirse una herramienta para auditoría de datos basada en minería de datos, dejando en claro que es necesario siempre la participación de un experto en calidad de datos que además tenga conocimiento del dominio. Además, dado que la evaluación de la calidad de los resultados obtenidos es difícil de hacer (determinar cuántos supuestos errores detectados son realmente errores, y cuántos errores reales no fueron detectados), proponen evaluar las herramientas en base a un conjunto de datos artificial que imite las regularidades de la base de datos real, y a otro conjunto de datos artificial perfecto al cual se le aplica un proceso de polución controlada. Ese mismo año nuevamente Grimmer junto a Holger Hinrichs presentaron "**A methodological approach to data quality management supported by data mining**" [120] donde proponen la utilización de minería de datos para evaluar y mejorar la calidad de los datos en forma sistematizada en ambientes de integración de datos de múltiples fuentes como es el caso de los data warehouses. Para esto proponen utilizar reglas de asociación, aunque aclaran que hay dos retos importantes que deben ser resueltos: la representación especial de los datos que requieren la mayoría de las reglas de asociación (por ejemplo, en cuanto al tipo de datos), y la gran cantidad de reglas que suelen devolver los algoritmos (demasiadas como para ser aprovechadas por las personas). También en 2003, en un reporte publicado por la NASA titulado "**Automated data quality assesment in the intelligent archive**" [118], David Isaac y Christopher Lynnes proponen la incorporación de técnicas de aprendizaje automático para gestionar la calidad de datos debido que, según consideran, el volumen de los datos que se recogen, y la velocidad a la que se hace, tornan imposible que las personas se encarguen de verificar la calidad de los datos obtenidos utilizando las técnicas usuales basadas en análisis estadísticos y visualización de información. Al hacerlo, indican, podrían crearse "archivos inteligentes", que sean capaces de reconocer cuando se están introduciendo datos que podrían comprometer la calidad global del archivo, reaccionar de forma apropiada ya sea rechazando los datos nuevos, corrigiéndolos mediante consultas a otras fuentes, o solicitando a los usuarios que los revean manualmente. También en 2003, en el mismo evento que [120] mencionado anteriormente, Sabrina Vázquez Soler y Daniel Yankelevich presentan en un artículo titulado "**Quality mining: a data mining based method for data quality evaluation**" [023] un mecanismo para el análisis sistemático de la calidad de datos, particularmente la correctitud, basado en reglas de asociación al que denominan QuAsAR (Quality Assessment using Association Rules).

Luego de un par de años de primer plano la minería de calidad de datos vio, aparentemente, disminuir su evolución. Durante un largo tiempo no se encuentran trabajos enfocados fuertemente en el área, y si bien sería posible incluir alguno que la tocara superficialmente ninguno parece tener a la minería de calidad de datos como tema central.

En 2006, en un artículo titulado "**Data quality mining in ontologies for utilities**" [127], Fabian Grüning propone utilizar ontologías en el proceso de minería de calidad de datos, modelando las conexiones entre las dimensiones de calidad de datos y las clases de algoritmos utilizados con el objetivo de medir la calidad de los datos y también el proceso de gestión de la calidad. La idea es añadir aspectos de calidad de datos a una ontología de dominio (como la CIM, Common Information Model, [128]) de forma de que los expertos en dicho dominio puedan expresar los problemas conocidos de calidad de datos sin necesidad de tener conocimientos profundos en la materia. Al año siguiente (2007), en un artículo titulado "**Data quality mining: employing**

classifiers for assuring consistent datasets" [041], el mismo autor propone utilizar máquinas de soporte vectorial (SVM) para medir y mejorar la calidad de los datos en la dimensión consistencia, anticipando que en futuros trabajos abordarían las otras dimensiones, como precisión, correctitud y actualidad (aunque no se ha encontrado ninguna referencia a esos otros trabajos). La idea central es que los clasificadores son útiles tanto para medir la consistencia de los datos como para mejorarla, ya que para ello basta con comparar el valor conocido con el valor determinado por el clasificador y si no coinciden remplazar el primero por el segundo. En forma simultánea, otra vez el mismo autor presenta "**Sasquatch: semantical approach of assuring high data quality by applying data mining techniques**" [129], una propuesta basada en [128] para construir un sistema de aseguramiento de la calidad basado en ontologías y minería de datos con dos ideas principales: por un lado, mapear los esquemas de datos de los sistemas de gestión a una ontología, preferiblemente específica de un dominio particular, para abstraer los datos del sistema particular, y por otro el uso de técnicas de minería de datos para obtener características de los datos que permitan indicar si un registro es correcto o no. El mismo año, en el trabajo "**Rule based measurement of data quality in nominal data**" [136] Jochen Hipp (el mismo que introdujo el término "minería de calidad de datos"), Markus Müller, Johannes Hohendorff y Felix Naumann proponen un modelo que permite evaluar la calidad de los datos bajo la dimensión exactitud (accuracy) y que consiste en catalogar los datos en tres conjuntos: "posiblemente correctos", "posiblemente incorrectos" y "sin decisión" mediante la utilización de reglas de asociación. En el año 2008 Fei Chiang y Renée J. Miller en "**Discovering data quality rules**" [130] proponen una metodología para identificar reglas que los datos deben cumplir para luego detectar los datos que nos las cumplen como posibles incorrecciones. La gran diferencia con las propuestas anteriores es que los autores se basan en dependencias funcionales condicionales (DFC) y no en reglas de asociación. En 2009, en "**Towards data quality and data mining using constraints in XML**" [131] Sumon Shahriar y Sarawat Anam abordan el análisis de la calidad de datos en documentos XML mediante el uso de técnicas de minería de datos y restricciones XML; en este caso definen a la calidad de datos como "la completitud y consistencia que puede lograrse mediante la aplicación de un conjunto de restricciones" (notar que no hacen mención a la correctitud o exactitud). En 2010 Saeed Farzi y Ahmad Baraani Dastjerd en su artículo "**Data quality measurement using data mining**" [019] presentan un método que, al igual que el original de Hipp, Guntzer y Grimmer, utiliza reglas de asociación aunque afirman que es más eficiente debido a que no es necesario calcular todas las reglas de asociación previamente, con lo que, según los autores, logran evaluar la calidad de una base de datos particular con el 71% de eficacia contra el 50% logrado con el método original.

A partir de 2010 se produjo un crecimiento importante de las propuestas basadas en la teoría de conjuntos difusos (FST, ver el capítulo 3.5.5). Ese año Fatemeh Ghorbanpour Alizamini, Mir Mohsen Pedram, Mohammad Alishahi y Kambiz Badie, en un artículo titulado "**Data quality improvement using fuzzy association rules**" [123], presentaron un método para medir la calidad de los datos bajo la dimensión exactitud (accuracy) usando sistemas difusos, para lo cual proponen mejorar el aprendizaje de reglas de asociación de forma de poder utilizar campos no discretos debido a que, si bien las reglas de asociación constituyen la técnica de minería de datos más estudiada y aplicable a la gestión de la calidad de datos, tiene el gran problema de que no lidian bien con valores continuos. En 2012, el artículo "**Proposing an improved semantic and syntactic data quality mining method using clustering and fuzzy techniques"** [132] de Hamid Reza Khosravani, se centra en la dimensión exactitud, tanto sintáctica como semántica, aunque en este caso recurre a las técnicas de agrupamiento para extraer reglas de validación, asignando automáticamente pesos a los atributos que indican cuánto aporta cada uno de ellos a la decisión. El método requiere contar con un conjunto de registros previamente saneado para poder determinar las reglas de validación iniciales.

Un enfoque bastante diferente a los anteriores es el presentado en 2012 por Thierno Diallo, Jean-Marc Petit y Sylvie Servigne en "**Discovering editing rules for data cleaning**" [133], donde proponen técnicas de aprendizaje de reglas de edición (ER) a partir de relaciones existentes respecto de una fuente de datos maestra considerada limpia (sin problemas de calidad), y que luego pueden utilizarse para evaluar y mejorar la calidad de datos de otras fuentes. Las reglas de edición son instrucciones que indican cómo debe modificarse un registro para cumplir con las condiciones impuestas. Los autores afirman que si bien las reglas de asociación y las dependencias funcionales condicionales ayudan a detectar errores en los datos, fallan al momento de indicar cuáles son los atributos involucrados y cómo realizar la corrección. El procedimiento propuesto por estos autores consta de tres pasos: 1) encontrar asociaciones uno-a-uno entre los registros de la fuente de datos a evaluar y el repositorio maestro; 2) descubrir reglas de edición a partir del repositorio maestro; 3) aplicar las reglas de edición en los datos a evaluar.

En 2013 la tesis de doctorado titulada "**A data mining approach to improve the automated quality of data**" [015], de Nawaf Abdullah Alkharboush, propone el uso de sistemas aproximados (ver el capítulo 3.5.6) para evaluar y mejorar la calidad de los datos. Se fundamenta en la idea de que los algoritmos de minería de datos, especialmente los de aprendizaje de reglas de asociación, están orientados a encontrar patrones frecuentes en los datos, mientras que la calidad de datos está orientada a lo opuesto, encontrar patrones de baja frecuencia que puedan ser indicadores de errores, lo que hace que la aplicación de minería de datos a la gestión de calidad de datos sea ineficiente. Propone entonces utilizar RST, lo cual sería más eficiente para la detección de extraños (outliers) que seguramente correspondan a registros con algún tipo de problema.

En el lado opuesto de los trabajos anteriores se encuentran varios que proponen apegarse a las técnicas estadísticas para abordar la gestión de la calidad de los datos. Por ejemplo, Diane Lambert en su artículo de 2003 "**What use is statistics for massive data?**" [134] defiende a la estadística ante quienes afirman que no es aplicable para grandes volúmenes de datos, y afirma que muchos estadísticos no tienen la capacidad, o las herramientas, para lidiar con datos masivos, pero que la estadística sí se puede aplicar a datos masivos. Adicionalmente compara cómo observa un estadístico un conjunto masivo de datos, y como lo hace un informático: los estadísticos consideran que el trabajo de analizar datos consiste en encontrar un modelo que establezca un valor central, correcto, y un factor de desviación o incertidumbre, mientras que los informáticos consideran que los datos no son ruidosos ni inciertos y que el trabajo consiste en analizarlos tal cual están. En el mismo sentido, en 2005, Alan F. Karr, Ashish P. Sanil y David L. Banks en "**Data quality: a statistical perspective**" [135] brindan un enfoque sobre cómo la estadística debería involucrarse en el abordaje de la calidad de los datos, utilizando análisis exploratorios de datos (EDAs, exploratory data analysis), árboles de clasificación y regresión (CART, classification and regression trees) o reglas de edición (ER, edit rules). Muestran ejemplos de su aplicación en dos casos reales aunque aclaran que solo dos casos no pueden abarcar todas las posibles situaciones de aplicabilidad.

4.3 - Aplicaciones existentes de la minería de calidad de datos

En este capítulo se analizan mecanismos sobre cómo se puede aplicar las diferentes técnicas de minería de datos para el aseguramiento de la calidad de los datos. Para hacer este análisis se parte de la propuesta de dimensiones y factores de calidad descrita en el capítulo 2.5. Las propuestas mencionadas en este capítulo pueden corresponderse a los trabajos relevados y presentados en el capítulo 4.2. En el capítulo 5 se describen tres propuestas nuevas propias.

Es importante aclarar que en algunos casos se recurre a técnicas que a primera vista podrían no encontrarse de pleno dentro del área de la minería de datos; sin embargo, si se considera la

definición amplia del concepto, cualquier técnica que permita obtener información o mejorar el conocimiento del mundo a partir de los datos en forma automatizada es considerada dentro de la minería de datos, y por lo tanto se le concede un espacio.

4.3.1 - Exactitud

La dimensión **exactitud** (**accuracy**, ver el capítulo 2.5.1) es, como se puede concluir luego de analizar el capítulo 4.2, la más abordada por los investigadores en el área de la calidad de datos y especialmente en la minería de calidad de datos. Esto es comprensible si se parte de la base de que para la mayoría de las personas una buena calidad de datos significa tener datos correctos y que no presenten errores [136].

4.3.1.1 - Correctitud semántica

La mejor forma de evaluar la correctitud semántica de un conjunto de datos es contrastarlos de alguna manera con el mundo real que pretenden representar, aunque esto difícilmente pueda ser hecho por varias razones, entre ellas la dificultad para abstraer el mundo real o que el mundo modelado puede no estar disponible (los datos corresponden al pasado o a una ubicación geográfica inalcanzable). En cambio, una técnica alternativa es la de construir un modelo a partir de datos sabidos semánticamente correctos y luego determinar qué tanto se ajustan esos datos, y otros, a dicho modelo. Esta técnica implica los siguientes pasos:

1. Obtener un conjunto de datos semánticamente correctos que represente en forma apropiada al mundo real. Este conjunto de datos es llamado **conjunto de datos de entrenamiento**.
2. A partir del conjunto de datos de entrenamiento, y como resultado de la aplicación de análisis asociativo (ver las secciones 3.3.3 y), obtener un **modelo descriptivo** del mundo real, representado como un conjunto de reglas de asociación con sus correspondientes medidas de confianza y respaldo.
3. Determinar qué tan bien se ajustan los **datos objetivo** (los que se pretende analizar) al modelo construido. Para esto se debe determinar cuántas reglas incumple cada observación; si resulta que una observación no incumple ninguna de las reglas se puede afirmar que es un dato semánticamente correcto (o tiene muy alta probabilidad de serlo), mientras que en otro caso se puede considerar candidata para una segunda revisión y potencialmente una corrección. En [121] proponen una mejora a este procedimiento básico: en lugar de solo considerar la cantidad de reglas incumplidas, considerar la calidad de las reglas incumplidas según la confianza de cada una: podría suceder que una observación particular esté en conflicto con varias reglas de asociación de baja confianza pero satisfaga otras reglas de asociación de muy alta confianza.
4. Tomar alguna acción con las observaciones que no logran un nivel de aceptabilidad mínimo. Esta acción podría ser presentar al analista el conjunto de observaciones que no resultan semánticamente correctas para que éste determine qué hacer con ellas (descartarlas, corregirlas, o aceptarlas). En el caso de pretender corregir una observación, podría hacerse de la siguiente manera [121]:
    1. Tomar el conjunto R de reglas de asociación que son incumplidas por la observación.
    2. Determinar el conjunto potencia, R^*, compuesto por todas las combinaciones posibles de reglas de R.
    3. Para cada elemento r^* de R^* calcular el puntaje de la observación resultante asumiendo

que cumple todas las reglas de r^*.

4. Tomar el elemento r^* cuyo puntaje sea mayor y modificar la observación para que deje de incumplir las reglas de r^*. Si esto no fuera posible, considerar el siguiente r^*, y así sucesivamente.

El procedimiento anterior tiene inconvenientes que deben ser considerados con atención:

- Para el paso 1 se requiere contar con un conjunto de datos de entrenamiento semánticamente correcto, lo que puede resultar muy difícil de conseguir. La construcción manual de un conjunto de datos de entrenamiento por parte expertos en el dominio no parece una opción viable ya que consume demasiado tiempo y esfuerzo, y no habría certeza de que no contenga errores semánticos propios, que resulte sesgado o incompleto. En su lugar se pueden analizar dos alternativas: por un lado, en la línea presentada en [122] se podría desarrollar un sistema generador de datos de entrenamiento a partir de un conjunto reducido de reglas proporcionadas por los expertos, aunque esto parece presentar el mismo problema que el caso anterior ya que no hay certeza de que no se cometan errores u omisiones; la otra alternativa es utilizar como conjunto de datos de prueba una parte del conjunto de datos objetivo y confiar en la ley de los grandes números que establece que los defectos individuales pasan desapercibidos en el conjunto cuando se manejan grandes volúmenes de datos, por lo que si se trabaja con un conjunto suficientemente grande las reglas tienden a rescatar las regularidades y a despreciar las particularidades.
- Para el paso 2 se requiere ejecutar un análisis asociativo, lo que puede consumir mucho tiempo y suele devolver como resultado una cantidad muy grande de reglas, tal vez demasiadas para ser tratadas en los pasos siguientes con eficiencia. Respecto del tiempo de ejecución, muchos autores han propuesto mejoras y optimizaciones a los algoritmos clásicos de descubrimiento de reglas de asociación. Respecto de la cantidad de reglas también hay algunas alternativas a considerar: eliminar las reglas que no tengan una suficiente confianza (en [121] establecen el umbral en 75%), o exigir que las reglas tengan (o no tengan) ciertos atributos en el antecedente.

4.3.1.2 - Correctitud sintáctica

La correctitud sintáctica es algo más fácil de evaluar en forma automática que la correctitud semántica, principalmente porque no está relacionada con los valores en sí mismos sino con la representación de los datos. La evaluación de la correctitud sintáctica puede estar dada simplemente por la validación de reglas sintácticas, las cuales son usualmente establecidas por los diseñadores de los sistemas de información. Sin embargo, si realmente la correctitud sintáctica correspondiese a la validación de reglas predefinidas sería más sencillo validar los datos durante su ingreso, y de esa manera no existirían los problemas de correctitud sintáctica en la base de datos. Lo que sucede es que cuando se construye un sistema de información se omite la definición o implementación de ciertos controles, a veces en forma deliberada y otras por omisión o desconocimiento, lo que lleva a que con el tiempo y el uso los datos pierdan calidad, sin considerar que incluso dichas reglas pueden ir mutando con el tiempo (por ejemplo, lo que en un momento era un número entero luego puede ser un número decimal, y lo que antes era un valor binario luego pasa a admitir otros valores no imaginados inicialmente). Además, es usual que los sistemas informáticos sean modificados con el tiempo para adaptarlos a nuevas necesidades o a cambios en la realidad, y estas modificaciones suelen ser hechas con menor cuidado y por personas diferentes que usualmente no tienen las mismas preocupaciones, lineamientos y conocimiento que los autores originales.

La detección de las reglas sintácticas encajan en lo que se llama **perfilamiento de datos** (***data***

profiling), que consiste en aplicar diversas técnicas para extraer las reglas sintácticas básicas a partir de los datos, principalmente referidas a los metadatos más que a los propios datos [137][138]. Entre las propiedades más destacadas se encuentran la cantidad de valores totales, la cantidad de valores diferentes, los tipos de datos, patrones frecuentes entre los datos, correlaciones entre datos, dependencias funcionales y de inclusión, entre otras [137]. De esta manera, el objetivo principal del perfilamiento de datos es descubrir las características de los datos, particularmente de la base de datos que los contiene, examinando únicamente a los propios datos. Normalmente estas bases de datos son desconocidas y carecen de otra información tal como documentación o metadatos incorporados; por esta razón, en algunos casos, el perfilamiento de datos en una base de datos es referido como **ingeniería reversa** de la base de datos [137].

Las técnicas y objetivos del perfilamiento de datos se caracterizan según se apliquen sobre un atributo de los datos (en términos relacionales, sobre una única columna de una tabla) o sobre más de un atributo a la vez. En el primer caso la información buscada suele considerar diferentes tipos de conteos (valores totales, valores únicos, valores nulos y no-nulos), tipo de datos (detectar si el atributo es textual, numérico, etc.), valores máximos y mínimos, patrones que modelen a los datos (en forma de expresiones regulares), candidatos a claves, etc. En el segundo caso se buscan dependencias entre diferentes atributos (de la misma entidad o de múltiples entidades, en términos relacionales entre columnas de la misma o de diferentes tablas), tales como correlaciones, inclusiones, dependencias funcionales, etc. Las técnicas más usuales para el perfilamiento monoatributo (de un solo atributo a la vez) en bases de datos relacionales constan de consultas SQL y la generación de histogramas (representaciones gráficas de resultados). Sin embargo, en ambos casos las organizaciones suelen utilizar herramientas especialmente diseñadas para el trabajo que no brindan mucha información sobre qué y cómo lo hacen, entre ellos *Information Analyzer* de IBM, SQL *Server Integration Services* (SSIS) de Microsoft, *Data Explorer* de Informatica [137].

Se debe tener en cuenta que normalmente los datos en sí mismos tienen deficiencias, por lo que no se puede esperar que todas las observaciones analizadas cumplan con las reglas detectadas, o expresado de otra manera, si se espera encontrar reglas que sean cumplidas cabalmente por absolutamente todos los datos probablemente no se encuentre ninguna. Es por eso que se suele hablar de perfilamiento imperfecto, donde se buscan reglas que sean cumplidas por un gran número de observaciones (por ejemplo, más del 95%) [137].

La complejidad computacional del perfilamiento de datos depende tanto del número de observaciones disponibles (filas) como del número de atributos de cada una (columnas), en parte porque muchas de las tareas involucran el ordenamiento de datos, y otras la combinación de atributos [137]. Esto requiere que las técnicas de perfilamiento escalen bien para contemplar las cada vez más grandes bases de datos de la actualidad. Esto indica que el perfilamiento de datos es un campo de aplicación de la minería de datos muy interesante. De hecho, en [137] los autores intentan separar una de la otra, pero terminan aceptando que tal separación no es absoluta.

4.3.1.3 - Precisión

La precisión declara la cantidad de información que se tiene sobre cada aspecto individual de las observaciones y la granularidad con que está representada. Muchas veces ocurre que algunos atributos de los datos son demasiado amplios y terminan siendo usados de manera no homogénea (cuando en el mismo atributo se almacenan datos con diferente precisión). La evaluación de la calidad de un conjunto de datos respecto de la precisión debe hacerse a nivel de cada atributo en forma individual, pudiendo posteriormente determinarse una métrica global para la observación.

Los casos en los que se visualiza más clara la necesidad de evaluar y mejorar la calidad de los

datos respecto de la precisión es en los atributos de texto (donde se permite ingresar texto libre) y en los temporales (fechas). Los atributos de texto más representativos son los que están pensados para contener nombres, direcciones y teléfonos. En el caso de los nombres muchos sistemas solo ofrecen un campo único para el dato dando lugar a que cada persona lo ingrese de la forma que le parezca: "Juan Pérez", "Ing. Juan Pérez", "Pérez, Juan", "J. Pérez", y un sinfín de alternativas. De esa manera la precisión de los datos es muy baja, y ciertas tareas automatizadas se ven entorpecidas o impedidas: no es posible indexar a los clientes por su apellido o enviarles correspondencia llamándolos por su nombre de pila, entre otras dificultades. En el caso de las direcciones ocurre lo mismo pero con mayor complejidad (calle, esquinas, número de puerta, apartamento, barrio, localidad, departamento, estado, región, país, código postal, sector, bloque, torre, kilómetro, etc.). También con los teléfonos, donde a menudo es necesario indicar el tipo de teléfono (fijo, móvil, fax) o vínculo (personal, laboral), o prefijo, etc. Todos estos casos se enmarcan en lo que se denomina **segmentación de textos** [139] y que consiste en transformar texto que fue ingresado en forma libre a un formato estructurado. Las técnicas existentes para esto se basan en tomar el texto libre de formato, dividirlo en unidades menores, llamadas ***tokens***, y luego ejecutar consultas y algoritmos de unificación sobre repositorios de datos especializados [135]. Las técnicas usuales de segmentación de textos pueden ser clasificadas en basadas en reglas (*rule-based*) o basadas en modelos supervisados (*supervised model-based*) [139]. Las primeras requieren que un experto en el dominio diseñe y mantenga un conjunto determinado de reglas, mientras que las segundas tratan de aprender automáticamente modelos de segmentación a partir de un conjunto de datos de entrenamiento compuestos por textos sin formato y sus correspondientes valores segmentados. El problema en este caso es que es difícil conseguir datos de entrenamiento para comprender todas las posibles características [139].

4.3.1.4 - Consistencia interna

La consistencia interna está dada por el cumplimiento de reglas semánticas entre atributos de un mismo registro de datos (observación). Aunque a nivel de calidad de datos es un factor que se diferencia de la correctitud semántica, cuando se analizan las posibilidades de aplicar técnicas de minería de datos se encuentra que para determinar el nivel de consistencia interna de una observación se puede usar el mismo procedimiento que para la correctitud semántica: determinar las reglas de asociación que rigen a la masa de observaciones para luego verificar cuáles reglas no son cumplidas por cada una de las observaciones objetivo.

4.3.1.5 - Consistencia externa

La consistencia externa se trata de establecer si ciertos atributos de una determinada entidad, llamada entidad secundaria, hacen referencia a algunos atributos de otra entidad (podría ser la misma), llamada entidad primaria, y si dicha referencia es cumplida por los datos disponibles. En el modelo relacional esto es denominado **integridad referencial**. Cuando las entidades tienen definidas las restricciones de este tipo y el sistema informático utilizado para el ingreso y almacenamiento de datos implementa mecanismos para forzar su cumplimiento (por ejemplo mediante claves foráneas) la consistencia externa está prácticamente asegurada, ya que el sistema impide registrar datos que incumplan las restricciones. Por otra parte, cuando las restricciones de integridad referencial no están bien definidas en el sistema informático pero son conocidas por los expertos puede realizarse una verificación manual determinando si las observaciones cumplen las restricciones. Sin embargo, cuando las restricciones de integridad referencial no son conocidas, como podría ser el caso de los sistemas legados, en bases de datos pobremente diseñadas y documentadas, o en base de datos no relacionales, se debe intentar deducir las restricciones de integridad referencial a partir de los datos disponibles, por ejemplo examinando la variedad de

valores diferentes de una propiedad particular en una entidad, y los valores de cada uno de las propiedades de otra entidad. Esto puede ser especialmente útil para bases de datos relacionales mal definidas, o para bases de datos estructuradas no relacionales (referenciales), tales como XML o LDAP. Sobre este asunto existen muchos trabajos, pero en éste no se ha profundizado en el tema.

4.3.2 - Completitud

La completitud (ver el capítulo 2.5.2) es una dimensión que indirectamente también ha recibido bastante atención por parte de la comunidad. Básicamente se trata de estimar cuántos datos sobre el mundo real se tienen y poder completar los faltantes o determinar que cierto dato faltante en realidad es correcto que esté omitido por las características del entorno.

4.3.2.1 - Cobertura

La cobertura hace referencia a cuántos de todos los posibles casos observables del mundo real están representados por los datos disponibles. Esto es una tarea muy difícil de determinar, debido a que solo observando los datos disponibles es posible decir cuáles casos del mundo real están representados, pero no cuáles no lo están.

Sin embargo, se puede analizar una alternativa: si todas las propiedades de los datos fuesen discretas (en caso de no serlo se podría aplicar alguna técnica de discretización) sería posible determinar todas las posibles combinaciones de ellas, y retener aquellas que estén presentes en los datos disponibles; a continuación se debe determinar cuáles de las restantes combinaciones son factibles pero no están representadas por los datos (pueden ocurrir en el mundo real pero no han sido observadas aún), para lo cual podría requerirse la intervención de expertos en el dominio o el acceso a fuentes de datos externas a partir de las cuales determinar reglas que permitan excluir combinaciones.

No se debe descuidar que el factor cobertura está relacionado con la cantidad de posibles combinaciones diferentes que deberían estar representadas entre los datos, y con la cantidad de observaciones correspondientes a cada combinación. Por ejemplo, dada una entidad *A* con un identificador a_0 y dos atributos discretos a_1 y a_2, cada uno de ellos con dos valores v_{11} y v_{12} para a_1 y v_{21} y v_{22} para a_2, es fácil notar que existen cuatro combinaciones diferentes: $\{v_{11},v_{21}\}$, $\{v_{11},v_{22}\}$, $\{v_{12},v_{21}\}$ y $\{v_{12},v_{22}\}$. Si entre los datos están representadas las cuatro combinaciones entonces podría decirse que hay 100% de cobertura; sin embargo, también podría decirse que si el universo de instancias de *A* está compuesto por 10 elementos, pero solo se tienen 4 observaciones, correspondientes a una combinación diferente cada una, entonces la cobertura sería apenas 40%. Según el procedimiento descrito se puede analizar la cobertura desde la primera alternativa pero no desde la segunda; de hecho, en principio parecería que respecto de la segunda alternativa no existe un mecanismo válido sin recurrir a expertos o fuentes externas.

4.3.2.2 - Densidad

Este factor de calidad es abordado especialmente en el capítulo 5.1, donde se presenta una propuesta concreta para su evaluación.

4.3.3 - Actualidad

Esta dimensión de calidad se aborda especialmente en el capítulo 5.2 donde se presenta una propuesta novedosa para estudiar la calidad de datos desde el punto de los factores de calidad vigencia, frescura y volatilidad utilizando técnicas de agrupamiento (clústering).

4.3.4 - Usabilidad

La dimensión usabilidad es abordada en el capítulo 5.3 a través de una propuesta de aplicación de técnicas de minería de datos.

5 - Propuesta de evaluación de la calidad de datos aplicando técnicas de minería de datos

En el capítulo 4 se mostraron aplicaciones existentes de las técnicas de minería de datos en tareas de análisis y mejoramiento de la calidad de los datos. En este capítulo se describen dos propuestas propias que fueron presentadas en diferentes eventos en los últimos años [048][140]. Ambas propuestas fueron ideadas con el objetivo primario de abordar dimensiones de calidad de datos, y particularmente factores de calidad de datos, que hubiesen recibido poca atención previamente, o para los cuales la aplicación de técnicas de minería de datos pudiera resultar de particular utilidad pero no se hubiese explorado lo suficiente aún. A lo largo del documento se explicó cómo la dimensión exactitud (2.5.1) es por lejos la que mayor atención ha recibido y sobre la cual existe mayor cantidad de trabajos, y en especial los factores correctitud semántica y sintáctica se han beneficiado de dicho trabajo. Por el contrario, la dimensión actualidad, con sus factores vigencia, frescura y oportunidad, ha sido prácticamente ignorada, y sin embargo es una dimensión que, como se explicó en el capítulo 2.5.3, puede resultar crucial para la operativa de las organizaciones y la toma de decisiones acertadas. En otro orden, la dimensión completitud (2.5.2), si bien ha recibido bastante atención, es por demás interesante para ser analizada mediante la aplicación de técnicas de minería de datos por las particularidades que presentan las estructuras de datos requeridas y las tareas llevadas a cabo para su estudio.

Adicionalmente, en este capítulo se introduce una tercera propuesta que si bien permanece en etapa de formulación y para la cual no se han presentado trabajos concretos se considera que puede representar un punto de partida para abordar la dimensión de calidad usabilidad.

5.1 - Análisis de la densidad de un conjunto de datos aplicando técnicas de minería de datos

5.1.1 - Descripción del procedimiento

Usualmente los valores faltantes son clasificados en tres categorías: **faltantes completamente aleatorios** (MCAR, *missing completely at random*), donde no hay un patrón que explique porqué faltan valores para un atributo; **faltantes aleatorios** (MAR, *missing at random*), donde sí podría encontrarse un patrón que vincule los valores faltantes con los valores de otros atributos; y **faltantes no aleatorios** (MNAR, *missing not at random*), donde los valores faltantes para un atributo solo dependen del propio atributo [141]. En principio MCAR y MNAR son similares ya que únicamente observando a los datos no es posible determinar cuál es el caso. Y aunque existen varias técnicas que abordan el tema casi todas ellas se enfocan en resolver el "*problema de los valores faltantes*", asumiendo que si hay valores faltantes entonces algo está mal y debe ser corregido, para lo cual se suelen utilizar técnicas tales como la imputación de valores (utilizando valores fijos, aleatorios o calculados a partir de otros valores disponibles) o incluso eliminando por completo los registros problemáticos [142][143][144][145] (está claro que eliminar datos para evitar problemas no es la mejor solución, pero para algunos autores, como se desprende de los trabajos citados, es preferible trabajar con menor cantidad de datos pero sin deficiencias que hacerlo con mayor cantidad de datos deficientes). Sin embargo, como se explicó en el capítulo 2.5.2, no siempre la ausencia de un valor implica un problema de calidad de datos, y al mismo tiempo la no ausencia de un valor, es decir, la

existencia de un valor concreto, sí podría representar un problema de calidad de datos.

La propuesta que se describe a continuación fue presentada en el **Alberto Mendelzon International Workshop on Foundation of Databases and the Web** (**AMW 2017**) realizado en junio de 2017 en Montevideo, Uruguay, y publicada en [048]. En ella se sugiere dar un paso atrás y evaluar primero si ante la ausencia o no de un valor para un determinado atributo en una observación particular se está ante un problema de calidad o no en lugar de comenzar asumiento que se trata de un problema de calidad que debe ser solucionado.

La propuesta consiste en tomar un conjunto de observaciones y para cada una de ellas determinar si sus propiedades pueden tener la representación que tienen en cuanto a la ausencia o presencia de un valor concreto, es decir, evaluar si cada valor nulo puede ser nulo (o por el contrario debería haber un valor concreto), y recíprocamente si cada valor no nulo es correcto que no lo sea (o si debiera ser nulo). Para ello esta técnica evalúa una propiedad a la vez construyendo un modelo de clasificación binario con dos clases, "*nulo*" y "*no-nulo*", utilizando las restantes propiedades como clasificadores.

Una vez obtenido el conjunto de observaciones a evaluar el algoritmo puede esquematizarse de la siguiente manera:

1. Eliminar todas las propiedades que funcionen como claves. Esto es necesario ya que si se utilizaran como clasificadores generarían reglas del estilo "si *clave=valor → propiedad={nula,no-nula}*", las cuales son triviales y podrían ser incorrectas.
2. Seleccionar la propiedad que se desea estudiar, la cual será llamada **propiedad bajo estudio**. El procedimiento descrito funciona evaluando una propiedad a la vez.
3. Entre las propiedades remanentes (luego de quitar las claves y la propiedad bajo estudio), seleccionar aquellas que podrían ser útiles como clasificadores para la propiedad seleccionada. Esta tarea suele resultar bastante desafiante, al punto que constituye un campo de investigación por sí mismo denominado **aprendizaje de características** (*feature learning*) [146], pero una técnica sencilla consiste en descartar aquellas propiedades que a simple vista tengan muchos valores nulos (esto es porque las técnicas de clasificación suelen requerir que los atributos clasificadores no tengan valores nulos).
4. Para las propiedades seleccionadas como clasificadores discretizar los valores que no sean discretos. Esto es necesario debido a que los algoritmos de clasificación suelen trabajar mejor cuando todas las propiedades son discretas. En el caso de las propiedades de tipo texto, donde usualmente los valores admitidos son muy disímiles, alcanza con saber si hay un valor o no, por lo que se sugiere remplazar cualquier valor no nulo por un texto fijo (por ejemplo, "*NONULO*"). En los restantes casos (números y fechas) la técnica más usual de discretización es la llamada encajonamiento, consistente en definir rangos de valores (aunque también puede ser útil la táctica del valor constante).
5. Clasificar manualmente las observaciones. Dado que las clases definidas son solo dos (*nulo* y *no-nulo*) la clasificación manual consiste en asignar la clase *no-nulo* a aquellas observaciones que tengan un valor concreto en la propiedad bajo estudio, y la clase *nulo* a las restantes. Luego, la propiedad bajo estudio debe ser también removida ya que de ahora en más la clase toma su lugar (se debe recordar que no se está evaluando cada valor concreto de la propiedad sino el hecho de que exista o no un valor para ella en cada observación).
6. Construir un modelo de clasificación. Para esto se debe utilizar un algoritmo de clasificación (ver las secciones 3.3.1 y 3.5); si bien se puede utilizar cualquiera conocido se recomienda

utilizar un algoritmo que además de indicar la clase para cada observación proporcione también un indicador de la confianza de la predicción, como es el caso de los árboles de decisión (ver el apéndice) (aunque casi todos los algoritmos lo hacen de una u otra forma). Además, si se asume que las deficiencias en los datos son la excepción y el conjunto de observaciones es suficientemente grande, entonces es válido utilizar como conjuntos de entrenamiento y de prueba todo el conjunto de observaciones ya que los problemas de densidad se perderán en la magnitud del modelo construido.

7. Aplicar el modelo de clasificación al conjunto de observaciones completo. Tras esto se obtendrá para cada observación individual una predicción indicando si la propiedad bajo estudio para ella debe ser *nulo* o *no-nulo*; además, si el algoritmo lo soporta, también se obtendrá la medida de confianza de la clasificación.
8. Evaluar los resultados obtenidos. La forma de hacer esto depende de si la clasificación tiene una medida de confianza o no:
    - Si la clasificación tiene medida de confianza:
        - Si la medida de confianza es mayor a un cierto umbral establecido de antemano, entonces se puede asumir que la clase indicada por el modelo es correcta, dando lugar a dos escenarios:
            - Si la clase asignada por el modelo (nulo/no-nulo) coincide con la clasificación realizada manualmente, entonces la observación no presenta ningún problema de densidad (incluso si el valor de la propiedad es nulo).
            - En otro caso la observación sí presenta un problema de densidad: tiene un valor nulo cuando no debiera tenerlo, o viceversa.
        - En otro caso no se tiene los suficientes elementos para tomar una decisión, siendo así la observación candidata para una revisión manual por parte de un experto.
    - Si la clasificación no tiene una medida de confianza la única alternativa es asumir que el modelo de clasificación es preciso y actuar como si se estuviera en el caso anterior con una confianza superior al umbral. Queda en evidencia la importancia de aplicar un algoritmo de clasificación que informe sobre la confianza de las clasificaciones.

5.1.2 – Ejemplo de aplicación

A continuación se describe un ejemplo de aplicación de la propuesta, utilizando un conjunto de datos ficticio pero con las características adecuadas para la ejemplificación del método.

Se considera la siguiente realidad: una empresa aseguradora mantiene una base de datos de todos sus clientes, pasados y actuales, incluso de aquellos que han solicitado la afiliación pero aún no han completado el procedimiento. Cada cliente es descrito en los siguientes términos: un identificador (no existen dos clientes con el mismo identificador; este identificador es asignado automáticamente por el sistema informático al momento de su registro por lo que se sabe que nunca puede ser omitido), su nombre y apellido, su año de nacimiento, su sexo y su profesión; además, también se registra la cuota mensual (que depende de múltiples factores como se verá a continuación), y un indicador de si ha sido dado de baja y la razón de dicha baja. La Tabla 4 muestra un ejemplo del conjunto de datos (los valores nulos son marcados con “-”).

Id	Nombre	FechaNac	Sexo	Profesión	Cuota	Baja	RazónBaja
1	A	1962	M	Albañil	5000	No	-
2	B	1948	M	Músico	-	Sí	No indicado
3	C	1951	F	Empleada	4500	Sí	Costo de la cuota
4	D	1971	F	Taxista	3000	-	-
5	E	1990	O	Estudiante	1000	No	-
…	…	…	…	…	…	…	…

Tabla 4: Ejemplo de datos sin procesar

Se observa a simple vista que el identificador, el nombre, el año de nacimiento, y la profesión no tienen valores nulos. Eso debería ser lo normal ya que esos datos son recabados al momento de que el cliente solicita la inscripción. Seguramente un sistema informático bien diseñado no permitirá registrar al cliente si alguno de esos datos falta; si bien es posible que el cliente se niegue a proporcionar alguno de esos datos, o no tenga un valor concreto para informar (por ejemplo, si es desempleado), siempre el funcionario a cargo del registro puede imputar un valor para satisfacer el requerimiento (el mismo podría no ser cierto, incluso el cliente podría mentir, pero en ese caso se estaría ante un problema de exactitud y no de completitud). Por otro lado, los campos cuota, baja y razón sí tienen valores nulos. Surgen entonces preguntas como las siguientes:

- ¿Es correcto que un cliente no tenga una cuota registrada? La respuesta es “tal vez”: si el cliente ha sido dado de baja entonces no tiene una cuota para pagar, o si aún no fue calculada su póliza tampoco se sabe cuánto debe abonar mensualmente.
- ¿Es correcto que no se sepa si un cliente fue dado de baja? La respuesta es “con seguridad no”.
- ¿Es correcto que no se sepa la razón de la baja de un cliente? La respuesta es “depende de si está dado de baja o no”; si no está dado de baja, no hay ningún valor correcto para el campo, pero si el cliente está efectivamente dado de baja entonces el campo debería tener un valor, y la ausencia de uno es una deficiencia.

Aplicando el paso 1 del procedimiento propuesto, se debe eliminar todas las claves. Si no se hiciese esto, cualquier algoritmo de clasificación produciría reglas del tipo “*si Id=1 → Cuota=5000*” y “*si Id=1 → RazonBaja=null*”; ambas reglas son triviales y coyunturales. En este caso el atributo *Id* debe ser eliminado; a simple vista ninguno de los restantes atributos es clave.

Aplicando el paso 2 se debe seleccionar el atributo que se desea evaluar. Entre los atributos resultantes luego de eliminar la clave, *Nombre*, *FechaNac* y *Sexo* parecen no ser candidatos, ya que ninguno de ellos debería poder ser nulo: todas las personas tienen un nombre, un año de nacimiento y un sexo, y aunque el cliente no lo quiera informar es claro que debe haber un valor, pudiendo el funcionario imputar uno de alguna manera (estimado, aleatorio, constante, etc). Se debe recordar que lo que se desea evaluar es si es correcto que haya un valor o no, sin importar el valor concreto en caso de haberlo. El campo *Profesion* se puede tratar de igual manera que los anteriores, por las mismas razones aunque no sea totalmente claro que siempre deba haber un valor cierto. Los últimos tres, *Cuota*, *Baja*, *RazonBaja*, son los que parecen más interesantes para su evaluación desde el punto de vista de la densidad. Para lo que resta del ejemplo se selecciona (en forma arbitraria) el atributo *Cuota*.

El tercer y cuarto paso del procedimiento indican que se debe seleccionar los atributos que serán

utilizados como clasificadores, y éstos deben ser discretizados en el caso de que no lo sean. Dado que los atributos *Baja* y *RazonBaja* tienen valores nulos, por ahora se probará sin ellos como clasificadores, quedando los restantes como tales (los algoritmos de clasificación no suelen trabajar bien con clasificadores con valores nulos, por lo que es necesario aplicar algún mecanismo de imputación de valores, lo que, de no hacerse bien, podría introducir un sesgo difícil de detectar). Luego, la discretización de los restantes atributos puede hacerse de la siguiente manera:

- El nombre del cliente difícilmente sea relevante para establecer cuánto debe pagar mensualmente por su póliza, por lo que todos los valores pueden ser sustituidos simplemente por el texto "*UNVALOR*" (lo que tiene un efecto similar a descartar el atributo como clasificador).
- El año de nacimiento, al ser un valor entero natural, se puede aplicar la técnica de definición de intervalos (encajonamiento), escogiendo los siguientes: "*NIÑO*", "*ADOLESCENTE*", "*JOVEN*", "*ADULTO*", "*MAYOR*", "*ANCIANO*" (los límites reales no son relevantes para el ejemplo); al mismo tiempo parece apropiado renombrar el atributo como "*FranjaEtarea*".
- El sexo ya es un valor discreto (uno de "*M*" por masculino, "*F*" por femenino, y "*O*" por otro).
- La profesión, en principio podría tratarse de igual manera que el nombre. Sin embargo, es claro que la similitud no es tal, ya que las diferentes profesiones conllevan diferentes riesgos laborales, siendo unas más riesgosas que otras (lo que para una empresa aseguradora es un indicador sumamente importante). Entonces, la discretización más apropiada parece ser la de selección de valores tabulados, escogiendo los siguientes: "*MUYALTO*", "*ALTO*", "*MEDIO*", "*BAJO*", "*MUYBAJO*", renombrando al atributo como "*RiesgoProfesion*".

El quinto paso establece que se debe clasificar manualmente las observaciones, asignado uno de "*NONULO*" o "*NULO*" según el atributo seleccionado (*Cuota*) tenga o no un valor concreto.

De esta manera, la Tabla 4 es transformada en la Tabla 5.

Nombre	**FranjaEtarea**	**Sexo**	**RiesgoProfesion**	**Cuota**
UNVALOR	ADULTO	M	ALTO	NONULO
UNVALOR	MAYOR	M	BAJO	NULO
UNVALOR	MAYOR	F	MEDIO	NONULO
UNVALOR	ADULTO	F	MEDIO	NONULO
UNVALOR	JOVEN	O	MUYBAJO	NONULO
…	…	…	…	…

Tabla 5: Ejemplo de datos procesados

Una vez que el conjunto de datos ya fue preprocesado, el sexto paso del procedimiento propuesto consiste en la aplicación de un algoritmo de clasificación, con el objetivo de construir un modelo de clasificación que pueda ser utilizado para clasificar observaciones no vistas anteriormente. En este caso, dado que se emplea Weka (ver el apéndice), se utilizará uno de los que dicho software proporciona, en particular el algoritmo denominado Random Tree. El criterio para seleccionar Random Tree es que en general los algoritmos basados en árboles son fáciles de interpretar y bastante eficientes, pero potencialmente cualquier otro algoritmo podría servir (excepto los más

básicos como ZeroR, OneR, etc).

Dado que no se tiene un conjunto de observaciones sabidas perfectas (sin deficiencias desde el punto de vista de la densidad) se ejecutó el experimento de varias maneras distintas:

- Utilizando todo el conjunto de observaciones como conjunto de prueba: precisión de 69,8% (698 de las 1000 observaciones fueron clasificadas correctamente), y kappa = 0.3322.
- Utilizando k-folds-validation, con k=10: precisión de 65.3%, y kappa = 0.2235.
- Utilizando división de muestra, con 66% para entrenamiento y 33% para prueba: precisión de 66.2%, y kappa = 0.2548.
- Utilizando división de muestra, con 75% para entrenamiento y 25% para prueba: precisión de 64.8%, y kappa = 0.2192.
- Utilizando división de muestra, con 90% para entrenamiento y 10% para prueba: precisión de 67.4%, y kappa = 0.2683.

Dado que la alternativa que mejores resultados arroja es la primera, se la toma como base para la evaluación de los resultados, un ejemplo de los cuales se muestran en la Tabla 6.

Nombre	**FranjaEtarea**	**Sexo**	**RiesgoProfesion**	**Cuota**	**Predicción**	**Confianza**
UNVALOR	ADULTO	M	ALTO	NONULO	NONULO	0.533
UNVALOR	MAYOR	M	BAJO	NULO	NULO	0.625
UNVALOR	MAYOR	F	MEDIO	NONULO	NULO	0.625
UNVALOR	ADULTO	F	MEDIO	NONULO	NONULO	0.643
UNVALOR	JOVEN	O	MUYBAJO	NONULO	NONULO	0.952
…	…	…	…	…	…	…

Tabla 6: Resultado de la evaluación con el método propuesto

Estableciendo un umbral de confianza en 0.66 se observa que el 63% de las observaciones son clasificadas por encima de él, 23% de las cuales reciben una clasificación diferente a la que muestra la observación. Ese 23% (14% del total de las observaciones) probablemente presenten problemas de densidad en cuanto al atributo *Cuota* (hay un valor nulo cuando no debiera haberlo, o hay un valor no nulo cuando debiera ser nulo). Esas observaciones son grandes candidatas para una revisión manual y una corrección. Por otra parte, de ese 63% clasificado con alta confianza, 77% (49% del total de las observaciones) son clasificadas por el modelo con la misma clase que se asignó manualmente, por lo que todo parece indicar (coincidencia de clases y alta confiabilidad) que esas observaciones no presentan problemas de densidad. Resta saber qué ocurre con el 37% restante de las observaciones debido a que, haya coincidencias de clases entre lo asignado por el modelo y lo asignado manualmente o no, la confiabilidad es baja y por lo tanto no se puede asegurar que el modelo o la clasificación manual sean correctos (probablemente si coinciden se tiene un grado más de seguridad, pero sin justificación formal). Con el umbral establecido en 0.75 se observa que solo el 33% de las observaciones tienen confiabilidad suficiente, de los cuales el 5% muestra claros problemas de densidad y el 28% restante parece no presentar ningún problema.

5.2 - Análisis de la vigencia, frescura y volatilidad de un conjunto de datos aplicando técnicas de minería de datos

5.2.1 - Descripción del procedimiento

La propuesta que se describe a continuación es una variante de la que fuera presentada en los ER Workshops realizados en el marco de la conferencia anual **Quality of Models and Models of Quality**, desarrollada en octubre de 2015 en Estocolmo, Suecia, y publicados en [140]. El método propuesto es aplicable a datos evolutivos, es decir datos para los cuales se tiene información que es periódicamente actualizada manteniendo el **historial** de cambios; en términos de bases de datos relacionales esto significa que cada vez que se debe actualizar los datos de una cierta entidad en lugar de hacer UPDATE sobre la tupla existente se debe hacer INSERT de una tupla nueva incluyendo el momento exacto (timestamp) en que fue producida, dejando la anterior tupla incambiada, generando así un conjunto de tuplas que describen la evolución de la entidad. El registro del momento en que un dato fue creado o modificado es condición necesaria para determinar si dicho dato puede ser actual o está obsoleto; además se requiere que se mantenga el historial de todas las entidades para determinar la velocidad de cambio (*volatilidad*) y con éste la antigüedad relativa al momento actual (*frescura*) y así la *vigencia* de cada dato.

Antes de proceder es necesario definir algunos conceptos básicos:

- **Objeto bajo estudio**: es cualquier entidad que pueda ser comprendida en base a sus propiedades. Por ejemplo, los países del mundo pueden ser un objeto bajo estudio, y el nombre y la población de cada uno son algunos de sus atributos.
- **Entidad**: corresponde a una instancia específica del objeto bajo estudio y que puede ser individualizado en base a un subconjunto de sus propiedades. Por ejemplo, cada uno de los países del mundo es una entidad, y puede ser identificado por su nombre. En términos relacionales una entidad es un conjunto de tuplas en una tabla.
- **Registro**: corresponde al estado de una entidad en un instante específico, como si fuese una foto de la entidad en ese instante. Los registros pueden ser realizados en diferentes intervalos de tiempo. Por ejemplo, en algunos casos un registro anual podría ser suficiente, en otros podría ser necesario tomarlos varias veces al año, o podría ser necesario hacerlo diariamente, cada hora, o hasta cada pocos minutos. En término relacionales, un registro es una tupla.

La técnica propuesta consiste en construir un modelo de clasificación usando un conjunto de datos de entrenamiento como paso inicial para determinar el **período de validez** de cada una de las propiedades del objeto bajo estudio. El período de validez es un indicador general del tiempo durante el cual puede considerarse que un cierto dato es actual en lo que a la propiedad respecta. Es importante notar que cada propiedad puede tener un período de validez diferente (por ejemplo, el nombre de una persona no suele cambiar nunca, la dirección o el teléfono lo hace cada cierto tiempo, y el salario mensual puede tener mayores variaciones). Para estimar el período de validez de cada entidad respecto de una propiedad determinada se considera la relación entre la suma total del tiempo transcurrido entre sucesivos cambios en el valor de dicha propiedad y la cantidad de cambios encontrados (es importante notar que un nuevo registro no necesariamente implica un cambio si la propiedad considerada no cambia de valor). Una vez que el modelo ha sido construido puede usarse para determinar si el último registro que se tiene para cada entidad es aún vigente o no respecto de la propiedad considerada de la siguiente manera: si el tiempo transcurrido entre la fecha del registro (timestamp) y la fecha actual es menor al período de validez estimado entonces se

puede considerar que dicho registro aún es actual (podría estimarse qué tan actual), mientras que en otro caso es obsoleto.

Nuevamente se hace notar que cada propiedad del objeto bajo estudio puede tener un período de validez diferente al resto, y que por tanto el algoritmo debe aplicarse a cada propiedad por separado. Se resalta también una diferencia entre el concepto de **propiedad** del objeto bajo estudio y el concepto de **atributo**: una propiedad puede agrupar varios atributos, y por tanto, para determinar si hay un cambio en una propiedad se debe determinar si hay algún cambio en cualquiera de los atributos que la modelan (por ejemplo, la dirección de una persona es una sola propiedad que podría estar representada por atributos como calle, número de puerta, esquina, número de apartamento, etc.).

El algoritmo puede esquematizarse de la siguiente manera (se describe una variante del presentado en [140] con algunas mejoras):

1. Tomar una base de datos compuesta por registros históricos de las entidades conocidas del objeto bajo estudio. Estos registros deben presentar dos características fundamentales: deben tener un atributo que identifique la entidad a la cual corresponden (“identificador”), y deben indicar cuándo fueron ingresados (“timestamp”).
2. Seleccionar la propiedad que será utilizada para analizar su actualidad. La propiedad elegida no puede ser ni el identificador de la entidad ni el timestamp.
3. Establecer o seleccionar algunos parámetros iniciales:
    - Una función de distancia, **DF**, que tome dos valores y determine si dichos valores son suficientemente diferentes como para considerar que hay un cambio. Esto permite lidiar con casos en los que los cambios observados son despreciables (por ejemplo, al medir una temperatura un cambio de 3°C es significativo pero podría no serlo uno de 0.1°C).
    - El tiempo mínimo, **MT**, que debe mediar entre el momento de generación de un registro respecto del registro anterior para que el mismo sea considerado como un nuevo registro. Esto permite ignorar correcciones puntuales o guardados parciales, que suelen ser frecuentes cuando se ingresan datos. Por ejemplo, durante el ingreso de los datos se registra mal el nombre de la calle en la que reside una persona, se nota el error justo después de generar el registro, se corrige y se vuelve a guardar generando otro registro; en este caso no es que realmente la persona haya cambiado de dirección sino que se hizo una corrección en los datos.
    - La cantidad mínima de registros, **MR**, que deben observarse para cada entidad de forma de aceptar los resultados determinados para ella. Esto es necesario ya que aceptar como válidos los resultados obtenidos con pocos registros podría conducir a conclusiones incorrectas.
4. Inicializar una estructura de datos llamada **CST** (*current state table*, tabla de estado actual) compuesta por los atributos que se describen a continuación:
    - Identificador de la entidad (**EntitiyId, EID**).
    - Fecha del último registro conocido de la entidad (**EntityTimestamp, ETS**).
    - Último valor conocido de la entidad (**EntityCurrentValue, ECV**).
    - Último valor aceptado de la entidad (**EntityLastAcceptedValue, ELV**).
    - Tiempo total entre cambios para la entidad (**EntityTT, ETT**).

- Cantidad total de cambios aceptados para la entidad (**EntityCC, ECC**).
- Cantidad de observaciones de la entidad (**EntityOC, EOC**).

5. Recorrer los registros de la base de datos completa, en orden según la fecha del registro (no es relevante si se utiliza el orden ascendente o descendente si se utilizan valores absolutos) y para cada registro *R* compuesto por el identificador de la entidad (*EntityId*), el valor de la propiedad considerada (*Value*) y la fecha de creación (*Timestamp*) aplicar el siguiente algoritmo:
   - [*Caso 1: entidad nueva*] Si el registro *R* corresponde a una entidad no observada anteriormente (no se encuentra ninguna entrada en la tabla *CST* cuyo *EntityId* sea el observado en *R*) entonces añadir una nueva entrada en la tabla *CST* de la siguiente manera: *CST.EntityId=R.EntityId*, *CST.EntityTimestamp=R.Timestamp*, *CST.EntityCurrentValue=R.Value*, *CST.EntityLastAcceptedValue=R.Value*, *CST.EntityTT*=0, *CST.EntityCC*=0, *CST.EntityOC*=1. Notar que los campos *EntityCurrentValue* y *EntityLastAcceptedValue* son iguales en este caso, y que dado que aún no se registra ningún cambio real entonces los campos *EntityTT* y *EntityCC* se inicializan en 0.
   - [*Caso 2: cambios no aceptados*] Sino, si el valor del registro (*R.value*) no es suficientemente diferente del último valor conocido (*CST.EntityCurrentValue*) ni del último valor aceptado de la entidad (*CST.EntityLastAcceptedValue*) según la función *DF*, o el tiempo transcurrido desde el registro correspondiente al último valor aceptado no es mayor al mínimo establecido (*MT*), entonces actualizar la entrada en la tabla *CST* para la entidad correspondiente de la siguiente manera: *CST.EntityCurrentValue=R.Value* (el valor de la entidad ha cambiado pero no el aceptado, al igual que tampoco cambia la fecha). Notar que no se modifican los campos *EntityLastAcceptedValue*, *EntityTT*, *EntityCC* ni *EntityTimestamp*.
   - [*Caso 3: cambios aceptados*] Sino, actualizar la entrada en la tabla *CST* para la entidad correspondiente de la siguiente manera: *CST.EntityCurrentValue=R.Value*, *CST.EntityLastAcceptedValue=R.Value*, *CST.EntityTT=CST.EntityTT*+abs(*R.Timestamp–CST.EntityTimestamp*), *CST.EntityCC=CST.EntityCC*+1, *CST.EntityOC=CST.EntityOC*+1, *CST.EntityTimestamp=R.Timestamp*.
6. Al terminar, recorrer la tabla CST y evaluar la actualidad de cada una de las entidades E de la siguiente manera:
   - [*Caso 1: insuficientes muestras para la entidad*] Si $E.EntityOC \leq MR$ entonces se tienen pocos registros para la entidad y no se puede determinar el período de validez de la entidad.
   - [*Caso 2: la entidad no cambió nunca de valor*] Sino, si *E.EntityCC*=0 entonces la entidad nunca cambió desde el primer valor conocido por lo que hay grandes posibilidades de que el único valor conocido será siempre el actual. En este caso la entidad es llamada **invariante**.
   - [*Caso 3: la entidad cambió repetidamente de valor*] Sino, estimar el período de validez de la entidad, ***VP*** (validity period), según el siguiente cálculo: *VP(E)=E.EntityTT/E.EntityCC.*

     En este caso también se puede estimar la probabilidad de que el último registro *R* conocido para la entidad *E* sea aún actual, ***PoC*** (probability of currency), de la siguiente

manera: $PoC(R)=1–age(R)/VP(E)$, donde age es una función que indica qué tan antiguo es el registro *R* respecto de la fecha actual.

7. Una vez que se ha calculado el período de validez de cada entidad (*VP*) y establecido la probabilidad de actualidad de cada una de ellas se tiene una valiosa información para determinar si los datos disponibles son suficientemente actuales como para tomar decisiones en base a ellos o si en cambio debieran buscarse otros datos más actualizados. Dicha información también es útil para planificar estrategias a los efectos de mantener siempre vigente la base de datos tratando de actualizar cada entidad antes de que su probabilidad de actualidad caiga por debajo de cierto umbral (actuar proactivamente en lugar de reactivamente).
8. Con lo descrito anteriormente aún quedan algunas cuestiones por resolver: para las entidades para las cuales se tienen menos registros que los establecidos ($E.EntityOC \leq MR$) no se puede tomar ninguna decisión, y además diseñar una estrategia de actualización para cada entidad en forma individual sería poco práctico o incluso imposible. Para resolver estos problemas se propone combinar dos técnicas de minería de datos:
    1. En primer lugar se propone utilizar la **agrupación** (clustering) para agrupar las entidades en **grupos de afinidad** (clusters) de forma tal que las entidades "suficientemente similares" entre sí pertenezcan al mismo grupo, utilizando como "objetivo" al período de validez (VP).
    2. Luego emplear la clasificación para establecer un criterio que permita asignar cada entidad al clúster correspondiente (cada clúster se convierte en una clase) en base a los restantes atributos (que sí están disponibles para las entidades).

    De esta forma se resuelven ambos problemas: para el caso de las entidades para las cuales hay pocos registros basta con identificar, en base a los criterios utilizados para la clasificación, el grupo de afinidad al que pertenecen y de esa forma estimar su período de validez como el igual al clustroide (el centro del clúster) del grupo de afinidad asignado.

Como se aclaró anteriormente el algoritmo es aplicable a una propiedad del objeto bajo estudio a la vez. Podría modificarse (ampliando la tabla *CST*) para permitir considerar múltiples propiedades al mismo tiempo, o ejecutarlo en múltiples oportunidades, una para cada propiedad interesante. En cualquier caso una vez determinado el período de validez para múltiples atributos puede estimarse también el período de validez de la entidad (**EVP**, *entity validity period*) aplicando alguna función de agregación, como valor mínimo, valor máximo, promedio, mediana, promedio ponderado, etc.

5.2.2 – Ejemplo de aplicación

A continuación se describe un ejemplo de aplicación de la propuesta, utilizando un conjunto de datos que, si bien son reales, fueron simplificados con el objetivo de que el ejemplo resulte sencillo de comprender.

El hecho de que la población mundial va en aumento no es algo que esté en discusión. Cuando determinadas organizaciones internacionales planifican actividades para ciertas regiones o países es importante conocer la situación de dichos países, incluyendo cómo evoluciona su población. No es lo mismo planificar una campaña para un país con unos pocos millones de habitantes que hacerlo para un país cientos de veces más poblado, y tampoco es lo mismo hacerlo para un país con una población estable que para otros con factores de crecimiento poblacional grandes. Tal vez, para algunos países un censo poblacional realizado hace cinco o diez años aún sea válido, pero hay países que en dicho período han experimentado un crecimiento demográfico enorme. Lo que se

propone en este ejemplo de aplicación es evaluar cómo varía la población de cada país, y así estimar la validez de los censos demográficos (al menos desde el punto de vista de cantidad de población, no de calidad de vida, servicios, educación, etc, que podría hacerse de igual manera si se tuviesen los datos disponibles). La idea es estimar un período de validez de la información demográfica para cada país, y luego agrupar (clusterizar) los países de acuerdo a dicho valor. Así, se tendrán los países agrupados en intervalos de vigencia.

Los datos para este ejemplo fueron tomados del Banco Mundial de Datos [147]. Están compuestos por el registro, para cada país (identificado por su nombre), de la población registrada o estimada para cada año entre 1960 y 2013 (para algunos países hay excepciones que no fueron consideradas). Estos datos fueron complementados con información general (continente y área total) tomados de Wikipedia [148] (por simplicidad se considera que la extensión del país ha sido la misma desde 1960 hasta hoy; es claro que esto no es verdad para todos los países, pero lo que se pretende evaluar en este ejemplo es la evolución de la población, y la extensión solo se utilizará como un atributo clasificador en ausencia de otro más apropiado).

Antes de poder comenzar a aplicar el procedimiento propuesto, es necesario preprocesar los datos para transformarlos del formato original al requerido por el mismo: recordar que se requiere que cada registro a procesar corresponda a un instante en particular (año en este caso). De esta manera, los datos originales se encontraban en el formato mostrado en la Tabla 7.

País	**Continente**	**Área (km^2)**	**Población 1960**	**Población 1961**	**Población 1962**	...	**Población 2013**
Afganistán	Asia	652230	84440	8953544	9141783	...	30551674
Albania	Europa	28748	1608800	1659800	1711319	...	23620
Andorra	Europa	468	13414	14376	15376	...	79218
Angola	África	1246700	4965988	5056688	5150076	...	21471618
…	…	…	…	…	…	…	…

Tabla 7: Datos sin procesar

Luego del preprocesamiento (el cual se realizó mediante un código Java especialmente desarrollado para esto) los datos se presentann en el formato mostrado por la Tabla 8.

País	**Continente**	**Área (km^2)**	**Año**	**Población**
Afganistán	Asia	652230	1960	84440
Afganistán	Asia	652230	1961	8953544
...	...	...	...	...
Afganistán	Asia	652230	2013	23620
Albania	Europa	28748	1960	1608800
Albania	Europa	28748	1961	1659800
...	...	...	...	...

Tabla 8: Datos ajustados al formato requerido por el método propuesto

Se observa que ahora se está en la situación indicada por el paso 1, donde el atributo "*País*" hace las veces de identificador, y el atributo "*Año*" las veces del instante (timestamp). De esta manera, los países del mundo son el objeto bajo estudio, cada país es una entidad, y se tienen muchos registros de cada país (uno para cada año).

Luego, siguiendo el paso 2, se selecciona al atributo "*Población*" como atributo a evaluar. En primer lugar, porque se partió de la base de que se quería evaluar la evolución de la población, y en segundo porque ya se mencionó que la superficie de cada país se mantuvo deliberadamente constante y que el atributo "*Continente*" es inherentemente constante.

Siguiendo con el paso 3, se establecen los tres parámetros configurables del procedimiento:

- Función de distancia (DF): *similitud al 10%*, lo que significa que el algoritmo considerará que dos valores numéricos son distintos si difieren en al menos 10% el uno del otro. Esto implica que, al evaluar dos registros consecutivos para el mismo país, si su población no varió en al menos un 10% se considera que no hay cambios.
- Tiempo mínimo entre registros (MT): 1 año. En este caso este parámetro no aporta mucho ya que los registros corresponden a cada año, por lo que nunca se encontrarán dos registros que disten menos de eso el uno del otro.
- Cantidad mínima de registros (MR): 10. Esto implica que para que pueda estimarse el período de validez de un país deben procesarse al menos 10 registros para él.

Para ejecutar los pasos 4, 5 y 6 se implementó otro código Java que ejecuta las tareas indicadas por el procedimiento propuesto. Este código produce, como salida, un archivo indicando para cada país el tiempo total entre cambios (*EntityTT*), la cantidad de cambios aceptados (*EntityCC*) y el promedio o período de validez (*VP*) como se observa en la Tabla 9.

País	**Continente**	**Área (km^2)**	**EntityTT**	**EntityCC**	**VP**
Afganistán	Asia	652230	1640991600000	11	149181054545 (4 años 266 días)
Albania	Europa	28748	1514761200000	7	216394457142 (6 años 314 días)
Andorra	Europa	468	1356998400000	14	96928457142 (3 años 26 días)
Angola	Africa	1246700	1640991600000	12	136749300000 (4 años 122 días)
...	...	...	...	...	...

Tabla 9: Resultado de la evaluación según el método propuesto

Los resultados obtenidos son útiles para evaluar la actualidad de los datos disponibles. En este momento es importante recordar que el conjunto de datos utilizado alcanza solo hasta el año 2013. Entonces, sería interesante determinar si la población actual (en 2018) de cada país es similar a la registrada en 2013 (en el rango del 10% que fue el establecido al inicio) o si en cambio la demografía para él ha cambiado de una forma de que los datos disponibles son obsoletos. Al año actual, 2018, y en vista de los resultados mostrados en la Tabla 9, se puede deducir que para Albania es posible considerar que la población se ha mantenido (aún faltarían 2 años más para que su población se haya visto incrementada en 10% desde 2013), mientras que lo contrario ocurre con Andorra, donde supuestamente ya en 2016 la población se habría incrementado en 10%; en el medio se encuentran Afganistán y Angola, aunque por poco los datos también son obsoletos.

A modo de curiosidad, dado que no se muestra la tabla completa por razones de espacio, se puede indicar lo siguiente:

- Emiratos Árabes Unidos es el país que más rápido evoluciona su población, aumentando ésta en al menos 10% cada 1¾ años. Qatar y Kuwait lo siguen con 2½ años. Esto indica que cualquier dato que se tenga de 2013 para estos países ya es totalmente obsoleto.
- Bélgica y Reino Unido son los que más lento evolucionan, requiriendo alrededor de 37 años

para incrementar su población en al menos 10%. Al contrario de lo anterior, los datos disponibles para 2013 muy probablemente aún sean suficientemente actuales en 2018 como para tomar decisiones.

- Uruguay incrementa su población en al menos 10% en aproximadamente 16 años. También se puede considerar que los datos de 2013 son actuales en 2018.

El paso 8 introduce dos técnicas propias de la minería de datos, para lo cual se utiliza el software Weka al igual que en el caso del estudio de la completitud del capítulo 5.1.

En primer lugar, se utiliza Weka para agrupar los países en grupos de afinidad, los cuales a su vez luego serán utilizados como clases. El algoritmo de clusterización utilizado es el llamado SimpleKMeans, el cual es una implementación de Weka del algoritmo KMeans [163][173][169]. Este algoritmo, como gran parte de los de su tipo, necesita que se le explicite la cantidad de clústeres buscados; en este caso se estableció dicho valor en 5. Los resultados obtenidos se muestran en la Tabla 10.

	Cluster 0 0	**Cluster 1 1**	**Cluster 2 4**	**Cluster 3 2**	**Cluster 4 3**
Clustroide	3 años 348 días	5 años 356 días	10 años 4 días	15 años 348 días	29 años 230 días
Distribución	38%	32%	15%	11%	4%
Ejemplos	Afganistán, Andorra, Angola, Arabia Saudita, Sudán.	Albania, Australia, Brasil, China, India, Somalia.	Argentina, Canadá, Cuba, Estados Unidos Jamaica.	Dinamarca, España Francia, Japón, Países Bajos, Uruguay.	Alemania, Bélgica, Finlandia, Reino Unido, Suecia.

Tabla 10: Resultado final de la evaluación según el método propuesto

En segundo lugar se utiliza nuevamente Weka para construir un modelo clasificador que permita asignar un país no observado anteriormente, o para el cual no se tuvo suficientes observaciones como para calcular su período de validez. Para esto se utiliza como clase a los grupos de afinidad descubiertos anteriormente, y como clasificadores a los dos atributos que hasta ahora no habían sido empleados: el continente y la superficie. El algoritmo de clasificación seleccionado para la tarea es Random Tree, ya que éste permite observar el árbol de clasificación. Luego de aplicar el algoritmo de clasificación, utilizando la muestra completa como conjunto de entrenamiento y validación a la vez, se obtuvo un 99% de clasificaciones correctas y un valor de kappa igual a 0.9932. De esta forma se comprueba que, si se toma casi cualquier país y se aplica al modelo de clasificación obtenido, se podrá determinar el clúster al cual pertenece, y con eso se podrá lograr un estimado de su período de validez. Por ejemplo, Curaçao y Saint Martin son dos países para los cuales se tienen datos de 1998 en adelante; si ellos no son incluidos en el análisis previo y no se calculase un período de validez para sus datos, de todas formas se podría estimarlo clasificándolos en el grupo de afinidad correspondiente. En el caso de Curaçao , país de América del Norte y con un territorio de 444 km², el modelo de clasificación lo asigna al clúster 1, y por tanto su período de validez puede ser estimado en 5 años y 356 días (casi 6 años); habiendo sido considerado en el procedimiento previo, se obtuvo que su período de validez real, según los datos disponibles, es de 5 años y 23 días, por lo que la estimación hecha mediante el modelo de clasificación es aceptable, y además, el clúster seleccionado, se corresponde con el asignado durante la clusterización. No tan buenos resultados se obtienen con Saint Martin : también de América del Norte, y apenas 34 km²; el modelo de clasificación lo asigna también al clúster 1, y por lo tanto su período de validez se estima en 5 años y 356 días, siendo que su período de validez real es de 4 años y 128 días, lo que debiera

ubicarlo en el clúster 0. De todas formas, con un 99% de precisión y un valor de kappa igual a 0.9932 se puede afirmar que el caso de Saint Martin es una excepción, y que el método propuesto muestra un camino prometedor al momento de evaluar la vigencia, actualidad y frescura de un conjunto de datos.

5.3 - Análisis de la usabilidad de un conjunto de datos utilizando técnicas de minería de datos

La dimensión usabilidad (ver el capítulo 2.5.4) es bastante particular en cuanto a la posibilidad de realizar análisis automáticos de datos para determinar si presentan una buena calidad o no. Esto es debido a que, como se explica en el referido capítulo, es una dimensión totalmente subjetiva y extremadamente dependiente del contexto. Es subjetiva porque solo las personas encargadas de realizar una tarea pueden indicar si los datos disponibles son adecuados para dicha tarea, especialmente en lo que respecta a los factores comprensibilidad, confiabilidad y utilidad, y es dependiente del contexto porque un conjunto de datos puede ser apropiado para una tarea pero no para otras, especialmente desde el punto de vista de la utilidad.

Por los motivos mencionados es muy difícil intentar establecer un mecanismo automático de evaluación de la calidad de un conjunto de datos en cuanto a la dimensión usabilidad. De todas maneras se puede intentar aplicar varias de las cosas descritas a lo largo del documento para esbozar una propuesta respecto del factor utilidad.

La propuesta sería la siguiente:

1. Partir de un conjunto de datos conocidos con el cual se deba trabajar para abordar una tarea particular.
2. Tomar una muestra aleatoria del conjunto anterior de un tamaño suficientemente representativo (por ejemplo 10%).
3. Etiquetar (clasificar) manualmente la muestra anterior, indicando para cada uno de los datos incluidos en ella si puede ser considerado útil o no para la tarea en cuestión. Esta clasificación debería ser hecha por las mismas personas que trabajarán en la tarea.
4. Aplicar alguna técnica de minería de datos, particularmente análisis asociativo o clasificación (considerando como clase la etiqueta asignada en el paso anterior) de forma de intentar encontrar patrones en los datos que permitan identificar cuáles son los datos útiles.
5. Utilizar los criterios determinados en el paso anterior sobre el resto de los datos de partida para obtener el subconjunto de ellos que probablemente resulten útiles. Se debe tener presente que este conjunto de datos puede ser considerado útil a los ojos de las personas que hicieron la clasificación manual inicial, y que para otras personas los criterios pudieran ser diferentes.

Adicionalmente, continuando con la utilidad, se podría intentar reducir la subjetividad de la usabilidad de los datos registrando qué tan seguido es dicho dato usado: los datos que son más consultados seguramente sean más útiles, mientras que los datos que son más frecuentemente insertados, actualizados o borrados pero rara vez consultados seguramente no sean de mucha utilidad [027].

Respecto de los otros factores correspondientes a la usabilidad, es incluso más claro que no es factible realizar un análisis similar: en cuanto a la accesibilidad, o bien los datos están accesibles o no lo están; en cuanto a la comprensibilidad, además de la subjetividad del factor y partiendo de la

base de que todos los datos tienen la misma estructura (atributos, tipos de datos y rangos de valores) no es necesario realizar ningún procedimiento automático para determinar si son comprensibles o no, ya que o bien todos los datos los son o ninguno lo es; lo mismo sucede con la confiabilidad aunque en este caso se incorpora otra característica de los datos que es la fuente de los mismos, pudiendo ser alguna más confiable que la otra pero difícilmente pueda determinarse a partir de los datos; y respecto del volumen, ni siquiera existe la posibilidad de tratar cada dato en forma independiente por lo que no hay análisis automático que pueda hacerse.

Como se mencionó en la introducción del capítulo esta propuesta permanece en etapa de formulación, aunque no se tienen perspectivas de concretarla en un futuro cercano.

6 - Conclusiones y trabajo futuro

6.1 - Conclusiones

El área de la calidad de datos es, como se mostró, muy amplia y que si bien es reconocida como de muy alta relevancia para el funcionamiento y desarrollo organizacional, es también un área que recibe mayor atención desde la academia que desde la industria. Esto parece ser principalmente por el hecho de que hasta hace relativamente poco tiempo los datos no eran considerados bienes activos de las organizaciones, sino un mero soporte para la toma decisiones; mientras los académicos estudiaban y proponían técnicas para el estudio de la calidad de los datos, las organizaciones y sus responsables se enfocaban en generar los datos para luego emplearlos de alguna manera (y normalmente solo emplean una parte reducida de todos los datos que recolectan). No fue hasta recientemente que las organizaciones comprendieron que contar con datos de baja calidad es casi tan malo como no contar con dato alguno.

Pero el análisis de la calidad de datos lejos está de ser un trabajo simple, y las técnicas *ad-hoc* no suelen dar los mejores resultados. En base a esto se han desarrollado técnicas específicas para el análisis de datos. Pero estas técnicas requieren un buen entendimiento del dominio de los datos, y también mucho trabajo por parte de expertos que sean capaces de detectar y corregir las falencias en ellos. Esto marca que es necesario desarrollar técnicas automatizadas de análisis de calidad de datos. Adicionalmente, la concepción tradicional de calidad de datos, y que aún se reconoce fuertemente presente, se enfoca en la correctitud de los datos, obviando otras características que debieran ser consideradas igualmente relevantes como la completitud, la vigencia y la usabilidad.

La minería de datos es otra área, en principio ajena a la calidad de datos, que también ha visto un gran incremento en el interés en tiempos recientes. La minería de datos se propone descubrir información escondida en grandes volúmenes de datos, y en datos de estructura compleja, de forma automatizada. Desde la clasificación de datos hasta la agrupación de los mismos en grupos de afinidad, incluyendo el análisis de asociaciones existentes entre datos aparentemente sin vínculo alguno, la minería de datos ofrece técnicas de análisis automatizado de los datos disponibles.

Dado que el estudio de la calidad de los datos exige un análisis de los mismos, y que la minería de datos ofrece herramientas para el análisis automatizado de datos, se puede concluir que la minería de datos debería poder ser aplicada para el estudio de la calidad de los datos. Esta idea no es en absoluto nueva, y de hecho tiene un nombre, “minería de calidad de datos”, pero sigue siendo infrautilizada, no solo en las organizaciones sino también en la academia. Esto se desprende de la poca cantidad de trabajos existentes.

Este trabajo realiza un estudio profundo de las áreas de investigación Calidad de Datos y Minería de Datos. Luego estudia las propuestas existentes para la aplicación de minería de datos a la evaluación de la calidad de los datos, encontrando que hay muy pocas propuestas con este enfoque. Finalmente, propone una solución para la evaluación de la densidad de los datos (dimensión completitud) y otra para la evaluación de la vigencia de los datos (dimensión frescura), las cuales aplican técnicas de minería de datos. Cada propuesta es aplicada a un conjunto concreto de datos y sus resultados son presentados.

Las soluciones propuestas en esta tesis fueron publicadas en workshops internacionales, uno de ellos especializado en en área de calidad de datos[140] y otro en el área de calidad de modelos

[048].

6.2 - Aportes

Los aportes de este trabajo se resumen en los siguientes puntos:

- Se hace un relevamiento del estado del arte del área de la calidad de datos, incluyendo las diferentes concepciones y los problemas detectados en su abordaje.
- Se hace también un estudio del estado del arte en el área de la minería de datos, incluyendo los problemas que aborda y las técnicas que propone para hacerlo.
- Se presenta el resultado de la investigación hecha sobre los trabajos existentes relacionados a la aplicación de técnicas de minería de datos para el abordaje de la calidad de los datos, presentando las propuestas existentes.
- Finalmente se presentan también tres técnicas novedosas de aplicación de la minería de datos en la calidad de datos, enfocada en dos áreas de la calidad de datos que usualmente no reciben la mayor atención o lo hacen de una manera descontextualizada:
    - En primer lugar se propone el estudio de la densidad de un conjunto de datos ([048]), evaluando para cada dato faltante si es realmente un problema o si por el contrario es correcto no contar con ese dato por las características de otros datos; al mismo tiempo se propone la idea de que el hecho de que un cierto dato no falte (es decir, se conozca un dato) también puede implicar un problema de completitud, ya que podría suceder que ese dato sí debiera estar ausente ya que no se corresponde con la realidad y por tanto sí tiene un efecto negativo en la completitud, indicando que el conjunto total de datos presenta un nivel de completitud mayor del que realmente tiene.
    - En segundo lugar se propone el estudio de la vigencia de un conjunto de datos ([140]), evaluando si cada dato disponible puede considerarse vigente o si por el contrario debiera considerarse obsoleto. La vigencia, y en general los atributos vinculados con la temporalidad, como la frescura y la actualidad (todos estos intrínsecamente relacionados), han sido históricamente ignorados como indicadores de la calidad de los datos, por lo que en este trabajo se pretende traerlos a un primer plano.
    - Se describe una posible alternativa para estudiar la usabilidad de un conjunto de datos. La usabilidad, con todos sus factores de calidad, es la dimensión más dependiente del contexto y subjetiva de las analizados.

6.3 - Limitaciones

Este trabajo presenta un panorama lo más abarcativo posible de las tres áreas abordadas (calidad de datos, minería de datos, y minería de calidad de datos), mostrando todas las visiones y la mayor cantidad de técnicas posibles. Sin embargo, es imposible abarcarlas completamente todas, con todas sus variantes, o llegar a un nivel de profundidad tan hondo como para estudiar al detalle cada algoritmo y comprender cabalmente su funcionamiento. Es claro que pueden quedar técnicas o algoritmos que no fueron incluidos en los respectivos estados del arte, pero se considera que lo estudiado y presentado es suficiente para conformar una idea cabal de cada área, suficiente para establecer un punto de partida en cualquier dirección en la cual se pretenda profundizar.

Si bien se hicieron pruebas experimentales con casi todos los algoritmos de minería de datos

estudiados, por cuestiones de interés y de espacio no se incluyen en este documento ejemplos de ejecución ni los resultados de las mismas. Tampoco se hicieron estudios comparativos de los diferentes algoritmos, en el entendido de que no era un objetivo del trabajo.

Las propuestas nóveles en el área de la minería de calidad de datos solo se presentan con aplicaciones de laboratorio, con conjuntos de datos que si bien fueron tomados como reflejo de la realidad, no son reales. Se intentó conseguir algunos conjuntos de datos reales pero las características de las mismas limitaban las posibilidades (en algunos casos no se podían extraer fuera de la organización, y en otros no se permitía publicar los resultados aún en forma difusa); por otra parte, los conjuntos de datos existentes en la web no presentan las características requeridas para la aplicación de las técnicas propuestas o requerían un preprocesamiento exigente. Además, no se ejecutaron los mismos casos utilizando diferentes algoritmos como forma de comparar resultados y rendimientos. La elección de los algoritmos en cada caso fue arbitraria, en base a resultados de otros autores y experiencias personales, sin otro criterio.

6.4 - Trabajo futuro

Según lo presentado a lo largo del documento, y como surge de las conclusiones presentadas en este mismo capítulo, se propone las siguientes líneas de trabajo:

- Aplicar los métodos propuestos a conjuntos de datos reales, y con gran tamaño. Para esto sería necesario que alguna organización permita acceder a un conjunto de datos real, sobre el cual poder realizar pruebas y además también permita publicar los resultados.
- Hacer estudios comparativos de la aplicación de diferentes algoritmos y técnicas de minería de datos en cada una de las propuestas realizadas. Esto implica, por ejemplo, aplicar diferentes técnicas de clasificación y agrupamiento, incluso con diferentes parámetros. También podría hacerse con datos de diferentes características.
- Realizar propuestas para abordar otras dimensiones de calidad igualmente poco tratadas en la literatura, pero cuya importancia al momento de la utilización de los datos puede ser mayúscula.
- En el área de la minería de datos, sería interesante hacer estudios comparativos de la ejecución de los métodos propuestos aplicando diferentes algoritmos en cada caso para luego obtener conclusiones sobre la aplicabilidad de cada uno sobre diferentes juegos de datos. En las secciones correspondientes ya se mencionaron algunos estudios comparativos, pero también se comentó que éstos no siempre son completos o que los resultados no están debidamente presentados o justificados.

Apéndice: algoritmos de minería de datos con Weka

Aquí se describen algunos de los algoritmos de minería de datos más utilizados en el ámbito académico. El objetivo es conformar una idea global de los algoritmos existentes, analizando su funcionamiento, entradas y salidas. No se pretende hacer un análisis comparativo ni llegar a comprender en profundidad el funcionamiento de cada uno.

Si bien existen múltiples productos de software que implementan este tipo de algoritmos, tales como RapidMiner (https://rapidminer.com), Keel (http://www.keel.es), Orange (http://orange.biolab.si) y Tanagra (http://chirouble.univ-lyon2.fr/~ricco/tanagra/index.html), entre muchos otros, la herramienta elegida para realizar las pruebas es Weka, principalmente por su facilidad de uso, su amplio arsenal de algoritmos disponibles así como conjuntos de datos de prueba, y también por el hecho de estar implementado en el lenguaje de programación Java y proveer un API que permite incorporarlo fácilmente a otras aplicaciones desarrolladas en el mismo lenguaje.

Weka, siglas de *Waikato Environment for Knowledge Analysis* es un software que implementa una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Fue desarrollado y es activamente mantenido desde 1999 por el grupo de Aprendizaje Automático de la Universidad de Waikato, en Nueva Zelanda. Actualmente se encuentra en la versión 3.8 (3.9 en desarrollo) y es distribuido bajo la licencia GNU GPL. Puede obtenerse en forma gratuita desde la siguiente URL: http://www.cs.waikato.ac.nz/ml/weka/. Los algoritmos provistos por Weka pueden ser aplicados directamente sobre un conjunto de datos mediante una interfaz gráfica intuitiva, así como también pueden ser invocados desde programas Java externos mediante su API. Contiene herramientas para realizar el preprocesamiento de los datos, clasificación, regresión, agrupamiento, determinación de reglas de asociación y visualización entre otras. También es adecuado para el desarrollo de nuevos esquemas de aprendizaje automático [149]. La casi totalidad de los algoritmos que ofrece Weka admiten parámetros para ajustar su funcionamiento a las características de los datos, los requerimientos, etc. El significado de estos parámetros para cada algoritmo puede consultarse en la documentación JavaDoc de Weka, en http://weka.sourceforge.net/doc.dev/index.html. Para los experimentos descritos en este apéndice se utilizó la interfaz gráfica de Weka, en particular el módulo Explorer que permite configurar e invocar los algoritmos disponibles a través de una interfaz intuitiva.

El formato de archivo ARFF

Si bien Weka admite tomar los datos a analizar desde archivos CSV (*comma separated values*, valores separados por coma), o directamente de bases de datos mediante la especificación de consultas SQL, también tiene un formato específico denominado ARFF (*attribute-relation file format*, formato de archivo atributo-relación). Este formato de archivo (definido explícitamente para Weka aunque posteriormente fue adoptado por otros sistemas) define cómo se representan los atributos, los tipos de datos y los datos en sí mismos [150].

Un archivo ARFF está compuesto por dos secciones: un encabezado, donde se describen los datos (cantidad de atributos, nombres y tipos de datos) y una sección de datos. En el encabezado se define el nombre del conjunto de datos mediante la directiva "*@RELATION*" y los atributos mediante la directiva "*@ATTRIBUTE*". Para cada atributo se debe proporcionar un nombre y un

tipo de datos; los tipos de datos soportados son *numeric* (para valores numéricos), *string* (para valores de texto), *date* (para fechas), y una lista de valores nominales encerradas entre llaves ('{' y '}'). Luego la directiva "*@DATA*" demarca el comienzo de la sección de datos, debiendo haber una entrada por línea, donde los valores deben ser separados con comas, y debe haber exactamente la misma cantidad de valores que atributos declarados, siendo la cadena vacía un valor válido y pudiendo utilizar el valor especial '?' para indicar que el valor real es desconocido. La Figura 9 muestra un ejemplo de un archivo ARFF correspondiente al juego de datos iris dataset provisto por Weka. Las líneas que comienzan con el carácter '%' son comentarios.

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988

@RELATION iris

@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth REAL
@ATTRIBUTE class  {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
```

Figura 9: Ejemplo de archivo ARFF

Algoritmos de análisis asociativo

A continuación se describen algunos de los muchos algoritmos que existen en el campo de la minería de datos. Los aquí representados son solo una selección arbitraria y casual, sin ninguna consideración en cuanto al uso, rendimiento o complejidad.

Algoritmo A-priori

El algoritmo A-**Priori** es considerado el iniciador del área de la búsqueda de reglas de asociación. Fue presentado por Rakesh Agrawal y Ramakrishnan Srikant en 1994 en su trabajo titulado "Fast algorithms for mining association rules" (algoritmos rápidos para minería de reglas de asociación) ([088]). El algoritmo se basa en el concepto de **conjuntos de atributos frecuentes** (*frecuent itemsets*), los cuales son conjuntos de atributos que aparecen juntos con una frecuencia inusual, mayor a un cierto umbral llamado **soporte mínimo** [126].

Para comenzar, A-Priori toma como entrada un conjunto de observaciones descritas por exactamente los mismos atributos nominales. La Tabla 11 muestra un ejemplo de un conjunto de datos de entrada compuesto por observaciones correspondientes a ventas de un supermercado; notar que cada uno de los atributos corresponde a un producto comercializado por el supermercado y el valor en la entrada correspondiente indica si la compra incluye o no dicho artículo, y en el caso del total indica el rango de precios; dado que los atributos deben ser nominales, en este caso se representa el costo total de la compra como 'bajo', 'medio', o 'alto'.

	Pan	Leche	…	Manteca	…	Total
Ticket 1	sí	sí	…	no	…	bajo
Ticket 2	sí	no	…	sí	…	medio

Tabla 11: Ejemplo de entrada para A-priori

El conjunto de observaciones de entrada es escaneado para encontrar conjuntos de atributos frecuentes de tamaño 1 (compuestos por los atributos que tienen el mismo valor en más observaciones que el soporte mínimo establecido), conformando el conjunto F_1. Luego repite iterativamente el siguiente procedimiento [088], basándose en la premisa de que si un conjunto de atributos no es frecuente ninguno de sus superconjuntos (conjuntos que lo contienen) tampoco lo es [151], por lo que para encontrar a los conjuntos frecuentes de atributos no es necesario examinar todas las combinaciones posibles de ellos sino que se puede descartar aquellas que contengan subconjuntos que no sean frecuentes:

1. Genera conjuntos candidatos de atributos, C_{k+1} compuesto por las combinaciones de $k+1$ elementos tomadas a partir de los conjuntos de atributos de k elementos de F_k.
2. Recorre el conjunto de observaciones y calcula el soporte de cada candidato de C_{k+1}.
3. Añade los conjuntos de atributos que superan el soporte mínimo al conjunto resultado F_{k+1}.

Los pasos anteriores se repiten hasta que una iteración no logre generar más conjuntos candidatos. Luego de obtener todos los conjuntos de atributos frecuentes el algoritmo termina por presentar las reglas en la forma x_1 & x_2 & … & $x_a \rightarrow y$. Finalmente, para cada regla construida determina su confianza y solo mantiene aquellas que superen un cierto umbral.

El algoritmo a-priori logra un aceptable rendimiento general aunque tiende a presentar problemas cuando hay demasiados conjuntos de atributos frecuentes, éstos son muy grandes (compuestos por muchos atributos), o se utiliza un soporte mínimo demasiado bajo. Además, generalmente produce una cantidad muy grande de conjuntos de atributos como para que puedan ser examinados manualmente por un analista. Aunque ha sido históricamente muy significativo estos problemas han ocasionado que muchos otros algoritmos se hayan ramificado a partir de él [152].

Algoritmo FP-Growth

Las aproximaciones del estilo A-Priori están basadas en el principio de que si un patrón de largo k no es frecuente entonces ningún patrón de largo $k+1$ puede ser frecuente [088][151]. La idea general es entonces generar patrones de largo $k+1$ a partir de los patrones de largo k y verificar si son frecuentes en el juego de datos. Esta táctica es poco eficiente cuando hay un gran número de conjuntos candidatos; por ejemplo, si hay 10^4 conjuntos frecuentes de largo 1 entonces el algoritmo A-Priori deberá construir más de 10^7 conjuntos candidatos de largo 2, lo que es muy costoso sin importar la técnica que se utilice [152]; más aún, cada uno de esos candidatos debe ser contrastado con el conjunto completo de datos para verificar si realmente es frecuente o no. Tras examinar el algoritmo A-Priori se detecta que el cuello de botella del mismo es la generación de los candidatos y la evaluación de cada uno [153]. El algoritmo **FP-Growth** (*frequent pattern growth*, crecimiento de patrones frecuentes) se propone encontrar todos los conjuntos de atributos frecuentes sin necesidad de generar primero candidatos que luego deban ser contrastados contra la base de datos. Para ello se basa en dos pilares [152][153]: una estructura de datos compacta para representar todo el conjunto de datos original, llamada **fp-tree** (*frequent pattern tree*, árbol de patrones frecuentes), que consiste en un árbol de prefijos con información cuantitativa sobre los patrones donde solo los patrones

frecuentes de tamaño 1 tienen nodos que los representan y el resto de la información sobre los patrones de tamaño mayor está dada por las ramas del árbol y por punteros especiales que vinculan nodos con el mismo nombre, y un algoritmo de minería de datos aplicado sobre la estructura anterior. Este algoritmo de crecimiento de conjuntos de patrones comienza por examinar los patrones de largo 1 y para cada uno de ellos construye una **base de patrones condicionales** compuesta por todos los prefijos que conducen desde la raíz del árbol hasta cada uno de los nodos correspondientes al patrón (atributo), y con dicha base construye un nuevo fp-tree para proceder en forma recursiva sobre éste.

Es importante notar que el algoritmo requiere exactamente de dos pasadas por el juego completo de datos, a diferencia de a-priori que debe recorrerlo múltiples veces en cada iteración. Además el árbol fp-tree es mucho más reducido que la base de datos original y sin embargo contiene toda la información necesaria para recrear todos los conjuntos de atributos frecuentes de la base de datos [152][153]. Según sus autores los estudios de rendimiento muestran que el método es eficiente y escalable tanto para encontrar conjuntos de atributos frecuentes grandes y chicos, y que es cercano a un orden de magnitud más rápido que a-priori, e incluso que otras técnicas más recientes [152] [153].

Algoritmos de clasificación

Varios autores han hecho experimentos con el objetivo de determinar si existe un algoritmo o técnica que sea superior en cualquier situación. Hasta el momento no se ha identificado ninguno que pueda ser utilizado en todos los casos. En [154] los autores aplican un test estadístico llamado *paired student t-test* para evaluar el rendimiento de varios métodos de clasificación (ZeroR, OneR, Naive Bayes, SMO, Decision Table, PART, J48, Random Tree) sobre tres diferentes juegos de datos. Con sus experimentos encontraron que Naive Bayes y SMO figuran en los tres casos como apropiados, mientras que PART y J48 lo hacen en dos casos. En [095] los autores también comparan diferentes algoritmos de clasificación y en forma sorpresiva para algunos juegos de datos el algoritmo OneR fue más eficiente que otros algoritmos más complejos. A diferencia de [154] SMO no tuvo los mejores resultados mientras que Naive Bayes, J48 y PART sí lo hicieron. En [155] se comparan los algoritmos C4.5, Ripper y PART llegando a la conclusión que PART parece mejor cuando se considera el tiempo insumido y también en la cantidad de reglas generadas. En [156] los autores hacen experimentos para comparar varias técnicas basadas en árboles. En [157] los autores también realizan un estudio comparativo de varios algoritmos de clasificación basados en árboles (J48, Random Forest, Random Tree, LMT y Decision Stump) usando Weka pero solo se limitan a desplegar tablas con resultados sin discusión alguna.

Algoritmo ZeroR

	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	50	0	0
Virginica	50	0	0

Tabla 12: Matriz de confusión de ZeroR

El algoritmo **ZeroR** (*zero rule*, cero reglas) es el algoritmo más simple de clasificación que se pueda imaginar (junto, tal vez, a la clasificación aleatoria) ya que se limita a asignar a todas las

observaciones la clase que aparezca en mayor número de oportunidades en el conjunto de datos de entrenamiento. Esto significa que ignora todos los atributos clasificadores y solo considera la clase. Aunque no es un mecanismo realmente útil para la clasificación es utilizado para establecer una base para comparar el rendimiento de otros algoritmos más elaborados [158].

Algoritmo OneR

El algoritmo OneR (*one rule*, una regla) es uno de los clasificadores más sencillos y sin embargo produce resultados muy buenos cuando son aplicados a los datasets estándar utilizados normalmente para evaluar algoritmos [158]. Fue propuesto en 1993 por Robert Holte, de la Universidad de Ottawa, Canadá, como una forma de criticar lo inadecuado de ciertos datasets para evaluar técnicas de clasificación automáticas [159]. El objetivo de Holte no era proponer un nuevo algoritmo de clasificación sino demostrar que los utilizados por sus colegas no eran lo suficientemente buenos. En la actualidad el algoritmo es principalmente utilizado con el objetivo de seleccionar los atributos más relevantes para la clasificación (elegir los selectores) para luego aplicar otro algoritmo más complejo.

El algoritmo OneR intenta determinar cuál es el atributo clasificador que aporta más información y basa su regla de clasificación únicamente en él. El algoritmo básico es la siguiente:

1. Para cada atributo *a*
    1.1. Para cada valor posible *v* del atributo *a*
        1.1.1. Seleccionar todas las observaciones que tengan el valor *v* en el atributo *a*.
        1.1.2. Determinar cuál es la clase *c* que aparece más frecuentemente en el subconjunto anterior.
        1.1.3. Añadir la siguiente cláusula a la regla de clasificación del atributo *a*: [si $a=v \rightarrow c$].
    1.2. Determinar la exactitud de clasificación de la regla del atributo *a* como la cantidad de observaciones del conjunto de prueba correctamente clasificadas por la regla construida para el atributo.
2. Seleccionar el atributo *a* que tenga mayor exactitud.

El algoritmo asume que todos los atributos son discretos, por lo que si no lo son debe aplicarse un proceso de discretización previo. Los valores faltantes son tratados por el algoritmo como un valor en sí mismo [159].

Algoritmo PART

PART es un algoritmo basado en dividir-y-conquistar propuesto inicialmente por Eibe Frank e Ian Witten en 2005 [160]. El objetivo del algoritmo es producir una lista ordenada de reglas, llamada **lista de decisión**, la cual es luego utilizada para aplicar, en orden, a las nuevas observaciones, asignando a cada una de la clase correspondiente a la primera regla coincidente, con una clase por defecto por si ninguna aplica. El nombre PART se debe a que está basado parcialmente en otros algoritmos, combinando técnicas del algoritmo C4.5 (basado en árboles de decisión) y de Ripper (otro algoritmo basado en reglas): toma de Ripper la táctica de construir una regla, quitar las instancias a las cuales aplica dicha regla y luego repetir el procedimiento con las instancias restantes, y toma de C4.5 la idea de construir en cada iteración, con el conjunto de datos actual, un árbol de decisión para así tomar como única regla la rama con mayor cubrimiento (la que abarque al mayor número de instancias) [160]. Para reducir el tiempo de construcción del árbol de

decisión completo para luego solo tomar una rama y descartar el resto PART construye solo un árbol parcial, integrando el proceso de poda a de construcción (a diferencia de C4.5 que aplica la poda sobre el árbol completo) para detectar ramas que de seguro no conducirán a mejores resultados (en [160] se puede ver una descripción detallada del algoritmo). Según sus autores una de las principales ventajas de PART sobre C4.5 y Ripper es que no requiere de la resolución de problemas de optimización global como sí sucede con estos últimos (la resolución de problemas de optimización global es una tarea que siempre es compleja y costosa).

Algoritmo J48

El algoritmo J48 es una implementación de código abierto (*open source*) del algoritmo C4.5 de Ross Quinlan, el cual es a su vez una extensión del famoso algoritmo ID3 del propio Quinlan [092]. El proceso de construcción del árbol de decisión es el siguiente: dado un conjunto de datos de entrenamiento se escoge un método de selección de atributos y un criterio para determinar cuál es la mejor partición en cada caso. Luego, se aplica el siguiente algoritmo, donde *D* es el conjunto de elementos a analizar, y *A* el conjunto de atributos "candidatos" [095]:

1. Crear un nodo raíz *N*.
2. Si todas los elementos de *D* están en la misma clase *c* retornar *N* etiquetado con la clase c ("clasificación por unanimidad").
3. Si la lista de atributos es vacía retornar *N* etiquetado con la clase que aparezca más veces en *D* ("clasificación por mayoría").
4. Aplicar el método de selección de atributo a (*D,A*) de forma de determinar el atributo *a* de *A* que mejor particiona a *D*.
5. Etiquetar *N* con el criterio de partición (compuesto por el atributo *a* y la condición de partición).
6. Si el atributo *a* es discreto y se admite partición múltiple (no se limita a árboles binarios), quitar *a* de *A*.
7. Para cada partición D_j resultante de partir *D* según el criterio establecido y el atributo *a*

   7.1. Si D_j es vacía añadir una hoja al nodo *N* con la clase que aparezca más veces en *D*.

   7.2. Sino añadir al nodo *N* un nodo consistente en el resultado de aplicar recursivamente el algoritmo a D_j y *A*.
8. Retornar *N*.

Algoritmo Random Forest

El algoritmo **Random Forest** (selva aleatoria) consiste en construir una cantidad de árboles en forma aleatoria (considerando en cada uno de ellos un subconjunto aleatorio de los atributos selectores) para luego clasificar cada observación con todos ellos y determinar la clase que más veces se obtiene como resultado. El error de generalización converge a un límite inferior a medida que se incrementa la cantidad de árboles generados [161]. El algoritmo es bastante robusto al ruido, ya que cada árbol utiliza solo un subconjunto de los atributos, y es relativamente insensible al tamaño de los datos ya que utiliza conjuntos aleatorios tomados del conjunto de datos de entrenamiento (observar que hay una doble aleatoriedad). Fue originalmente propuesto en 2001 por Leo Breiman y Adele Cutler, quienes a su vez se inspiraron en el trabajo de Tin Kam Ho en los laboratorios Bell. En la actualidad los términos "*Random Forests*", "RF", "*RandomForests*" y

"*RandomForest*" son marcas registradas de Salford Systems [161][162].

En algoritmo está explicado con detalle en el sitio web oficial de sus autores [162]. La idea básica es la siguiente: dado un conjunto de datos de entrenamiento compuesto por *N* observaciones con *M* atributos cada una, construye K árboles de la siguiente manera:

1. Toma *N* observaciones del conjunto de datos de entrenamiento en forma aleatoria y con repetición (cada observación puede ser seleccionada más de una vez).
2. Toma un conjunto de *m* atributos en forma aleatoria (donde *m* es un parámetro establecido de antemano, con *m* muy inferior a *M*).
3. Construye el árbol a su tamaño máximo posible, sin podarlo.

Una vez construidos los K árboles, cada observación que se desee clasificar es aplicada a todos ellos, cada uno de los cuales clasificará dicha observación de alguna manera dando una clase como resultado. La clase que resulte seleccionada mayor cantidad de veces es la asignada definitivamente a la observación.

Además de su función básica, el algoritmo Random Forest también utiliza algunos mecanismos para lidiar con datos faltantes, para la detección de extraños y para la detección de duplicados. Para esto utiliza el concepto de proximidad, intentando verificar si dos observaciones son suficientemente próximas mediante el siguiente procedimiento:

1. Crea una matriz de ceros de tamaño *NxN*, donde la entrada *[i,j]* indica qué tan relacionadas están las observaciones *i* y *j*.
2. Para cada árbol *k*:

   2.1. Aplica todas las observaciones al árbol *k*.

   2.2. Para aquellas observaciones *i* y *j* que coincidan en la misma clase, incrementa la entrada *[i,j]* de la matriz en 1.
3. Normaliza cada entrada dividiendo su valor entre la cantidad de árboles (*K*). Notar que debe resultar una matriz simétrica con la diagonal principal en 1 (esto es debido a que toda observación siempre coincide consigo misma).
4. Asume que las entradas *[i,j]* cuyo valor sea superior a un cierto umbral preestablecido son similares y actúa en consecuencia:
   - Si dos observaciones son muy similares y a una de ellas le falta algún valor le imputa el valor de la otra (imputación de valores faltantes).
   - Si dos observaciones son demasiado próximas tal vez sean la misma (reconocimiento de duplicados).
   - Si alguna observación tiene valores de proximidad demasiado cercanos a 0 con el resto de las observaciones seguramente se trate de un extraño (detección de extraños).

Algoritmos de agrupación (clustering)

Cuando se pretende comparar dos algoritmos de agrupamiento una cuestión importante es cómo determinar qué significa una buena agrupación, o cuándo una agrupación es mejor que otra, ya que no existe un criterio que sea independiente de los datos como del objetivo de la agrupación. Esto implica que es el analista quien debe proporcionar el criterio para seleccionar el mejor algoritmo

según sus necesidades [163]. Una forma de hacerlo es aplicar los algoritmos a un conjunto de datos previamente etiquetado (con una clase correctamente establecida) y luego determinar si la cantidad de clústeres encontrado por cada algoritmo coincide con la cantidad de etiquetas existentes, y si la agrupación realizada por el algoritmo para cada instancia coincide con la etiqueta correspondiente a dicha instancia. Varios autores han hecho experimentos con el objetivo de comparar los diferentes algoritmos de agrupación existentes. En [163] comparan los algoritmos EM y K-Means utilizando Weka para evaluar la calidad de vinos tintos, encontrando que K-Means logró una mayor efectividad que EM aunque también resultó más lento. En [164] los autores comparan los resultados obtenidos de la aplicación de tres algoritmos, EM, K-Means y Cobweb, sobre la base de datos ISBSG, que reúne datos sobre procesos de desarrollo de software a nivel mundial, llegando a la conclusión de que EM es el que produce los mejores clústeres, seguido de cerca por K-Means y bastante lejos por Cobweb, aunque aclaran que el mal desempeño de Cobweb podría deberse a que las características de los datos no son adecuadas para él. En [165] el autor evalúa el funcionamiento de K-Means, HCA, SOM y EM, encontrando que SOM es más apropiado cuando el número de clústeres es grande, pero que K-Means y EM producen mejores resultados cuando el conjunto de datos es muy grande; además, SOM es más preciso que los otros tres ya que éstos son más sensibles al ruido.

Algoritmo Simple K-Means

El algoritmo Simple K-Means es una implementación propia de Weka del algoritmo K-Means propuesto por J. B. MacQueen en 1967 [166] y que aún hoy es uno de los algoritmos de agrupamiento más utilizados. Está basado en una técnica de particionamiento según la cual particiona las observaciones que componen la muestra en *k* conjuntos disjuntos, donde *k* es el número de clústeres deseado establecido de antemano por el analista. Para ello primero selecciona inicialmente *k* observaciones diferentes y asigna cada una de ellas a uno de los *k* clústeres como su centro, llamado **clustroide**. Aunque existen múltiples propuestas sobre como realizar esa primera selección el mecanismo más habitual es hacerlo en forma completamente aleatoria [167]. A continuación el algoritmo repite iterativamente dos pasos:

1. Asigna (en la iteración inicial) o reasigna (en sucesivas iteraciones) cada una de las observaciones al clúster cuyo clustroide se encuentre más cercano.
2. Luego de cada iteración ajusta el centro de cada clúster como la media (*mean*) de todos los miembros del clúster.

	Clúster 0	Clúster 1	Clúster 2
Clase corespondiente	Iris-versicolor	Iris-setosa	Iris-virginica
Asignaciones correctas	47	50	36
Asignaciones incorrectas	14	0	3

Tabla 13: Relación entre la asignación realizada por el algoritmo y la clasificación real de las observaciones

El proceso iterativo se detiene cuando luego de una iteración no hay cambios en los clústeres o cuando se alcanza un número máximo de iteraciones preestablecido por el analista [166][168]. Un elemento fundamental del algoritmo es la forma de computar la distancia entre las observaciones y el clustroide de cada uno de los clústeres para determinar cuál es el más cercano. Para esto normalmente se utiliza la distancia euclidiana aunque podrían utilizarse otras funciones de distancia, teniendo en cuenta que con algunas de ellas el algoritmo podría no converger; en

particular, Weka permite utilizar además la distancias Manhattan (también llamada distancia de la ciudad o distancia del taxista, [170]), Chebyshev ([171]) y Minkowsky ([172]). Otra cuestión interesante a destacar es que el algoritmo es particularmente sensible a la elección inicial de los clustroides, y por lo tanto tiende solo a encontrar un óptimo local y no global; por esta razón es usual ejecutar el algoritmo muchas veces con diferentes elecciones de los clustroides iniciales y al final seleccionar una de ellas en base a algún criterio [167][173]. Por otra parte también es importante la elección del número de clústeres (*k*); si bien en algunas oportunidades se puede tener una idea de cuántos clústeres se desean, en otras la única alternativa es ejecutar el algoritmo con diferentes valores de *k* y luego utilizar algún criterio para elegir el más apropiado [167]. Notar que el número de clústeres debe ser determinado en forma previa al comienzo de la ejecución del algoritmo (otros algoritmos determinan la cantidad óptima de clústeres durante la ejecución).

Algoritmo EM

El algoritmo EM (*expectation maximization*) [174] está basado en la técnica estadística de máxima verosimilitud (*maximum likelihood*), la cual tiene por objetivo fundamental ajustar un modelo probabilista a un conjunto de datos específico, estimado sus parámetros desconocidos; en otras palabras el método funciona ajustando ciertas variables de un modelo estadístico de forma tal que dicho modelo pueda describir cada observación conocida y estimar otras. Al igual que el algoritmo k-means, EM requiere que se establezcan de antemano la cantidad de clústeres buscados, aunque existe la posibilidad de solicitarle al propio algoritmo que determine la cantidad a partir de los datos de entrenamiento. Básicamente el algoritmo consta de dos etapas que se repiten iterativamente:

1. **Paso E** (*expectation*): estima la probabilidad de que cada una de las observaciones pertenezca a cada uno de los clústeres. Al comenzar, en la iteración *t=0*, el vector de probabilidades puede ser generado aleatoriamente o puede ser determinado mediante la aplicación de otro algoritmo.
2. **Paso M** (*maximization*): ajusta el vector de probabilidades de cada clúster calculando la media del clúster en base a las observaciones que fueron colocadas en él considerando el grado de relevancia de cada una.

Algoritmo Canopy

El algoritmo Canopy fue presentado en el año 2000 por Andrew McCallum, Kamal Nigam y Lyle H. Ungar ([175]) como una respuesta al problema que ellos veían en los otros algoritmos de agrupamiento existentes en el momento: todos eran aplicables bajo alguna cierta restricción que no necesariamente se cumple en la realidad. Según los autores hay tres formas en las que un problema de agrupamiento puede ser impracticable: puede haber demasiadas observaciones para analizar (volumen muy grande de datos), cada observación puede estar compuesta por demasiados atributos (alta dimensionalidad de los datos) o puede haber demasiados clústeres por descubrir. En general todos los algoritmos de agrupamiento funcionan correctamente cuando al menos una de las anteriores dificultades no está presente; sin embargo, afirman, ha habido poco trabajo en el estudio de técnicas de agrupamiento que aborden el tema cuando se presentan los tres problemas a la vez. Para poder realizar el agrupamiento aún en los casos en los cuales las tres dificultades están presentes (o para mejorar el rendimiento en los otros casos) el algoritmo Canopy propone relajar el concepto de distancia permitiendo utilizar una función más sencilla y de resultados aproximados. De esta manera el algoritmo agrupa las observaciones de la muestra en conjuntos solapados llamados **doseles** (*canopies*), de forma tal que cada una de las observaciones pertenece al menos a un dosel pero puede pertenecer a más de uno a la vez. Así las observaciones que pertenecen a un

único dosel son candidatas a conformar un único clúster, quedando en duda aquellas que pertenecen a más de uno a la vez. La idea tras la primera etapa es que si dos observaciones no se encuentran en el mismo dosel entonces están lo suficientemente separados entre sí y por tanto no pueden pertenecer al mismo clúster. Dado que la construcción de los doseles se realiza utilizando medidas aproximadas el cumplimiento de dicha propiedad no puede garantizarse, pero al permitir que los doseles se solapen es posible lograr una confiabilidad elevada de que sí se cumpla [175].

Si bien el algoritmo puede utilizarse como técnica de agrupamiento en sí mismo, por lo general es utilizado como técnica de pre-agrupamiento, es decir, como una forma de pre-agrupar las observaciones en conjuntos reducidos (los doseles) para luego aplicar algún otro algoritmo de agrupamiento más estricto (como K-Means, EM, u otro) pero con la salvedad de que solo es necesario medir distancias entre observaciones que comparten un dosel puesto que aquellas observaciones que no lo hacen no pueden pertenecer al mismo clúster [061]. De esta manera no se mejora (ni empeora) los resultados obtenidos con otros algoritmos de agrupamiento pero sí produce mejoras significativas en cuanto al rendimiento y la escalabilidad [175]. El punto fundamental de este algoritmo es la elección de la función de distancia aproximada (a veces llamada métrica de cercanía), lo cual debe ser hecho por el analista con conocimiento del dominio. Estas funciones de distancia aproximada se suelen definir en base a un subconjunto de los atributos de las observaciones convenientemente seleccionados por el analista [175].

El mecanismo para generar los doseles es el siguiente: dado un conjunto de observaciones *D* y dos medidas *T1* y *T2* tales que *T1>T2* (estas medidas son llamadas umbrales y pueden ser provistas por el analista o determinadas automáticamente a partir de los datos mediante algún tipo de validación cruzada), se aplica el siguiente mecanismo:

1. Seleccionar una observación al azar de *D* y crear un nuevo dosel con dicha observación como su centro.
2. Determinar la distancia aproximada (o cercanía) de todas las observaciones restantes al centro anterior y:
   2.1. Asignar a todas aquellas observaciones cuya distancia al centro del dosel creado en el punto 1 sea menor a *T1* a dicho dosel (estas observaciones pertenecen al dosel actual, pero podrían también pertenecer a otro dosel).
   2.2. Quitar de *D* a aquellas observaciones cuya distancia al centro del dosel sea menor a *T2* (estas observaciones están “demasiado cerca” del centro del dosel como para poder pertenecer también a otro dosel).
3. Si *D* no está vacío, volver al punto 1 (para crear un nuevo dosel); en caso contrario finalizar.

Notar que la elección de *T1* y *T2* es un problema tan complejo como el agrupamiento en sí mismo ya que depende de la función de distancia aproximada elegida y de los valores de los atributos involucrados en el cálculo; algunos autores sugieren utilizar una etapa previa compuesta por la ejecución de otro algoritmo de agrupamiento (como k-means) sobre un subconjunto muy reducido y elegido al azar de las observaciones y luego estimar los valores de *T1* y *T2* en base a las distancias entre los clustroides resultantes [175].

Referencias

[001] - H. Henderson; Encyclopedia of Computer Science and Technology; Ed. Facts On File, Inc., ISBN: 978-0-8160-6382-6, 2009. {1409621987599}

[002] - C. Fox; A. Levitin; T. Redman; The notion of data and its quality dimensions; Information processing and management, Vol. 30, No. 1. págs. 9-19, 1994. {1408901444988}

[003] - B. Otto, Y. W. Lee, I. Caballero; Information and data quality in business networking: a key concept for enterprises in its early stages of development; Electron Markets, Vol. 21, pp. 83–97, 2011. {1538327277939}

[004] - I. Caballero, M. Piattini; Assessment and improvement of information quality through information management process concept; Proceedings of the 5th International Conference on the Quality of Information and Communications Technology, pp. 95-102, 2004. {1538321438043}

[005] - L. Pipino; Y. Lee; R. Wang; Data quality assessment; Communications of the ACM, Vol. 45, No. 4ve p. 211-218, 2002. {1402015639992}

[006] - I. Rafique; P. Lew; M. Qanber Abbasi; Z. Li; Information quality evaluation framework: extending ISO 25012 data quality model; World Academy of Science, Engineering and Technology, Vol. 6, 2012. {1404431738206}

[007] - D. Strong; Y. Lee; R. Wang; 10 potholes in the road of information quality; IEEE Computer, August 1997, 1997. {1405707056728}

[008] - T. Redman; Data: an unfolding quality disaster ; DM Review, August 2004, 2004. {1406594600497}

[009] - M. Wuerstl; Data quality defined; QBase White Paper, 2010. {1405640321004}

[010] - C. Tongchuay; P. Praneetpolgrang; Knowledge quality and quality metrics in knowledge management systems; Fifth International Conference on eLearning for Knowledge-Based Society, Bangkok, Thailand , 2008. {1406854365795}

[011] - D. Ballou; H. Pazer; Modeling data and process quality in multi-input, multi-output information systems; Management Science, Vol. 31, No. 2, p. 150-162, 1985. {1410027969249}

[012] - M. Scannapieco; P. Missier; C. Batini; Data quality at a glance; Datenbank-Spektrum 14: p. 6-14, 2005. {1401563925297}

[013] - D. Strong; Y. Lee; R. Wang; Data quality in context; Communications of the ACM, Vol. 40, No. 5, 1997. {1402015501619}

[014] - G. Kumar Tayi; D. Ballou; Examining data quality; Communications of the ACM, Vol. 41, No. 2, p. 54-57, 1998. {1402015168044}

[015] - N. Alkharboush; A data mining approach to improve the automated quality of data; School of Electrical Engineering and Computer Science, Queensland University of Technology, 2013. {1406393934880}

[016] - M. Scannapieco; T. Catarci; Data quality under the computer science perspective ; Archivi & Computer 2, p. 1-15, 2002. {1406596340211}

[017] - R. Wang; H. Kon; S. Madnick; Data quality requirements analysis and modeling; Nineth

International Conference of Data Engineering, Vienna, Austria, 1993. {1405735346893}

[018] - L. Colín O.; Las normas ISO 9000:2000 de sistemas de gestión de la calidad; Revista Virtual Pro,10/2007, Sistemas de Gestión de calidad, 2007. {1409420150451}

[019] - S. Farzi; A. Dastjerdi; Data quality measurement using data mining; International Journal of Computer Theory and Engineering, Vol. 2, No. 1, 2010. {1400808617693}

[020] - K. Orr; Data quality and systems theory; Communications of the ACM, Vol. 41 Iss. 2, p. 66-71, 1998. {1408839770084}

[021] - Y. Wand; R. Wang; Anchoring data quality dimensions in ontological foundations; Communications of the ACM, Vol. 39, No. 11, p. 86-95, 1996. {1405640435914}

[022] - R. Wang; H. Kon; Toward total data quality management (TDQM); TDQM-92-02, 1992. {1402197107324}

[023] - S. Vázquez Soler; D. Yankelevich; Quality mining: a data mining based method for data quality evaluation; Proceedings of the 6th International Conference on Information Quality, 2003. {1400809262549}

[024] - R. Wang; M. Reddy; H. Kon; Toward quality data: an attribute based approach; Decision support systems, No. 13, p. 349-372, 1995. {1402015369409}

[025] - V. Peralta; Data freshness and data accuracy: a state of the art; Tech. Rep. TR0613, Instituto de Computacion, Facultad de Ingenieria, Universidad de la Republica, Uruguay, 2006. {1405639210270}

[026] - E. Rahm; H. Hai Do; Data cleaning: problems and current approaches; Bulletin of the Technical Committee on Data Engineering, Vol. 23 No. 4, p. 3-13, 2000. {1404181141968}

[027] - M. Bobrowski; M. Marré; D. Yankelevich; Measuring data quality; Reporte 99-002, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, UBA, 1999. {1405640495756}

[028] - O. Bernabé, I. Caballero, C. Guerra-García, M. Piattini; Sentando las bases para definir un modelo de calidad de datos para los artefactos usados en el proceso de planificación de un proyecto de desarrollo de software; 6th Iberian Conference on Information Systems and Technologies (CISTI) , 2011. {1538324182841}

[029] - J. Merino, I. Caballero, B. Rivas, M. Serrano, M. Piattini; A data quality in use model for big data; Future Generation Computer Systems Vol. 63, pp. 123–130, 2016. {1538322890599}

[030] - I. Caballero, M. Serrano, M. Piattini; A data quality in use model for big data; ER Workshops 2014, LNCS 8823, pp. 65–74, 2014. {1538320234945}

[031] - H. Habermann; Managing data quality; Report of Proceedings Senior Management Seminar, Mombasa, Kenya, p. 51-58, 2009. {1405640697121}

[032] - D. Ballou; S. Madnick; R. Wang; Assuring information quality; Journal of Management Information Systems, Vol. 20, No. 3, p. 9–11, 2004. {1406855358656}

[033] - D. Ballou; G. Kumar Tayi; Enhancing data quality in data warehouses environments; Communications of the ACM, Vol. 42, No. 1, p. 73-78, 1998. {1402016302424}

[034] - T. Redman; The impact of poor data quality on the typical enterprise; Communications of the ACM, Vol. 41, No. 2, p. 79-82, 1998. {1407593279561}

[035] - A. G. Carretero, F. Gualo, I. Caballero, M. Piattini; MAMD 2.0: Environment for data quality processes implantation based on ISO 8000-6X and ISO/IEC 33000; Computer Standards & Interfaces, Vol. 54, pp. 139–151, 2017. {1538323659905}

[036] - S. Madnick; R. Wang; Y. Lee; H. Zhu; Overview and framework for data and information quality research; ACM Journal of Data and Information Quality, Vol. 1, No. 1, Article 2, 2009. {1403146585016}

[037] - B. Otto, Y. W. Lee, I. Caballero; Information and data quality in networked business; Electron Markets, Vol. 21, pp. 79–81, 2011. {1538322350427}

[038] - J. Barateiro; H. Galhardas; A survey of data quality tools; Datenbank-Spektrum 14, p. 15-21, 2005. {1407117453339}

[039] - N. Askham, D. Cook, M. Doyle, H. Fereday, M. Gibson, U. Landbeck, R. Lee, C. Maynard, G. Palmer, J. Schwarzenbach; The six primary dimmensions for data quality assessment: defining data quality dimensions; DAMA UK Working Group on Data Quality Dimensions, 2013. {1405640260695}

[040] - C. Batini, C. Cappiello, C. Francalanci, A. Maurino; Methodologies for data quality assessment and improvement; ACM Computing Surveys, Vol. 41, No. 3, Article 16, 2009. {1402195505523}

[041] - F. Grüning; Data quality mining: employing classifiers for assuring consistent datasets; Proceedings of the 3rd International ICSC Symposium, ITEE-2007, Oldenburg, Germany, 2007. {1401061782114}

[042] - R. Wang, D. Strong; Beyond accuracy: what data quality means to data consumer; Journal of Management Information Systems, Vol. 12, No. 4, p. 5-34, 1996. {1402015553319}

[043] - M. Jarke, M. Jeusfeld, C. Quix, P. Vassiliadis; Architecture and quality in data warehouses: an extended repository approach; Information Systems, Vol. 24, No.3, 1999. {1408843541436}

[044] - M. Bovee, R. Srivastava, B. Mak; A conceptual framework and belief-function approach to assessing overall information quality; Proceedings of the Sixth International Conference on Information Quality, p. 311-328, 2001. {1406595940379}

[045] - Y. Lee, D. Strong, B. Kahn, R. Wang; AIMQ: a methodology for information quality assessment; Information and Management 40, p. 133–146, 2001. {1406603308533}

[046] - D. Barone, F. Stella, C. Battini; Dependency discovery in data quality; CAiSE 2010, LNCS 6051, p. 53–67, 2010. {1410050902622}

[047] - S. Wang, H. Wang; Mining data quality in completeness; Proceedings of the 12th International Conference on Information Quality, MIT, 2007. {1400808877488}

[048] - S. Pío Alvarez, A. Marotta; L. Tansini; Data density assessment using classification techniques; Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, 2017. {1503713468007}

[049] - P. Paygude, P. Devale; Automated data validation testing tool for data migration quality assurance; International Journal of Modern Engineering Research (IJMER), Vol.3, No.1, p. 599-603, 2013. {1406393125991}

[050] - Q. Liu, G. Timms, Y. Shu, D. Smith, A. Terhorst; Provenance-aware automated data quality control; 5th eResearch Australasia Conference, Melbourne, Australia, 2011.

{1406393330295}

[051] - M. Gregory, S. Ramage; Improving data quality as the basis for automated generalisation; Proceedings of the 24th International Cartographic Conference, 2009. {1406394000276}

[052] - R. Wang; A product perspective on total data quality management; Communications of the ACM, Vol. 41, No. 2, 1998. {1402022177702}

[053] - M. Scannapieco, A. Virgillito, C. Marchetti, M. Mecella, R. Baldoni; The DaQuinCIS architecture: a platform for exchanging and improvingdata quality in cooperative information systems; Information Systems 29, p. 551–582, 2004. {1406769096950}

[054] - C. Batini, F. Cabitza, C. Cappiello, C. Francalanci; A comprehensive data quality methodology for web and structured data; International Journal of Innovative Computing and Applications, Vol. 1 No. 3, p. 205-218 , 2008. {1406771095388}

[055] - D. Hand, H. Mannila, P. Smyth; Principles of data mining; The MIT Press, ISBN: 026208290x, 2001. {1404180981288}

[056] - J. Firestone; Data mining and KDD: a shifting mosaic; White Paper No. two, 1997. {1399946819858}

[057] - G. Mariscal, O. Marbán, C. Fernández; A survey of data mining and knowledge discovery process models and methodologies; The Knowledge Engineering Review, Vol. 25, No. 2, p. 137-166, 2010. {1399946745506}

[058] - M. North; Data mining for the masses; Ed. Global Text Project, ISBN: 978-0615684376, 2012. {1398989241877}

[059] - J. Marín; Introducción a data mining; http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/DM2008.html (último acceso: 03/02/2018), 2008. {1398997211248}

[060] - C. Brodley, T. Lane, T. Stough; Knowledge discovery and data mining; American Scientist, Vol. 87, No. 1, p. 54-61, 1999. {1413855748823}

[061] - A. Kumar, Y. Ingle, A. Pande, P. Dhule; Canopy Clustering: A Review on Pre-Clustering Approach to K-Means Clustering; International Journal of Innovations and Advancement in Computer Science, Vol. 3, No. 5, 2014. {1471805124835}

[062] - J. Leskovec, A. Rajaraman, J. Ullman; Mining of massive datasets; http://www.mmds.org (último acceso: 03/02/2018), 2014.

[063] - M. Stonebraker, R. Agrawal, U. Dayal, E. Neuhold, A. Reuter; DBMS research at a crossroads: the Vienna update; Proceedings of the 19th International Conference on Very Large Data Bases, p. 688-692, 1993. {1407458575700}

[064] - E. Gervilla García, R. Jiménez López, J. Montaño Moreno, A. Sesé Abad, B. Cajal Blasco, A. Palmer Pol; La metodología del data mining: una aplicación al consumo de alcohol en adolescentes; Adicciones, Vol. 21 No. 1, p. 65-80, 2009. {1399950992654}

[065] - U. Fayyad; G. Piatetsky-Shapiro; P. Smyth; From data mining to knowledge discovery in databases; AI Magazine, Vol. 17, p. 37-54, 1996. {1399946676637}

[066] - M. Halkidi; Quality assessment and uncertainty handling in data mining process; EDBT PhD 2000 Workshop, Konstanz, Germany, 2000. {1400811710231}

[067] - S. Das, B. Saha; Data quality mining using genetic algorithms; International Journal of

Computer Science and Security, Vol. 3, No. 2, p. 105-112, 2009. {1401061553339}

[068] - K. Sahedani; A survey: fuzzy set theory in data mining; International Journal of Advanced Research in IT and Engineering, Vol. 2, No. 7, 2013. {1416440586345}

[069] - K. Cios, W. Pedrycz, R. Swiniarski, L. Kurgan; Data mining: a knowledge discovery approach; Ed. Springer, ISBN 978-0-387-36795-8, 2007. {1483907831587}

[070] - U. Fayyad, G. Piatetsky-Shapiro, P. Smyth; Knowledge discovery and data mining: towards a unifying framework; Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, p. 82-88, 1996. {1416263757672}

[071] - R. Brachman, T. Anand; The process of knowledge discovery in databases; Advances in knowledge discovery and data mining, p. 37-57, 1996. {1483907367624}

[072] - I. Davidson, A. Grover, A. Satyanarayana. G. Kumar Tayi; A general approach to incorporate data quality matrices into data mining algorithms ; Proceedings of the 10th ACM SIGKDD International conference on Knowledge discovery and data mining, p. 794-798 , 2004. {1400812378066}

[073] - P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth; CRISP-DM 1.0, step-by-step data mining guide; The CRISP-DM consortium, 2000. {1452047651642}

[074] - G. Piatetsky; What main methodology are you using for your analytics, data mining, or data science projects?; https://www.kdnuggets.com/2014/10/new-poll-methodology-analytics-data-mining-data-science.html (último acceso: 03/02/2018), 2014. {1452052738299}

[075] - J. Mc Caffrey; Clasificación y predicción mediante boosting adaptativo; MSDN Magazine, Abril 2013, 2013. {1400986354622}

[076] - tutorialspoint.com; Data mining tutorial; https://www.tutorialspoint.com/data_mining/index.htm (último acceso: 03/02/2018). {1404180847596}

[077] - K. Ali, S. Manganaris, R. Srikant; Partial classification using association rules; Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, p. 115-118, 1997. {1407377202223}

[078] - Z. Reitermanová; Data splitting; WDS'10 Proceedings of Contributed Papers, Part I, p. 31–36, 2010. {1480275604249}

[079] - A. Viera; J. Garrett; Understanding interobserver agreement: the kappa statistic; Family Medicine, May 2005, p. 360-363, 2005. {1462669757671}

[080] - J. Landis, G. Koch; The measurement of observer agreement for categorical data; International Biometric Society Biometrics, Vol. 33, No. 1, p. 159-174, 1977. {1470542228666}

[081] - C. Corso; Aplicación de algoritmos de clasificación supervisada usando Weka; http://www.investigacion.frc.utn.edu.ar/labsis/Publicaciones/congresos_labsis/cynthia/CNIT_2009_Aplicacion_Algoritmos_Weka.pdf (último acceso: 03/02/2018), 2009. {1462653251334}

[082] - A. del Valle Benavides; Curvas ROC; Trabajo Final de Grado en Matemáticas, Departamento de Estadística e Investigación Operativa, Universidad de Sevilla, España, 2017. {1537328435113}

[083] - P. Sewaiwar, K. Kant Verma; Comparative study of various decision tree classification algorithm using Weka; International Journal of Emerging Research in Management and Technology,

Vol. 4, No. 10, 2015. {1462655221589}

[084] - D.M.W Powers; Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation; Journal of Machine Learning Technologies, Volume 2, Issue 1, p p-37-63 , 2011. {1537492934613}

[085] - Wikipedia; Curva ROC; https://es.wikipedia.org/wiki/Curva_ROC (último acceso: 20/09/2018). {1537496459110}

[086] - R. Weber; Data mining en la empresa y en las finanzas utilizando tecnologías inteligentes; Revista Ingeniería de Sistemas, Vol. XIV, No. 1, 2000. {1416413273072}

[087] - J. Hipp, U. Göntzer, G. Nakhaeizadeh; Algorithms for association rule mining: a general survey and comparison; SIGKDD Explorations, Vol. 2, No. 1, p. 58-64, 2003. {1401419195471}

[088] - R. Agrawal, R. Srikant; Fast algorithms for mining association rules ; Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994. {1405639410349}

[089] - R. Agrawal, T. Imielinski, A. Swami; Mining association rules between sets of items in large databases; Proceedings of the 1993 ACM SIGMOD Conference, 1993. {1405639604339}

[090] - R. Espinosa, J. Zubcoff, M. Zorrilla, J. Mazón; Hacia la consideración de aspectos de calidad de datos en procesos de minería: el caso de las técnicas de clasificación; Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos, Vol. 4, No. 1, 2010. {1404429696798}

[091] - Wikipedia; Bonferroni's principle; http://rationalwiki.org/wiki/Bonferroni%27s_principle (último acceso: 03/02/2018). {1414546752897}

[092] - J. Ali, R. Khan, N. Ahmad, I. Maqsood; Random forests and decision trees; International Journal of Computer Science Issues, Vol. 9, No. 5/3, 2012. {1462654064550}

[093] - L. Rokach, O. Maimon; Data Mining and Knowledge Discovery Handbook (cap. 9); Ed. Springer, ISBN: 978-0-387-25465-4, 2009. {1400993925415}

[094] - I. Witten, E. Frank, M. Hall; Data mining: practical machine learning tools and techniques; Ed. Morgan Kaufmann, ISBN: 978-0-12-374856-0, 2011. {1401060958306}

[095] - C. Nasa, Suman; Evaluation of different classification techniques for web data; International Journal of Computer Applications, Vol. 52, No.9, p. 34-40, 2012. {1462653666826}

[096] - I. Ben-Gal; Bayesian networks; Encyclopedia of Statistics in Quality & Reliability, Ed. Wiley & Sons, 2007. {1406488383895}

[097] - J. Marín; Introducción a las redes neuronales aplicadas; http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf (último acceso: 03/02/2018), 2009. {1400956164401}

[098] - J. Mc Caffrey; Profundización en las redes neuronales; MSDN Magazine, Mayo 2012, 2012. {1400950184807}

[099] - W. Mc Culloch, W. Pitts ; A logical calculus of the ideas immanent in nervous activity; Bulletin of mathematical biophisycs, Vol. 5, 1943. {1452459032217}

[100] - M. Nielsen; Neural networks and deep learning; Ed. Determination Press, 2015. {1452221852486}

[101] - J. Mc Caffrey; Propagación inversa en redes neuronales para programadores; MSDN

Magazine, Octubre 2012, 2012. {1400976372171}

[102] - J. Mc Caffrey; Clasificación y predicción con el uso de redes neuronales; MSDN Magazine, Julio 2012, 2012. {1400953980015}

[103] - M. Mitchell; Genetic algorithms: an overview; Complexity, Vol. 1, No. 1, p. 31-39, 1995. {1484516233811}

[104] - N. Upasani, H. Om; Outlier detection: a survey on techniques involving fuzzy and/or neural approaches; IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, Kanpur, India, 2013. {1404589522825}

[105] - J. Sobrino; Difusos (fuzzy set theory); Documentos de Trabajo de la Facultad de Ciencias Económicas y Empresariales, No. 15, 1988. {1416329706693}

[106] - A. Sobrino Cerderiña; Sobre la teoría de conjuntos borrosos y su aplicación en la formalización de la vaguedad de los lenguajes naturales; Agora 4:5, 2009. {1416329872570}

[107] - E. Hüllermeier; Fuzzy sets in machine learning and data mining; Applied Soft Computing, Vol. 11, No. 2, p. 1493–1505, 2011. {1416440424304}

[108] - S. Rissino, G. Lambert-Torres ; Rough set theory: fundamental concepts, principals, data extraction, and applications ; Data Mining and Knowledge Discovery in Real Life Applications, I-Tech, ISBN 978-3-902613-53-0, 2009. {1416439549305}

[109] - Y. Caballero, R. Bello, L. Arco, B. Cárdenas, Y. Márquez, M. García; La teoría de los conjuntos aproximados para el descubrimiento de conocimiento; Dyna, Año 77, No. 162, p. 261 270, 2010. {1416413425119}

[110] - B. Walczak, D. Massart; Rough sets theory; Chemometrics and Intelligent Laboratory Systems, Vol. 47, No. 1, p. 1–16, 1999. {1416439788432}

[111] - Yoav Freund; Robert E. Schapire; A short introduction to boosting; Journal of Japanese Society for Artificial Intelligence, Vol. 14, Nro. 5, pp. 771-780, 1999., 1999. {1400990376320}

[112] - D. Boswell; Introduction to support vector machines; 2002. {1455678809705}

[113] - S. Gunn; Support vector machines for classification and regression; Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, 1998. {1456078762320}

[114] - R. Burbidge, B. Buxton; An introduction to support vector machines for data mining; Proceedings of 12th Conference of Young Operational Research, Nottingham, UK), p. 3–15, 2003. {1456079680729}

[115] - StatSoft R&D department; Electronic statistics textbook; StatSoft, Inc., http://www.statsoft.com/textbook (último acceso: 03/02/2018), 2013. {1456022756347}

[116] - R. Agrawal, G. Psaila; Active data mining; Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining, p. 3-8, 1998. {1407379496032}

[117] - R. Agrawal, G. Psaila, E. Wimmers, M. Zaït; Querying shapes of histories; Proceedings of the 21st VLDB Conference, 1995. {1407460605345}

[118] - D. Isaac, C. Lynnes; Automated data quality assesment in the intelligent archive; White Paper prepared for the Intelligent Data Understanding program, 2003. {1406391517479}

[119] - T.i Dasu, T. Johnson; Hunting of the snark: finding data glitches using data mining

methods; Proceedings of the 1999 Conference on Information Quality, MIT, 1999. {1417883963685}

[120] - U. Grimmer, H. Hinrichs; A methodological approach to data quality management supported by data mining; Proceedings of the 6th International Conference on Information Quality, 2003. {1401240604613}

[121] - J. Hipp, U. Güntzer, U. Grimmer; Data quality mining, making a virtue of necessity; Proceedings of the 6th ACM SIGMOD workshop on research issues in data mining and knowledge discovery, 2001. {1400809147171}

[122] - D. Luebbers, U. Grimmer, M. Jarke; Systematic development of data mining-based data quality tools; Proceedings of the 29th VLDB Conference, 2003. {1401061969579}

[123] - F. Alizamini, M. Pedram, Mo. Alishahi, K. Badie ; Data quality improvement using fuzzy association rules; 2010 International Conference on Electronics and Information Engineering, 2010. {1418090463615}

[124] - J. Maletic, A. Marcus; Data cleansing: beyond integrity analysis; Proceedings of the 2000 Conference on Information Quality, 2000. {1408300154311}

[125] - A. Marcus, J. Maletic; Utilizing association rules for the identification of errors in data; Technical Report CS-00-04, Division of Computer Science, Department of Mathematical Sciences, University of Memphis, 2000. {1418622538448}

[126][151] - S. Hori, H. Taki, T. Wahio, H. Motoda; Applying data mining to a field quality watchdog task; Electrical Engineering in Japan, Vol. 140, No. 2, 2002. {1400809960543}

[127] - F. Grüning; Data quality mining in ontologies for utilities; EnviroInfo 2006: Managing Environmental Knowledge, p. 501-504, 2006. {1401240451124}

[128] - Wikipedia; Common Information Model (computing) (último acceso: 17/03/2018). {1521344419228}

[129] - F. Grüning; Sasquatch – semantical approach of assuring high data quality by applying data mining techniques; 4th European Semantic Web Conference, 2007. {1401419009145}

[130] - F. Chiang, R. Miller; Discovering data quality rules; Proceedings of VLDB 2008, 2008. {1407596693709}

[131] - S. Shahriar, S. Anam; Towards data quality and data mining using constraints in XML; International Journal of Database Theory and Application, Vol. 2, No. 1, 2009. {1404180591007}

[132] - H. Reza Khosravani; Proposing an improved semantic and syntactic data quality mining method using clustering and fuzzy techniques; International Journal of Applied Information Systems, Vol. 3, No. 3, 2012. {1401242105395}

[133] - T. Diallo, J. Petit, S. Servigne; Discovering editing rules for data cleaning; Proceedings of the 9th International Workshop on Quality in Databases, 2012. {1407596798125}

[134] - D. Lambert; What use is statistics for massive data?; Lecture Notes-Monograph Series, Vol. 43, Crossing Boundaries: Statistical Essays in Honor of Jack Hall, p. 217-228, 2003. {1413855866102}

[135] - A. Karr, A. Sanil, D. Banks; Data quality: a statistical perspective; National Institue of Statistical Sciences Technical Report Number 151, 2005. {1406855139835}

[136] - J. Hipp, M. Müller, J. Hohendorff, F. Naumann ; Rule based measurement of data quality

in nominal data; Proceedings of the 12th International Conference on Information Quality, 2007. {1417566220440}

[137] - Z. Abedjan, L. Golab, F. Naumann; Profiling relational data - a survey; The VLDB Journal, Vol. 24, No. 4, p. 557–581, 2015. {1480466778215}

[138] - Z. Abedjan, L. Golab, F. Naumann; Data profiling: a tutorial; Proceedings of the 2017 ACM SIGMOD Conference, 2017. {1518938562832}

[139] - E. Agichtein, V. Ganti; Mining reference tables for automatic text segmentation; Proceedings of KDD'04, 2004. {1417481831807}

[140] - S. Pío Alvarez, A. Marotta, L. Tansini; Data currency assessment through data mining; Advances in Conceptual Modeling. Lecture Notes in Computer Science, vol 9382, 2015. {1477248318695}

[141] - R. A. Little, D. Rubin; Statistical analysis with missing data; Pub: John Wiley & Sons, ISBN: 978-0-471-18386-0, 2002. {1489889657021}

[142] - N. Horton, K. Kleinman; Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models; The American Statistician, Vol. 61, No. 1, 2007. {1489952208418}

[143] - D. Newman; Missing data: five practical guidelines; Organizational Research Methods, Vol. 17, No. 4, p. 372-411, 2014. {1489953085254}

[144] - V. Sessions, J. Grieves, S. Perrine; A technique for incorporating data missing not at random (MNAR) into bayesian networks; 21st International Conference on Information Quality, Article 12, 2016. {1489884333132}

[145] - T. Little, T. Jorgensen, K. Lang, E. Moore; On the joys of missing data; Journal of Pediatric Psychology, Vol. 39, No. 2, p. 151–162, 2014. {1489952656817}

[146] - Y. Bengio, A. Courville, P. Vincent; Unsupervised feature learning and deep learning: a review and new perspectives; Computer Research Repository - arXiv:1206.5538v3, 2014. {1489901202910}

[147] - The World Data Bank; Population, total; http://data.worldbank.org/indicator/SP.POP.TOTL (último acceso: 03/02/2018), 2014. {1424047957787}

[148] - Wikipedia; Anexo: Países por superficie; https://es.wikipedia.org/wiki/Anexo:Pa%C3%ADses_por_superficie (último acceso: 03/02/2018). {1516327424813}

[149] - Machine Learning Group at the University of Waikato; Weka 3: data mining software in Java; http://www.cs.waikato.ac.nz/~ml/weka/ (último acceso: 03/02/2018), 2015. {1424052800816}

[150] - Machine Learning Group at the University of Waikato; ARFF (book version); http://weka.wikispaces.com/ARFF+%28book+version%29 (último acceso: 03/02/2018), 2016. {1462683225199}

[151] - P. Tanna, Y. Ghodasara; Using apriori with Weka for frequent pattern mining; International Journal of Engineering Trends and Technolog, Vol. 12, No. 3, 2014. {1462730543964}

[152] - J. Han, J. Pei, Y. Yin, R. Mao; Mining frequent patterns without candidate generation: a frequent-pattern tree approach; Data Mining and Knowledge Discovery, Vol. 8, p. 53–87, 2004.

{1470600963721}

[153] - J. Han, J. Pei, Y. Yin; Mining frequent patterns without candidate generation; Proceedings of the 2000 ACM SIGMOD international conference on Management of data, p. 1-12, 2000. {1470600684351}

[154] - R.Sujatha, D.Ezhilmaran; Evaluation of classifiers to enhance model selection; International Journal of Computer Science and Engineering Technology, Vol. 4, No. 1, 2013. {1462653491777}

[155] - S.Vijayarani, M.Divya; An efficient algorithm for generating classification rules; International Journal of Computer Science and Technology, Vol. 2, No. 4, 2011. {1462654900742}

[156] - Y. Zhao, Y. Zhang; Comparison of decision tree methods for finding active objects; Advances in Space Research, Vol. 41, No. 12, p. 1955-1959, 2007. {1462654718362}

[157] - P. Sewaiwar, K. Kant Verma; Comparative study of various decision tree classification algorithm using Weka; International Journal of Emerging Research in Management and Technology, Vol. 4, No. 10, 2015. {1462655221589}

[158] - S. Sayad; An introduction to data science; http://www.saedsayad.com (último acceso: 03/02/2018), 2016. {1462656045418}

[159] - C. Nevill-Manning, G. Holmes, I. Witten; The development of Holte's 1R classifier; Proceedings of the 2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems, p. 239 , 1995. {1462652909004}

[160] - E. Frank, I. Witten; Generating accurate rule sets without global optimization; Working paper series, Department of Computer Science, The University of Waikatu, New Zealand, 1998. {1462654464597}

[161] - S. Rameshpant Kalmegh; Comparative analysis of Weka data mining algorithm RandomForest, RandomTree and LADTree for classification of indigenous news data; International Journal of Emerging Technology and Advanced Engineering, Vol. 5, No. 1, 2015. {1462655407339}

[162] - L. Breiman, A. Cutler; Random forests; https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm (último acceso: 03/02/2018), 2004. {1462655549985}

[163] - Y. Jung, M. Kang, J. Heo; Clustering performance comparison using K -means and expectation maximization algorithms; Biotechnology and Biotechnological Equipment 2014, Vol. 28, No. S1, p. 44-48, 2014. {1472003941042}

[164] - M. Garre, J. Cuadrado, M. Sicilia; Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software; Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 3, No. 1, p. 6-22, 2007. {1473019365850}

[165] - O. Abu Abbas; Comparisons between data clustering algorithms; The International Arab Journal of Information Technology, Vol. 5, No. 3, 2008. {1473024669737}

[166] - J. MacQueen; Some methods for classification and Analysis of multivariate observations; Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967. {1471798315436}

[167] - X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. Mc Lachlan, A. Ng, B. Liu, P. Yu, Z. Zhou, M. Steinbach, D. Hand, D. Steinberg; Top 10 algorithms in data mining;

Knowledge and Information Systems, Vol. 14, p. 1–37, 2007. {1404357675658}

[168] - N. Alldrin, A. Smith, D. Turnbull; Clustering with EM and K-Means; Tech Report, University of San Diego, California, 2003. {1472003680559}

[169] - Machine Learning Group at the University of Waikato; A tutorial on clustering algorithms: K-Means clustering; http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html (último acceso: 03/02/2018), 2003. {1424305925967}

[170] - Wikipedia; Taxicab geometry; https://en.wikipedia.org/wiki/Taxicab_geometry (último acceso: 03/02/2018). {1471802024839}

[171] - Wikipedia; Chebyshev distance; https://en.wikipedia.org/wiki/Chebyshev_distance (último acceso: 03/02/2018). {1471802146558}

[172] - Wikipedia; Minkowski distance; https://en.wikipedia.org/wiki/Minkowski_distance (último acceso: 03/02/2018). {1471802276201}

[173] - R. Patil, S. Deshmukh, K. Rajeswari; Analysis of simple KMeans with multiple dimensions using Weka; International Journal of Computer Applications, Vol. 110, No. 1, 2015. {1471798523310}

[174] – Wikibooks; Data Mining Algorithms In R/Clustering/Expectation Maximization; (último acceso: 22/09/2018). {1537678812072}

[175] - A. Mc Callum, K. Nigam, L. Ungar; Efficient clustering of high-dimensional data sets with application to reference matching; Proceedings of the 6th ACM SIGKDD Internationail Conference on Knowledge Discovery and Data Mining ACM - SIAM Symposium on Discrete Algorithms, 2000. {1471804735271}