Preprocesado y limpieza de datos

(parte de) La curación de datos en la práctica

Máster en Data Science

M1968 - El Ciclo de Vida de los Datos: de la Adquisición a la Presentación (2018-2019)

Diego García Saiz
Grupo de Ingeniería de Software y Tiempo Real
Departamento de Ingeniería Informática y Electrónica

Limpieza de datos (Data Cleaning)

- La limpieza de datos es un proceso de detección y corrección/eliminación de errores e inconsistencias en los datos, con el objetivo de mejorar la calidad de los mismos.
- Los problemas de calidad en los datos los podemos encontrar tanto en colecciones de datos “single-source”, por ejemplo en ficheros o bases de datos (BDs), como en fuentes de datos integradas o múltiples.
- En las colecciones de datos de tipo fichero o BDs, pueden existir problemas de calidad como:
  - Errores ortográficos al introducir los datos.
  - Valores no introducidos o perdidos (Missing values)
  - Valores inválidos
  - Y muchos otros.

Limpieza de datos (Data Cleaning)

- En las fuentes de datos integradas, como los data warehouses o las bases de datos federadas, los problemas son aun más difíciles de detectar y solucionar, por lo que la limpieza de los datos es aun más importante.
  - Un ejemplo de estos problema es la posible existencia de datos redundantes con diferentes representaciones.
  - En estos casos, es necesario detectar esta información redundante y eliminarla, para garantizar la consistencia de los datos.

- En definitiva, la corrección de los datos es indispensable para su análisis, ya que problemas como datos perdidos o la redundancia pueden llevarnos a conclusiones erróneas.

La limpieza de datos (Data cleaning) en el proceso de curación

- Datos estructurados (relacional/SQL, otros)
- Datos semiestructurados (XML, JSON, otros.)
- Datos no estructurados (texto, binarios, otros)
- Streams de datos
- Series temporales
- Etc.

- Análisis estadístico
- Aprendizaje automático
- Minería de Streams
- Extracción de la información
- Análisis semántico
- Etc.

- **Calidad de los datos**
- Análisis de confianza del emisor
- Anotaciones /meta-data)
- Validación
- **Corrección de errores**
- Mantenimiento
- Etc.

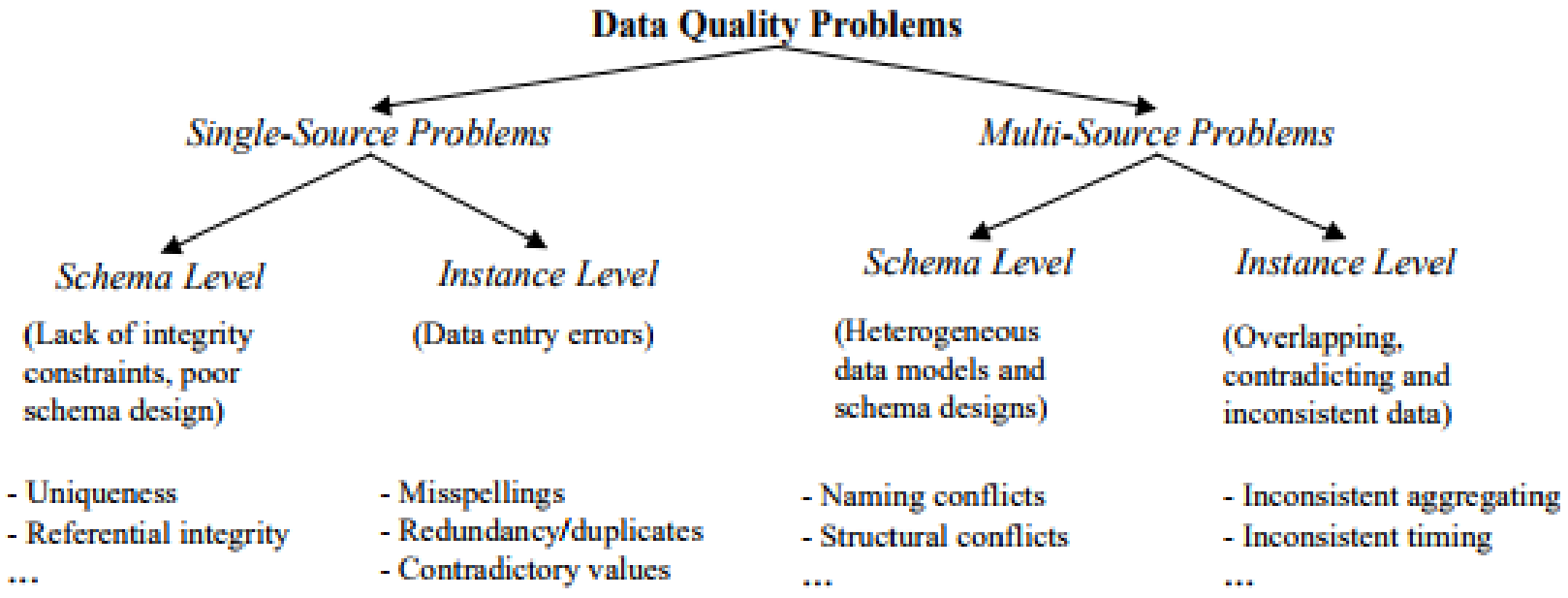
- Sistemas relacionales
- Bases de Datos NoSQL
- Bases de datos NewSQL
- Datos en memoria
- Consistencia de los datos
- Integridad de los datos
- Seguridad
- Privacidad
- Etc.

- Técnicas supervisadas (clasificación, regresión…)
- Técnicas no supervisadas (agrupamiento, reglas…)
- Soporte a la decisión
- Visualización
- Etc

Limpieza de datos: requerimientos

- La finalidad es detectar y corregir o eliminar errores e inconsistencias.
- Hay que limitar la necesidad de inspeccionar los datos de forma manual. En el área del Big Data, se necesitan métodos automáticos o semiautomáticos.
- Tiene que ser un proceso extensible y adaptable de forma que pueda cubrir diferentes fuentes de datos. Esto es, que sea flexible, pero también tenga en cuenta el dominio.
- No se debe de realizar de forma asilada, sino en conjunción con otros procesos.
- La transformación de los datos debe de ser especificada de forma tal que sea fácilmente reproducible e incluso reutilizable en otras fuentes de datos.

Problemas de calidad de los datos



Problemas “single-source”

- Las colecciones de datos “single-source” pueden encuadrarse dentro de dos grandes grupos:
  - Fuentes sin esquema (ficheros), con pocas o ninguna restricción sobre como deben de almacenarse los datos. La probabilidad por tanto de entrar errores e inconsistencias es notablemente alta.
  - Sistemas de bases de datos basados en esquemas (schema-related), que fuerzan a que se cumplan las restricciones definidas sobre los datos, como son las unicidad, el tipo de datos, la integridad referencial…
    - En estos sistemas, los errores e inconsistencias que se suelen encontrar son diferentes a los de las fuentes de datos sin esquemas.
    - Muchos problemas en los datos que siguen un esquema ocurren debido a errores en la definición de los modelos de datos o a una mala aplicación de las reglas de integridad (restricciones).
    - También pueden existir errores e inconsistencias a nivel de instancias (p.e. valores de las filas en los sistemas relacionales), como los errores ortográficos, que no pueden ser prevenidos a nivel de esquema.

Problemas “single-source”

- **Valores no permitidos:** descuento = 890%
- **Incoherencias entre atributos**: edad = 89, fechaNacimiento = ‘1999-01-02’. Ó ciudad = ‘Santander’ y cp = ‘98765’
- **Violación de la unicidad**: Dos personas con el mismo DNI
- **Violación de la integridad referencia**: Diego pertenece al Departamento de Teología de los Datos (el departamento no existe)
- **Valores perdidos**: teléfono=000-000-000 or edad=NULL.

Problemas “single-source”

- **Errores ortográficos**: ciudad=‘Astander’.
- **Valores “críptico”**: categoría=‘B’, rol=‘prof pr’
- **Valores embebidos (multiples valores en uno)**: nombre=John Smith Santander 20/08/1978’, dirección = ‘Calle General Dávila 123, Santander, Cantabria, 39003’
- **Almacenamiento de valores incorrectos**: ciudad = ‘Francia’

Problemas “single-source”

- **Valores duplicados**: almacenar la misma asignatura dos veces con diferentes códigos.

- **Referencias erróneas**: Diego da clases en la asignatura de Economía del Grado de Derecho (va a ser que no)

- **Detección de outiers**: un sensor que mide una saturación de CO2 del 99% en un autobús municipal… (real story)

Problemas “multiple-sources”

- Los problemas encontrados en datos “single-source” pueden verse agravados cuando trabajamos con fuentes de datos múltiples, como son los data warehouses.
  - **Diferencias entre los modelos de datos y los esquemas**: han de ser abordadas en el proceso de transformación e integración.
  - **Conflictos estructurales y de nombres**: mismo nombre para diferentes objetos (tabla categorías en dos bases de datos diferentes con significados diferentes) o, viceversa, diferentes nombres para un mismo objeto (usuario y usuarios)
  - **Conflictos estructurales**: diferentes estructuras de datos (tablas, JSON…), diferentes tipos de datos, diferentes reglas de integridad…
  - **Conflictos en la representación**: incluso si hay dos atributos con el mismo significado, los valores almacenados pueden diferir. Por ejemplo, un atributo para almacenar la fecha de nacimiento puede hacerlo en diferentes formatos en cada base de datos a integrar.
  - **Pueden existir datos solapados**: hay que evitar la redundancia y duplicados.

Problemas “multiple-sources”

Customer (source 1)

CID	*Name*	*Street*	*City*	*Sex*
11	Kristen Smith	2 Hurley Pl	South Fork, MN 48503	0
24	Christian Smith	Hurley St 2	S Fork MN	1

Client (source 2)

Cno	*LastName*	*FirstName*	*Gender*	*Address*	*Phone/Fax*
24	Smith	Christoph	M	23 Harley St, Chicago IL, 60633-2394	333-222-6542 / 333-222-6599
493	Smith	Kris L.	F	2 Hurley Place, South Fork MN, 48503-5998	444-555-6666

Customers (integrated target with cleaned data)

No	*LName*	*FName*	*Gender*	*Street*	*City*	*State*	*ZIP*	*Phone*	*Fax*	*CID*	*Cno*
1	Smith	Kristen L.	F	2 Hurley Place	South Fork	MN	48503-5998	444-555-6666		11	493
2	Smith	Christian	M	2 Hurley Place	South Fork	MN	48503-5998			24	
3	Smith	Christoph	M	23 Harley Street	Chicago	IL	60633-2394	333-222-6542	333-222-6599		24

Limpieza de datos: enfoques

- **Análisis de datos**: para detección de errores e inconsistencia que deban ser eliminadas.
- **Definición de reglas de transformación y flujos de trabajo**: depende de la cantidad de fuentes de datos, de su heterogeneidad y de su “suciedad”, un mayor número de pasos para la transformación limpieza de los datos han de ser definidos.
- **Transformación**: ejecución de las reglas de transformación.
- **Verificación**: la eficiencia y efectividad de las transformaciones debe de ser testada y evaluada. Pueden necesitarse varias iteraciones de análisis, transformación y verificación.

Análisis de datos para la curación y limpieza

- Hay varios enfoques en el análisis de datos para la limpieza. Dos de ellos son:

  - “Data profiling”, enfocado en analizar los datos almacenados en atributos individuales. Obtiene información como el tipo de dato, la longitud, el rango de valores, frecuencia, varianza, unicidad, ocurrencia de valores perdidos, etc…

  - Minería de datos, que ayuda a descubrir patrones específicos en grandes conjuntos de datos, como por ejemplo, la relación entre varios atributos. Se suelen utilizar técnicas descriptivas (clustering, otras).

Análisis de datos para la curación y limpieza

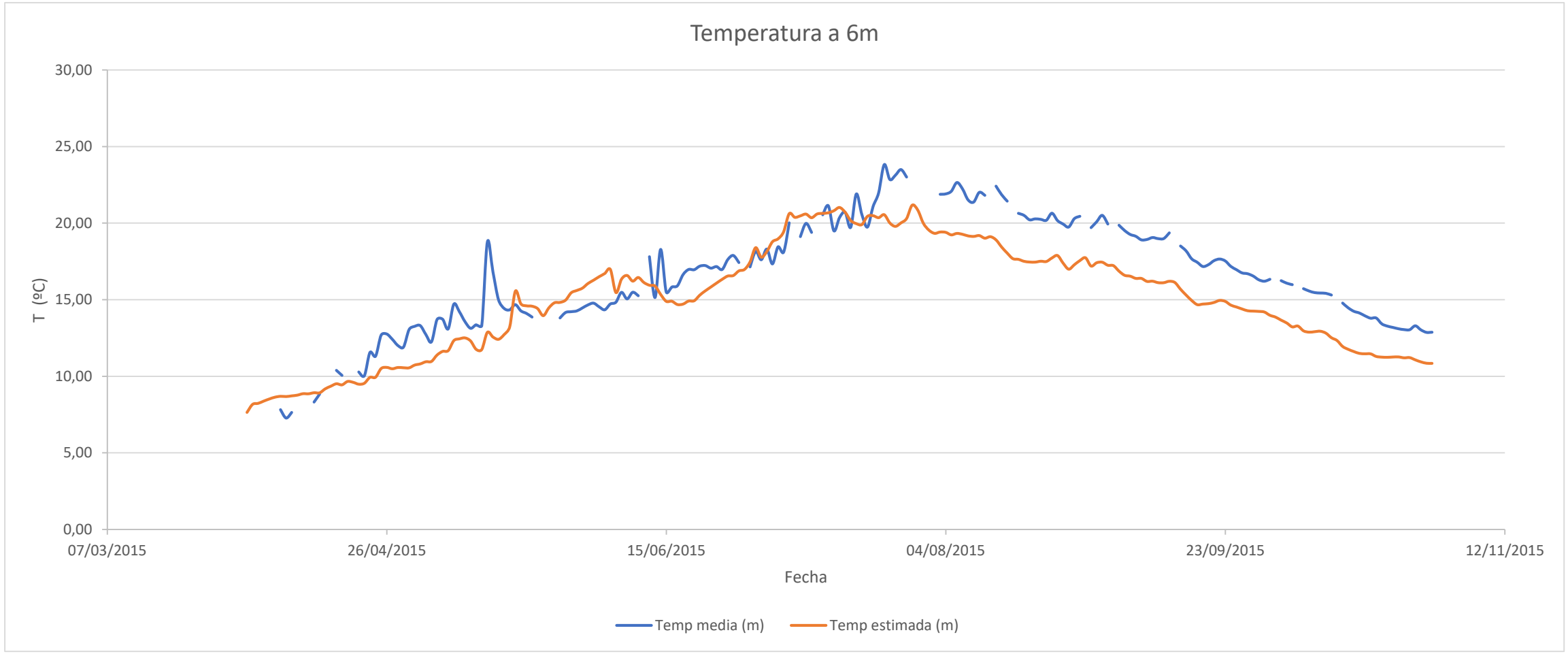
Problems	Metadata	Examples/Heuristics
Illegal values	cardinality	e.g., cardinality (gender) > 2 indicates problem
	max, min	max, min should not be outside of permissible range
	variance, deviation	variance, deviation of statistical values should not be higher than threshold
Misspellings	attribute values	sorting on values often brings misspelled values next to correct values
Missing values	null values	percentage/number of null values
	attribute values + default values	presence of default value may indicate real value is missing
Varying value representations	attribute values	comparing attribute value set of a column of one table against that of a column of another table
Duplicates	cardinality + uniqueness	attribute cardinality = # rows should hold
	attribute values	sorting values by number of occurrences; more than 1 occurrence indicates duplicates

Datos perdidos (missing data): un problema recurrente

- Uno de los problemas que se encuentran con mayor frecuencia es el de los valores perdidos.

- Estos valores perdidos se deben a muy diferentes factores, especialmente cuando son datos recogidos de sensores:
  - Clima extremo
  - Fallo en el equipo
  - Fallo de potencia
  - Señal de comunicación inestable

Datos perdidos (missing data): un problema recurrente



Datos perdidos (missing data): un problema recurrente

- Hay multitud de enfoques y técnicas para lidiar con este problema:
  - Estadísticas (uso de la media o la moda, por ejemplo).
  - Interpolación lineal.
  - Métodos de minería de datos supervisados:
    - Máquinas de soporte vectorial
    - Técnicas de regresión.
  - Matrices probabilísticas
  - Métodos Bayesianos
  - Y muchos otros…

Definición de transformaciones de los datos

- Cuando se manejan simultáneamente datos de diferentes esquemas, es necesario realizar diversos pasos de transformación mapeo.
- Es imprescindible especificar las transformaciones en un lenguaje de programación apropiado, de forma que estas transformaciones y la limpieza de los datos requieran el mínimo de programación de código necesario.
- Varias herramientas ETL ofrecen diferentes mecanismos particulares de transformación de los datos.
- Un enfoque bastante flexible es el uso de un lenguaje estandarizado, con SQL, pata realizar las transformaciones de los datos y usar todas las posibilidades que ofrecen estos lenguajes y sus extensiones.

Resolución de conflictos con múltiples fuentes de datos

- Diferentes tipos de transformaciones han de ser aplicadas sobre las fuentes de datos individuales de cara a prepararlos para la integración con otras fuentes.

- Por ello, además de una posible “traducción” de los esquemas, otros procesos deben de ser ejecutados:

  - Extracción de valores en atributos “libres” (free-form attributes): este tipo de atributos suelen contener valores que deberían estar divididos en varios atributos diferentes. Es muy común en los campos que almacenan nombres o direcciones.

  - Validación y corrección: en este paso, se examinan los valores de cada instancia de la Fuente de datos para detector posibles errores a ese nivel, y corregirlos de inmediato. Para ello, se pueden usar diferentes herramientas, como diccionarios en el caso de los errores gramaticales, o mecanismos de detección de dependencia para evitar la incongruencia entre conjuntos de atributos, como por ejemplo la edad y la fecha de nacimiento.

  - Estandarización: para facilitar la integración, los atributos han de ser convertidos a un formato uniforme y consistente. Por ejemplo, las fechas deben representarse de igual forma, las abreviaturas han de ser resueltas consistentemente (consultando diccionarios de sinónimos, por ejemplo, o aplicando técnicas de machine learning para resolver similitudes), etc.

- Muchas de las tareas de limpieza en múltiples fuentes de datos, como la eliminación de duplicidades, no deben de ser ejecutadas hasta que no se haya realizado una correcta limpieza individual de cada una de las fuentes individuales.