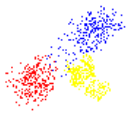
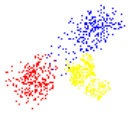


Integración

- La información en la mayoría de los casos debe ser integrada desde múltiples fuentes de datos.

- Algunos problemas típicos:
  - *Identificación de la entidad*
  - *Redundancia*
  - *Detección y Resolución de conflictos entre valores*



Identificación de la entidad (entity identification problem)

- La misma entidad tiene distintos nombres en diferentes fuentes de datos, ej, customer_id, cust_number.

- Para esto se utilizad Metadata donde se almacena información sobre las entidades en cada fuente de datos, ej: nombre, significado, tipo de datos, rango, valores nulos, etc.

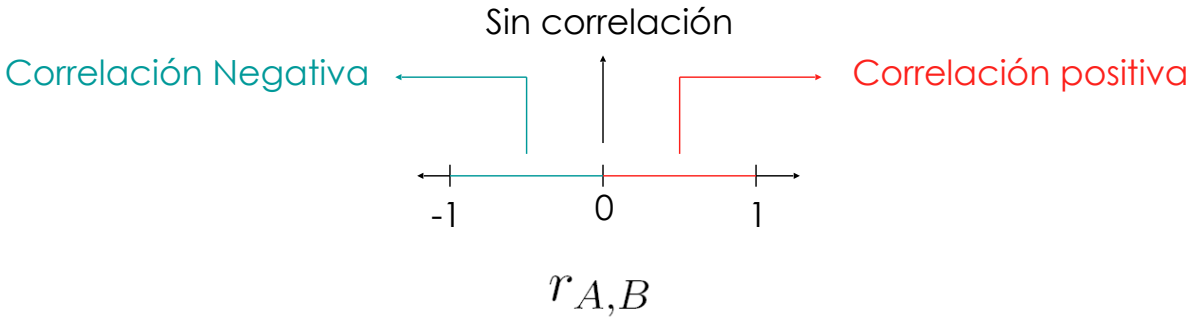
Redundancia

- Un atributo es redundante si puede ser derivado de otro. Errores en la identificación de la entidad suelen llevar a situaciones de redundancia
- Tablas no normalizadas también llevan a redundancias

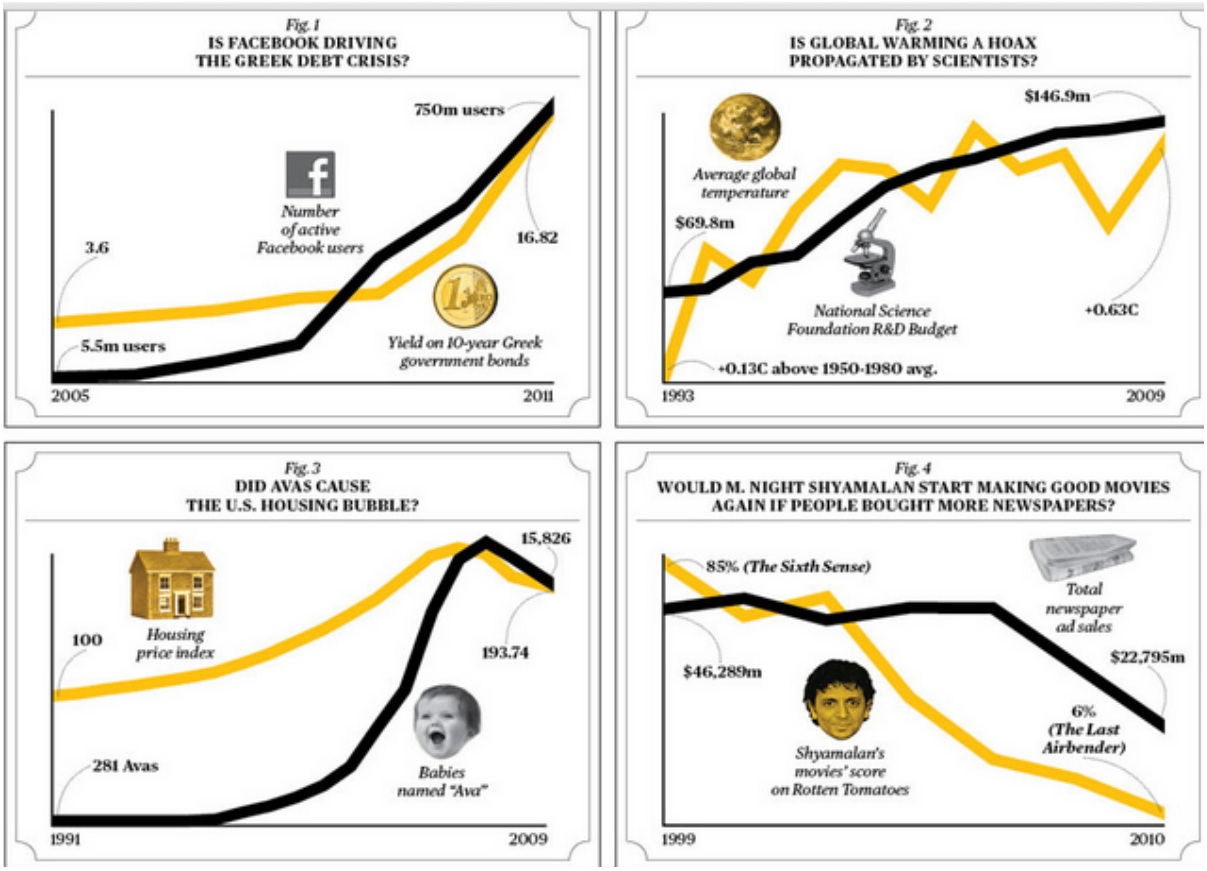
- Puede ser detectada realizando un análisis de correlación (Pearson):

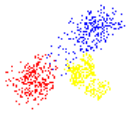
$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \overline{A})(b_i - \overline{B})}{N\sigma_A\sigma_B} \qquad -1 \leq r_{A,B} \leq 1$$

Redundancia (Cont.)



- Notar que correlación no implica causalidad.

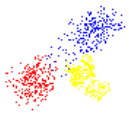




Detección y Resolución de conflictos entre valores

- Para la misma entidad, los valores del atributo proveniente de distintas fuentes de datos es diferente.

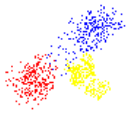
- Problema causado por diferencias de representación, escala, codificación, etc.



Detección y Resolución de conflictos entre valores (Cont..)

- **Diferencias en representación:** Por ejemplo, para una cadena de hoteles, el precio de una habitación en un hotel en distintas ciudades puede representar conceptos distintos, p.ej, uno incluye desayuno, impuestos u otro tipo de servicio que el precio en otra ciudad no contempla.

- **Diferencias en niveles de abstracción:** El total de ventas puede significar ventas totales en la cadena completa en una base de datos y en otra puede ser ventas totales en el hotel.

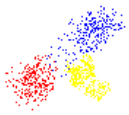


Detección y Resolución de conflictos entre valores (Cont..)

- **Diferencias de codificación:** Los valores para una entidad se nombran distintos en distintas fuentes de datos. Ej. Sexo ⟶ M,F ó 1,2 , etc.

- **Diferencias de escala:** Medidas en una base de datos pueden estar en centímetros, en otra en metros, etc.

- Al hacer matching entre atributos de diferentes fuentes de datos es necesario tener en cuenta la estructura de la información, ej. En una base de datos el descuento se aplicaba sobre un item y en otra el descuento se aplica sobre el total de la orden



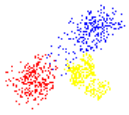
Transformación

- Los datos se transforman y consolidan de tal forma de quedar listos para los procesos de minería de datos.
- La transformación puede envolver los siguientes procesos:

Smoothing: binning, regresión, clustering.

Normalización: Se modifica la escala de los atributos de tal forma que todos los valores queden dentro de un rango específico. Ej, 0 y 1, -1 y 1.

Construcción de características (feature construction): Nuevos atributos son construidos desde el mismo set de datos para mejorar el proceso de data mining.

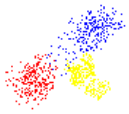


Transformación(Cont..)

Agregación: Roll Up de algunos atributos (ventas mensuales, anuales, etc.)

Generalización: Datos son reemplazados por datos de niveles más altos (jerarquías de concepto). P. Ej, pasar de calles a comunas, ciudades a países, edad a un concepto más alto como “adulto-joven”, “senior”,etc.

TECNICAS PARA TRANFORMAR DATOS

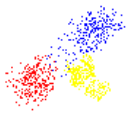


Normalización

Útil en algoritmos que consideran distancias (KNN, clustering, etc.)

Normalización Min-Max: Realiza una transformación lineal en los datos originales. Si los rangos iniciales son *min_A y max_A,* y los rangos finales son *new_min_A* y *new_max_A,* la transformación de un valor v a un valor v' queda:

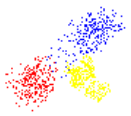
$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$



Normalización(Cont..)

Normalización z-score: Los valores para un atributo A son normalizados en base a la media y la desviación estándar:

$$v' = \frac{v - \overline{A}}{\sigma_A}$$



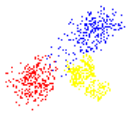
Normalización(Cont..)

Normalización decimal (by decimal scaling)**:** Los valores para un atributo A son normalizados moviendo los puntos decimales:

$$v' = \frac{v}{10^j}$$

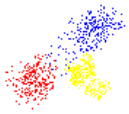
Donde *j* es el menor entero tal que $Max(|v'|) < 1$

Ej: rango inicial -971 a 990 ⟶ -0.971 a 0.990



Reducción de Datos

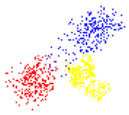
- A veces la cantidad de información hace que sea impracticable procesar la base de datos
- La idea es reducir la base de datos manteniendo la integridad en un alto porcentaje.
- Los algoritmos de minería de datos deben producir resultados muy similares en la base de datos reducida
- Algunas estrategias de reducción son las siguientes:
  - Reducción de dimensionalidad
  - Selección de un subconjunto de atributos
  - Condensación de datos
  - Discretización (Bining, generación de rangos)



Construcción de características

Se construyen nuevos atributos a partir de los existentes de tal forma de ayudar al proceso de data mining. Por ejemplo, se podría agregar el atributo ***área*** a partir de los atributos ***alto*** y ***ancho***, esto puede ayudar a encontrar patrones que se perderían si no se hiciera esta modificación.

En algoritmos de clasificación se construyen inicialmente muchas características y luego en base a procesos de selección se dejan las mejores



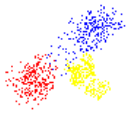
Selección de un subconjunto de atributos

- Muchas veces existen atributos en las bases de datos que para cierto análisis son irrelevantes.

- Los patrones descubiertos sobre una cantidad reducida de atributos son más entendibles por el usuario.

- Se necesitan algoritmos eficientes, para **n** atributos existen $\{2^n-1\}$ posibles subconjuntos de atributos

- Algunos algoritmos conocidos son: Stepwise forward selection, Stepwise backward elimination, Plus R take away R, Exhaustive search, etc.



Stepwise Forward Selection (SFS)

- El proceso comienza con un conjunto vacío de atributos. Se encuentra el mejor de todos los atributos y se agrega al conjunto, en cada iteración se determina el mejor atributo de los que quedan y se va agregando al conjunto. Debe existir un criterio de detención
- Se necesita de una heurística de decisión para decidir cuándo una característica es "mejor" que otra. Ej: discriminante de Fisher

Discriminante de Fisher

- Sean X los datos, con C clases, y definiendo como n_i el numero de elementos de clase i.

$$p_i = \frac{n_i}{\sum_{i=1}^{C} n_i}$$

Probabilidad de clase

$$u_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j^i$$

Media de clase i

$$\hat{u} = \sum_{i=1}^{C} p_i u_i$$

Media global

Discriminante de Fisher

- Este score maximiza la varianza entre clusters y minimiza la varianza dentro de cada cluster.

$$\hat{S}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_j^i - u_i)(x_j^i - u_i)^T$$

Matriz de covarianza

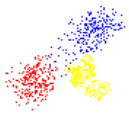
$$S_w = \sum_{i=1}^{C} p_i \hat{S}_i$$

Within-class scatter matrix

$$S_B = \sum_{i=1}^{C} p_i (u_i - \hat{u})(u_i - \hat{u})^T$$

Between-class scatter matrix

$$\boxed{J_F = Trace(S_w^{-1} S_b)}$$



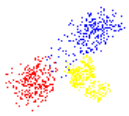
Stepwise Forward Selection (SFS) (pseudocódigo)

1. Inicializar S={NULL}

2. Seleccionar el mejor siguiente feature: $x^+ = \arg\max_{x \notin S_k} [J(S_k + x)]$

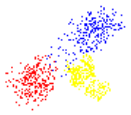
3. Actualizar $S_k = S_k + x^+$, k=k+1

4. Ir a 2, hasta cumplir criterio de término.



Stepwise Backward Selection

- El proceso comienza con todos los atributos en el conjunto inicial, en cada iteración se va eliminando el peor elemento del conjunto.
- También es necesario definir una heurística y un criterio de detención dependiendo del problema a solucionar
- Una versión combinada de SFS y SBE podría ser en cada paso agregar la mejor y eliminar la peor partiendo de algún conjunto inicial que no sea el vacío ni el conjunto de todos los atributos



Stepwise Backward Selection (SBS) (pseudocódigo)

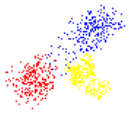
1. Inicializar S={X} (todas)

2. Remover el peor feature:

$$x^{-} = \arg\max_{x \in S_k} \left[J(S_k - x) \right]$$

3. Actualizar $S_k = S_k - x^{-}$, k=k+1

4. Ir a 2, hasta cumplir criterio de término.

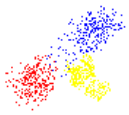


Plus R take away R

- El proceso comienza con un conjunto vacío de atributos, en cada paso se agregan las mejores L características (atributos) y se eliminan las peores R

Exhaustive search

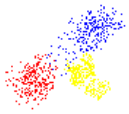
- Se prueban todas las posibles combinaciones de conjuntos de C atributos, se elige la mejor combinación.
- Este método asume a priori el número de características que va a seleccionar



Branch and Bound

- Se recorre un árbol de búsqueda explorando todas las posibles combinaciones de atributos, optimiza su búsqueda podando ramas de combinaciones peores que las que ya se han explorado.

- Este método entrega un óptimo global, asumiendo que la métrica es monotónica.



Branch and Bound (pseudocódigo)

1. Construir un árbol ordenado que satisface:

$$J_1(Z_1) \geq J_2(Z_1, Z_2) \geq .... \geq J_m(Z_1, ..., Z_m)$$

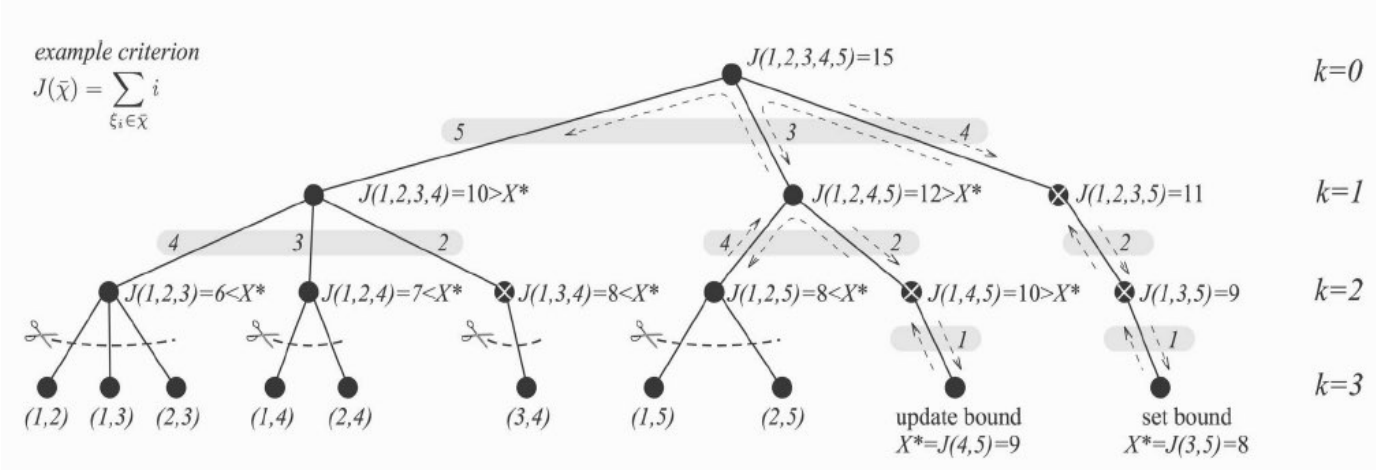
donde J_k significa que k variables son eliminadas.
El orden de Z_p es preferentemente determinado por un criterio de discriminación.

2. Recorrer el árbol de derecha a izquierda en profundidad.

3. Evaluar el criterio en cada nivel y reordenarlo segun corresponda.

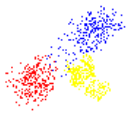
4. Podar el arbol si corresponde: Si un nodo no cumple criterio, entonces sus hijos tampoco lo haran debido a monotonicidad

Branch and Bound



Reducción de Dimensionalidad

- Se aplican transformaciones de los datos de tal forma de obtener una versión comprimida de ellos.
- La versión comprimida tiene bajos porcentajes de pérdida de información.
- Técnica más utilizada: Principal Components Analysis (PCA).



Principal Components Analysis (PCA)

- Método de reducción de dimensionalidad

- Los atributos se transforman en un nuevo conjunto de atributos más reducido, donde cada uno de los atributos del conjunto nuevo es una combinación lineal de los atributos iniciales