DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Integración de datos - Datos

Fernando Berzal, berzal@acm.org

Integración de datos

- Descripción de fuentes de datos
- Integración de esquemas
  - Emparejamiento de esquemas [schema matching]
  - Correspondencias entre esquemas [schema mapping]
  - Gestión de modelos
- Emparejamiento de datos [data matching]
  - Técnicas de emparejamiento
  - Escalabilidad
- Wrappers
- Apéndices:
  - Emparejamiento de cadenas [string matching]
  - Procesamiento de consultas

Emparejamiento de datos

Problema

Encontrar datos estructurados (tuplas, XML, RDF…) que se refieren a la misma entidad del mundo real.

p.ej. (José García, 958 123 123, Granada Spain)
vs. (Pepe G., 123 123, Granada España)

Otro problema fundamental en integración de datos:

- Integración de múltiples bases de datos con el mismo esquema.
- Integración de tuplas provenientes de fuentes de datos con esquemas diferentes.
- Resolución de consultas

Emparejamiento de datos

Formalización del problema

Dadas dos tablas relacionales X e Y con el mismo esquema, asumimos que cada tupla de X e Y describe una entidad del mundo real (p.ej. personas).

Una tupla $x \in X$ casa con una tupla $y \in Y$ si ambas hacen referencia a la misma entidad en el mundo real.

Cada par (x,y) identificado será un emparejamiento [match].

Objetivo: Encontrar todos los emparejamientos (x,y).

Emparejamiento de datos

EJEMPLO

Table X

	Name	Phone	City	State
X_1	Dave Smith	(608) 395 9462	Madison	WI
X_2	Joe Wilson	(408) 123 4265	San Jose	CA
X_3	Dan Smith	(608) 256 1212	Middleton	WI

Table Y

	Name	Phone	City	State
y_1	David D. Smith	395 9426	Madison	WI
y_2	Daniel W. Smith	256 1212	Madison	WI

Matches

(x_1, y_1)
(x_3, y_2)

Variantes del problema

- Tablas X e Y con esquemas diferentes.
- Emparejamiento de tuplas de la misma tabla.
- Datos no relacionales (p.ej. XML o RDF).

Emparejamiento de datos

Emparejamiento de datos
≠ Emparejamiento de cadenas

- En teoría, podríamos considerar cada tupla como una cadena (concatenando sus campos) y utilizar técnicas de emparejamiento de cadenas.
- En la práctica, eso haría difícil aprovechar el conocimiento específico que utilizan algunas técnicas más sofisticadas.

  p.ej. Tuplas que describen personas deberían emparejarse cuando coinciden nombre y teléfono (o su e-mail), aunque esto puede resultar difícil de especificar si las tuplas se convierten en cadenas.
  ➔ **Mejor mantenemos los campos separados.**

Emparejamiento de datos

Desafíos prácticos

(los mismos que al emparejar cadenas)

- **Precisión [accuracy]:**
  Variaciones en el formato de los datos, uso de abreviaturas, convenciones diferentes, omisiones y errores en los datos.

- **Escalabilidad [scalability]:**
  Emparejar cada tupla con todas las demás no es práctico, $O(n^2)$, por lo que deberemos reducir el número de comprobaciones necesario.

Emparejamiento de datos

Técnicas

- Emparejamiento basado en reglas.
- Aprendizaje supervisado: Clasificadores.
- Aprendizaje no supervisado: Agrupamiento [clustering].
- Métodos probabilísticos.
- Emparejamiento colectivo.

Uso de reglas

Se elaboran (manualmente) reglas para decidir cuándo dos tuplas corresponden a la misma entidad.

- Se examinan tuplas de conjuntos de datos reales.
- Se evalúan y refinan las reglas sobre el conjunto de datos original (o, mejor, sobre un conjunto de datos de prueba independiente del conjunto utilizado para elaborar las reglas).

Uso de reglas

Combinación lineal

$$sim(x,y) = \sum_{i=1}^{n} \alpha_i \cdot sim_i(x,y)$$

- Se calcula la similitud entre tuplas como una combinación lineal de las similitudes individuales (usando pesos α_i con $\Sigma\ \alpha_i = 1$).
- Habitualmente, se emparejan x e y si sim(x,y) $\geq \beta$
- Alternativa: Se emparejan cuando sim(x,y) $\geq \beta$, se descarta el emparejamiento si sim(x,y) $\leq \gamma$ y se revisan manualmente los casos intermedios.

Uso de reglas

Combinación lineal

EJEMPLO

$$sim(x,y) = \sum_{i=1}^{n} \alpha_i \cdot sim_i(x,y)$$

Table X

	Name	Phone	City	State
X_1	Dave Smith	(608) 395 9462	Madison	WI
X_2	Joe Wilson	(408) 123 4265	San Jose	CA
X_3	Dan Smith	(608) 256 1212	Middleton	WI

Table Y

	Name	Phone	City	State
Y_1	David D. Smith	395 9426	Madison	WI
Y_2	Daniel W. Smith	256 1212	Madison	WI

sim(x,y) =

$0.3s_{name}(x,y) + 0.3s_{phone}(x,y) + 0.1s_{city}(x,y) + 0.3s_{state}(x,y)$

- $s_{name}(x,y)$: Jaro-Winkler
- $s_{phone}(x,y)$: Distance de edición (sin prefijos)
- $s_{city}(x,y)$: Distancia de edición
- $s_{state}(x,y)$: Emparejamiento exacto (sí ➔ 1, no ➔ 0)

Uso de reglas

Combinación lineal

$$sim(x,y) = \sum_{i=1}^{n} \alpha_i \cdot sim_i(x,y)$$

Ventajas

- Conceptualmente simple, fácil de implementar.
- Se pueden aprender los pesos α a partir de un conjunto de datos de entrenamiento (regresión lineal).

Inconveniente

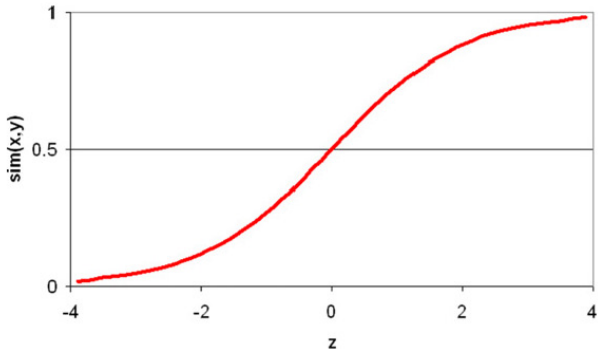
- Un incremento Δ en el valor de cualquier sim_i se traduce en un incremento $\Delta\alpha_i$ en el valor de sim(x,y)

Uso de reglas

Regresión logística

$$sim(x,y) = 1/(1+e^{-z})$$

$$z = \sum_{i=1}^{n} \alpha_i \cdot sim_i(x,y)$$



- Método más habitual de emparejamiento de datos.
- Útil cuando existen múltiples señales que pueden contribuir a que dos tuplas emparejen: cada señal contribuye a que las tuplas acaben emparejando.

Uso de reglas

Reglas más complejas

Cuando queremos aprovechar el conocimiento que tengamos sobre el problema…

EJEMPLO

Los datos de dos personas casan si sus nombres coinciden aproximadamente y, además, o bien sus teléfonos o bien sus direcciones coinciden exactamente.

```
if s_name(x,y)<0.8
   return false;
else if (e_phone(x,y) || e_address(x,y))
   return true;
else
   return false
```

Uso de reglas

Ventajas

- Conceptualmente simple.
- Fácil de comprender, implementar y depurar.
- Eficiente.
- Permite codificar conocimiento específico.

Inconvenientes

- Puede requerir mucho esfuerzo.
- Dificultad para establecer los parámetros adecuados.
- Puede que no sepamos cómo elaborar las reglas.

SOLUCIÓN: Técnicas de aprendizaje automático (I.A.)

Aprendizaje supervisado

Dado un conjunto de datos de entrenamiento, aprender un modelo M que nos permita emparejar pares de tuplas.

- **Conjunto de entrenamiento**
  $T = \{(x_1,y_1,l_1) \ldots (x_n,y_n,l_n)\}$,
  donde cada (x_i,y_i) es un par de tuplas l_i una etiqueta ("yes" si x_i casa con y_i y "no" en otro caso)

- **Conjunto de características**
  $\{f_1, \ldots, f_m\}$,
  cada una de las cuales cuantifica un aspecto que pueda ser relevante para emparejar las tuplas.

Aprendizaje supervisado

Dado un conjunto de datos de entrenamiento, aprender un modelo M que nos permita emparejar pares de tuplas.

- Se convierte cada ejemplo de entrenamiento (x_i,y_i,l_i) de T en un par $(<f_1(x_i,y_i), \ldots, f_m(x_i,y_i)>, c_i)$
  - $v_i = <f_1(x_i,y_i), \ldots, f_m(x_i,y_i)>$ es un vector de características que codifica el par de tuplas (x_i,y_i) en términos de las características $\{f_1, \ldots, f_m\}$.
  - c_i es una versión adecuada de la etiqueta l_i (por ejemplo, yes/no o 1/0)
- Como resultado, T es ahora $T' = \{(v_1,c_1), \ldots, (v_n,c_n)\}$.

Aprendizaje supervisado

Dado un conjunto de datos de entrenamiento, aprender un modelo M que nos permita emparejar pares de tuplas.

- Se aplica un algoritmo de aprendizaje sobre T' para aprender un modelo M: árboles de decisión, SVMs [Support Vector Machines], redes neuronales…
- Se utiliza el modelo M para realizar predicciones a la hora de emparejar nuevos pares de tuplas:
  - Dado un par (x,y), se transforma en un vector de características $v = <f_1(x,y), \ldots, f_m(x,y)>$.
  - Se aplica M sobre v para determinar si x e y emparejan o no.

Aprendizaje supervisado

EJEMPLO

Datos de entrenamiento

<a_1 = (Mike Williams, (425) 247 4893, Seattle, WA), b_1 = (M. Williams, 247 4893, Redmond, WA), yes>
<a_2 = (Richard Pike, (414) 256 1257, Milwaukee, WI), b_2 = (R. Pike, 256 1237, Milwaukee, WI), yes>
<a_3 = (Jane McCain, (206) 111 4215, Renton, WA), b_3 = (J. M. McCain, 112 5200, Renton, WA), no>

match names match phones match cities match states check area code against city

$v_1 = <[s_1(a_1,b_1), s_2(a_1,b_1), s_3(a_1,b_1), s_4(a_1,b_1), s_5(a_1,b_1), s_6(a_1,b_1)], 1>$

$v_2 = <[s_1(a_2,b_2), s_2(a_2,b_2), s_3(a_2,b_2), s_4(a_2,b_2), s_5(a_2,b_2), s_6(a_2,b_2)], 1>$

$v_3 = <[s_1(a_3,b_3), s_2(a_3,b_3), s_3(a_3,b_3), s_4(a_3,b_3), s_5(a_3,b_3), s_6(a_3,b_3)], 0>$

Aprendizaje supervisado

EJEMPLO

Entrenamiento del modelo: Regresión lineal

$v_1 = <[s_1(a_1,b_1), s_2(a_1,b_1), s_3(a_1,b_1), s_4(a_1,b_1), s_5(a_1,b_1), s_6(a_1,b_1)], 1>$

$v_2 = <[s_1(a_2,b_2), s_2(a_2,b_2), s_3(a_2,b_2), s_4(a_2,b_2), s_5(a_2,b_2), s_6(a_2,b_2)], 1>$

$v_3 = <[s_1(a_3,b_3), s_2(a_3,b_3), s_3(a_3,b_3), s_4(a_3,b_3), s_5(a_3,b_3), s_6(a_3,b_3)], 0>$

- Objetivo: Aprender una regla $s(a,b) = \sum \alpha_i s_i(a,b)$
- Entrenamiento: Regresión por mínimos cuadrados. Encontrar los pesos que minimizan el error cuadrático

$$\sum_{i=1}^{3} \left(c_i - \sum_{j=1}^{6} \alpha_i s_i(v_i) \right)^2$$

Aprendizaje supervisado

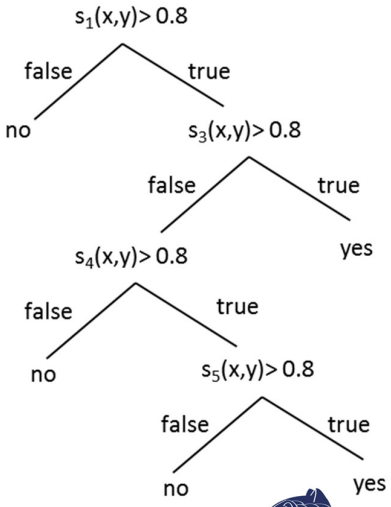
EJEMPLO

Entrenamiento del modelo: Árbol de decisión

$v_1 = <[s_1(a_1,b_1), s_2(a_1,b_1), s_3(a_1,b_1), s_4(a_1,b_1), s_5(a_1,b_1), s_6(a_1,b_1)], \text{yes}>$

$v_2 = <[s_1(a_2,b_2), s_2(a_2,b_2), s_3(a_2,b_2), s_4(a_2,b_2), s_5(a_2,b_2), s_6(a_2,b_2)], \text{yes}>$

$v_3 = <[s_1(a_3,b_3), s_2(a_3,b_3), s_3(a_3,b_3), s_4(a_3,b_3), s_5(a_3,b_3), s_6(a_3,b_3)], \text{no}>$



- Objetivo:
  Construir un árbol de decisión.
- Entrenamiento:
  Algoritmo "greedy" heurístico de tipo divide y vencerás (p.ej. ID3 o C4.5)

Aprendizaje supervisado

Ventajas (en comparación con el uso de reglas)

- No hace falta decidir manualmente si una característica en particular resulta útil.
- Al automatizar el proceso, no tenemos por qué limitar el conjunto de características considerado.
- Se construyen automáticamente reglas todo lo complejas que resulte necesario.

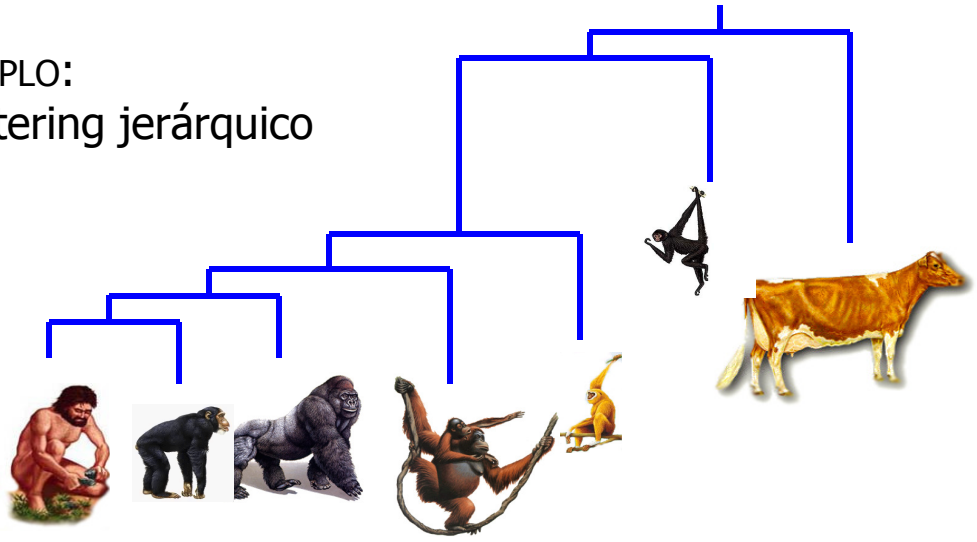
Inconvenientes

- Las reglas puede que no resulten interpretables.
- Se requiere un conjunto de entrenamiento con ejemplos ya etiquetados, que puede resultar difícil de conseguir o preparar.

Aprendizaje no supervisado

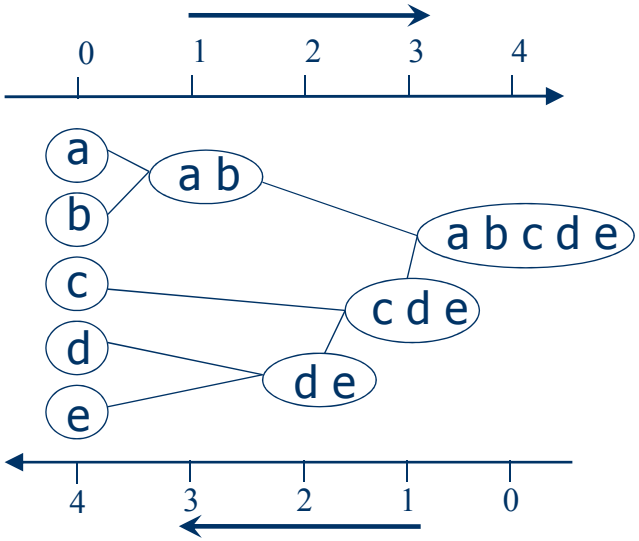
Se construye un modelo M que nos permita emparejar pares de tuplas sin necesidad de un conjunto de entrenamiento previamente etiquetado.

EJEMPLO:
Clustering jerárquico



Aprendizaje no supervisado

Clustering jerárquico aglomerativo
(AGNES: AGglomerative NESting)



vs. Divisivo (DIANA: Divisive ANAlysis)

Aprendizaje no supervisado

Clustering jerárquico aglomerativo

- Se comienza con cada tuplas como cluster individual.
- En cada paso, se combina el par de clusters más cercanos (hasta que se haya obtenido el número de clusters deseado o la medida de similitud entre los dos clusters más cercanos quede por debajo de un umbral preestablecido).
- Resultado: Partición del conjunto de tuplas en un conjunto de clusters (todas las tuplas del mismo cluster hacen referencia a la misma entidad).

Aprendizaje no supervisado

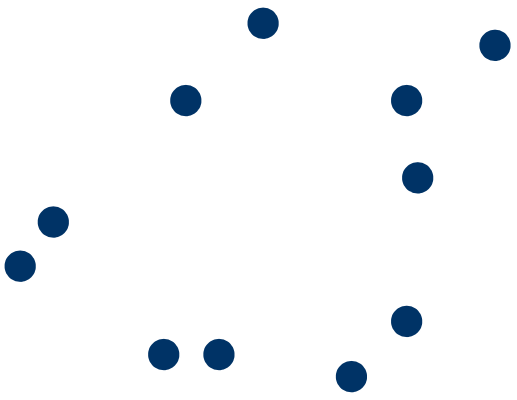
Clustering jerárquico aglomerativo

Inicialización:

Clusters de casos individuales

y matriz de similitudes

	p1	p2	p3	p4	p5
p1					
p2					
p3					
p4					
p5					



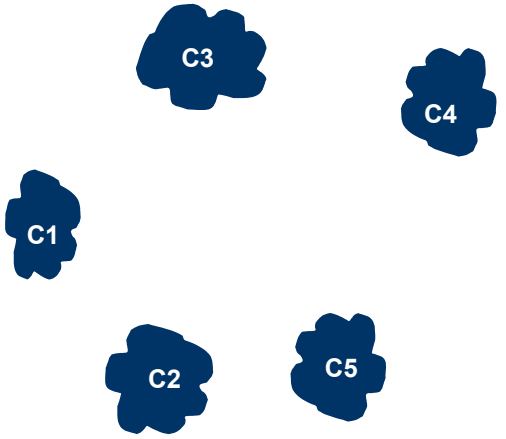
Aprendizaje no supervisado

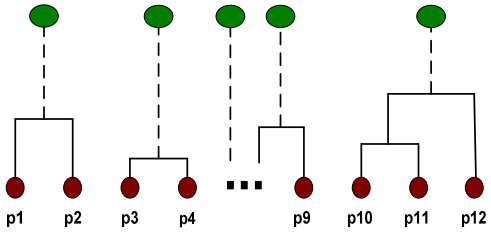
Clustering jerárquico aglomerativo

Tras varias iteraciones:

Varios clusters...

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					





Aprendizaje no supervisado

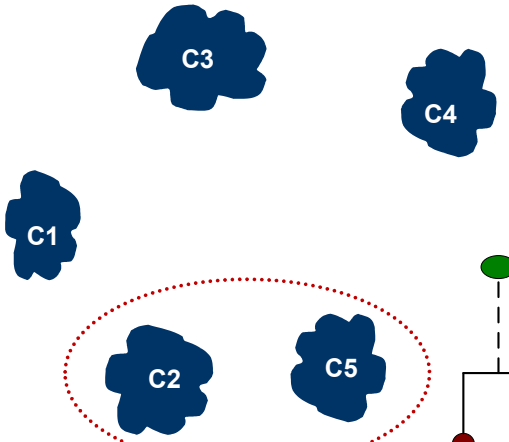
Clustering jerárquico aglomerativo

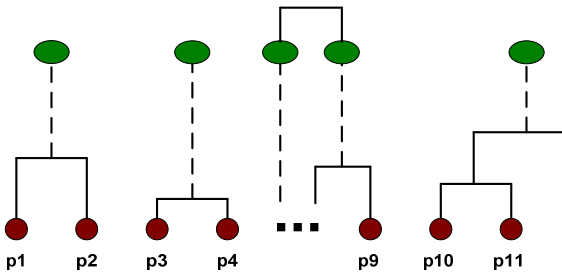
Combinamos los clusters 2 y 5

y actualizamos la matriz de similitudes

¿cómo?

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					



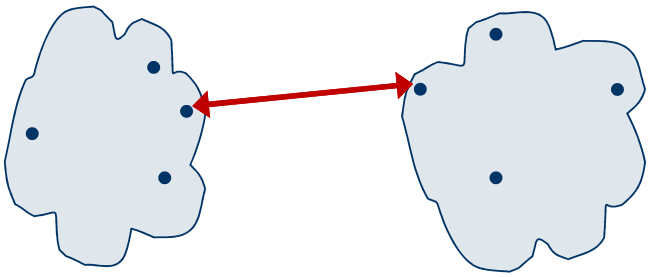


Aprendizaje no supervisado

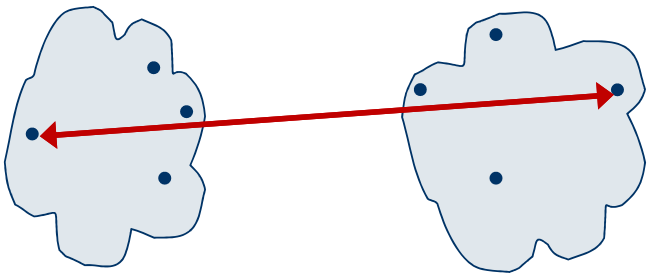
Clustering jerárquico aglomerativo

¿Cómo se mide la distancia entre clusters?

- MIN
  [single-link]



- MAX
  [complete-link]
  (diameter)

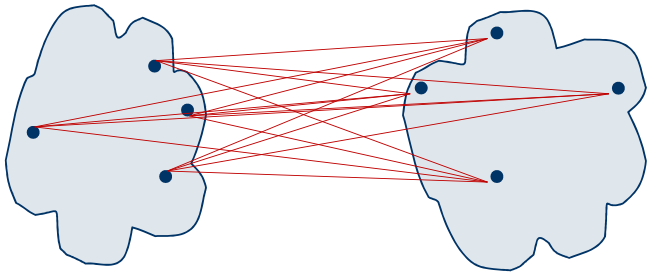


Aprendizaje no supervisado

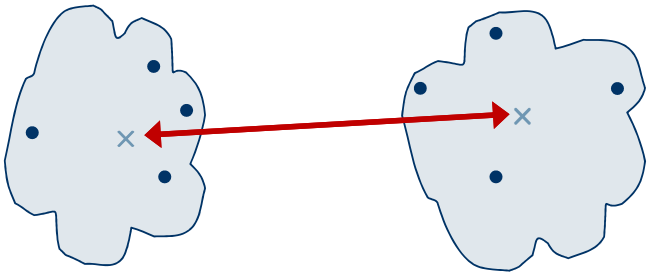
Clustering jerárquico aglomerativo

¿Cómo se mide la distancia entre clusters?

- Promedio
  [average-link]



- Centroides
  [canonical tuple]
  p.ej. BIRCH

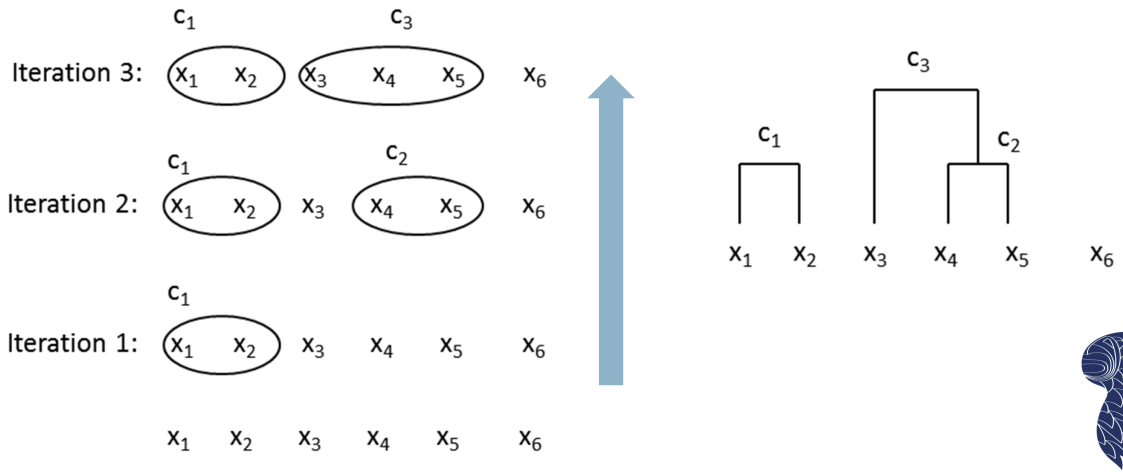


Aprendizaje no supervisado

Clustering jerárquico aglomerativo

EJEMPLO

$sim(x,y) = 0.3s_{name}(x,y) + 0.3s_{phone}(x,y) + 0.1s_{city}(x,y) + 0.3s_{state}(x,y)$



Aprendizaje no supervisado

Ideas clave

- Convierte el problema de emparejar tuplas en un problema de construir entidades (agrupamientos).
- Lo que vamos aprendiendo nos sirve para mejorar:

  Las tuplas de un clúster nos permiten ir construyendo "perfiles de entidades" que luego utilizamos para emparejar otras tuplas.

  ➔ Combinar información potencialmente útil.

Métodos probabilísticos

IDEA: Modelar utilizando distribuciones de probabilidad.

Ventajas

- Permiten incorporar de forma natural nuevas fuentes de conocimiento.
- Aprovechan las técnicas probabilísiticas ya desarrolladas (estadística e I.A.)

Inconvenientes

- Costosos computacionalmente.
- Decisiones difíciles de comprender (y depurar).

Métodos probabilísticos

La mayor parte de las técnicas utilizan
modelos generativos (p.ej. Redes bayesianas)

- Se codifica una distribución de probabilidad completa
- Se describe cómo generar datos acordes a dicha distribución de probabilidad.

Algunas técnicas recientes utilizan
modelos discriminativos
(p.ej. conditional random fields)

- Se codifican sólo las probabilidades necesarias para realizar el emparejamiento (p.ej. la probabilidad de una etiqueta dado un par de tuplas).

Métodos probabilísticos

Redes bayesianas

MOTIVACIÓN

- Dado un conjunto de variables X, un estado es una asignación de valores a todas las variables de X.
- Una distribución de probabilidad P asigna a cada estado s una probabilidad P(s).
- Razonamiento con modelos probabilísticos: P(A=a)?
- Problema: No podemos enumerar todos los estados si tenemos muchas variables (número exponencial de estados diferentes).

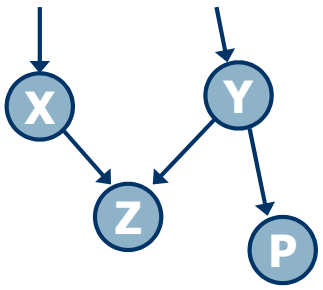
Métodos probabilísticos

Redes bayesianas

Una forma compacta de representar distribuciones de probabilidades:

Representan mediante un grafo dirigido acíclico dependencias entre variables, especificando sus distribuciones de probabilidad conjuntas.

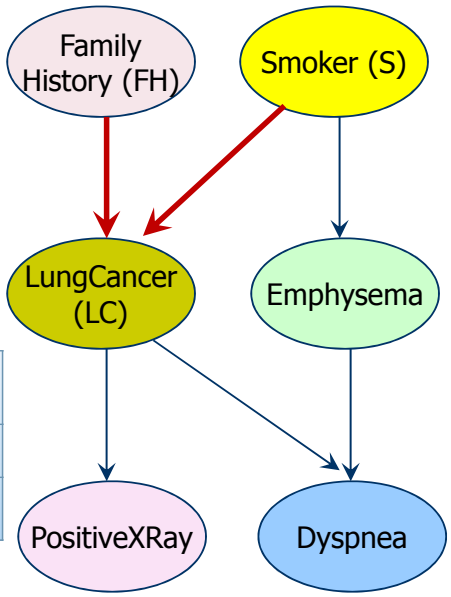
- Nodos: Variables.
- Enlaces: Dependencias probabilísticas.



Métodos probabilísticos

Redes bayesianas

EJEMPLO



CPT [Conditional Probability Table] para la variable LungCancer:

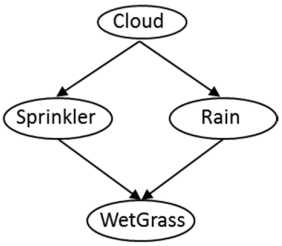
P(LC|...)	(FH,S)	(FH, ~S)	(~FH,S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~ LC	0.2	0.5	0.3	0.9

Muestra la probabilidad condicional de que alguien desarrolle cáncer de pulmón para cada combinación de las variables que lo "causan".

Métodos probabilísticos

Redes bayesianas

OTRO EJEMPLO



Cloud	
t	f
0.3	0.7

Cloud	Sprinkler	
	on	off
t	0.2	0.8
f	0.8	0.2

Cloud	Rain	
	t	f
t	0.6	0.4
f	0.3	0.7

Sprinkler	Rain	WetGrass	
		t	f
on	t	1	0
on	f	1	0
off	t	1	0
off	f	0.1	0.9

P(C,S,R,W) = P(C)·P(S|C)·P(R|C)·P(W|R)

Para calcular P(C,S,R,W) sólo necesitamos 4 distribuciones locales de probabilidad (CPTs): 9 parámetros en vez de 15 (2^4-1).

Métodos probabilísticos

Redes bayesianas

RAZONAMIENTO (A.K.A. INFERENCIA)

Calcular P(A) o P(A|B)
donde A y B son subconjuntos de variables

Problema:

La inferencia exacta es un problema NP-duro
(tiempo exponencial en el número de variables).

Soluciones:

- Algoritmos polinómicos para ciertas clases de BNs.
- Algoritmos aproximados.

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE

Típicamente, se recurre a un experto para crear el grafo de la red bayesiana y, a continuación, se aprenden las CPTs a partir de un conjunto de entrenamiento.

Dos casos diferenciados:

- Aprendizaje sin valores desconocidos
- Aprendizaje con valores desconocidos (EM).

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE SIN VALORES DESCONOCIDOS

Sean θ las probabilidades que deseamos estimar:

Queremos calcular las probabilidades θ^* que maximicen la probabilidad de observar el conjunto de datos de entrenamiento D: $\theta^* = \arg\max_\theta P(D|\theta)$

θ^* puede obtenerse contando las veces que aparecen los distintos valores en D.

¿Y si no tenemos suficientes datos para ciertos estados? Las probabilidades se suavizan…
p.ej. Suavizado de Laplace, $\theta_i = (x_i+1)/(N+k)$

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE SIN VALORES DESCONOCIDOS

Training data D

$d_1 = (1,0)$
$d_2 = (1,0)$
$d_3 = (1,1)$
$d_4 = (0,1)$

CPTs to be learned

A	
1	0
?	?

A	B	
	1	0
1	?	?
0	?	?

CPTs learned from training data

A	
1	0
0.75	0.25

A	B	
	1	0
1	0.33	0.67
0	1	0

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE CON VALORES DESCONOCIDOS: ALGORITMO EM

El conjunto de entrenamiento puede incluir valores desconocidos (a.k.a. valores nulos):

- Ejemplos para los que no se observó una variable.
- Situaciones en las que no resulta aplicable una medida.

No se puede contar ocurrencias para aprender (i.e. inferir CPTs), por lo que recurrimos al algoritmo EM…

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE CON VALORES DESCONOCIDOS: ALGORITMO EM

EM [Expectation Maximization]

- Se desconocen tanto las probabilidades θ como los valores desconocidos en el conjunto de datos D.
- Se estiman iterativamente ambos: Se utiliza uno para predecir el otro y viceversa, hasta que el algoritmo converja (partiendo de una asignación inicial, claro).

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE CON VALORES DESCONOCIDOS: ALGORITMO EM

Training data D

$d_1 = (?,0)$
$d_2 = (?,0)$
$d_3 = (?,1)$

The EM algorithm

1. Initialize θ and let it be θ^0. Set n = 0.
2. (Expectation) Use θ^n to estimate missing values of D. Let the resulting set be D^n.
3. (Maximization) Compute θ^{n+1} using counting over D^n.
4. Exit and return θ^n if no increase is achieved for $P(\theta^n | D^n)$. Otherwise repeat Steps 2-3 with n = n + 1.

Métodos probabilísticos

Redes bayesianas

APRENDIZAJE CON VALORES DESCONOCIDOS: ALGORITMO EM

Training data D

$d_1 = (?,0)$
$d_2 = (?,0)$
$d_3 = (?,1)$

θ^0

A	
1	0
0.5	0.5

A	B	
	1	0
1	0.6	0.4
0	0.5	0.5

D^0

$d_1 = \begin{bmatrix} P(A=1) = 0.44 \\ P(A=0) = 0.56 \end{bmatrix}, B=0$

$d_2 = \begin{bmatrix} P(A=1) = 0.44 \\ P(A=0) = 0.56 \end{bmatrix}, B=0$

$d_3 = \begin{bmatrix} P(A=1) = 0.54 \\ P(A=0) = 0.46 \end{bmatrix}, B=1$

θ^1

A	
1	0
0.47	0.53

A	B	
	1	0
1	0.38	0.62
0	0.29	0.71

Métodos probabilísticos

Redes bayesianas

- El algoritmo EM también intenta encontrar las probabilidades θ que maximicen $P(D|\theta)$, si bien puede que no encuentre el óptimo global θ^* y converja a un máximo local.

- Los **modelos generativos** (como las redes bayesianas) codifican distribuciones de probabilidad completas y especifican cómo generar datos acordes a dichas distribuciones de probabilidad. Ahora bien, ¿cómo las aplicamos a nuestro problema de integración de datos?

Métodos probabilísticos

Naïve Bayes

- Teorema de Bayes
  $P(M|S_1..S_n) = P(S_1..S_n|M)\ P(M) / P(S_1..S_n)$

- Si asumimos que $S_1..S_n$ son independientes dado M:
  $P(S_1..S_n|M) = \Pi\ P(S_i|M)$

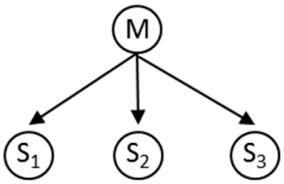
- Además,
  $P(S_1..S_n) = P(S_1..S_n|M=t)P(M=t)+P(S_1..S_n|M=f)P(M=f)$

Esto es, sólo necesitamos conocer $P(S_i|M)$ y $P(M)$…

Métodos probabilísticos

Naïve Bayes

- El modelo en el que $S_1..S_n$ son independientes dado M viene representado por la siguiente red bayesiana:

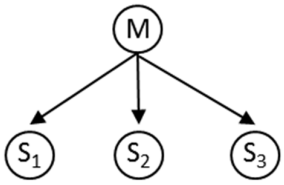


- Calcular $P(M|S_1..S_n)$ es realizar una inferencia en la red bayesiana del modelo Naïve Bayes, que puede realizarse fácilmente dadas las CPTs $P(S_i|M)$ y $P(M)$.

Métodos probabilísticos

Naïve Bayes

CPTs DADO UN CONJUNTO DE ENTRENAMIENTO



$(a_1, b_1,$ yes)
$(a_2, b_2,$ yes)
$(a_3, b_3,$ no)

<t, $s_1(a_1,b_1), s_2(a_1,b_1), s_3(a_1,b_1)$>
<t, $s_1(a_2,b_2), s_2(a_2,b_2), s_3(a_2,b_2)$>
<f, $s_1(a_3,b_3), s_2(a_3,b_3), s_3(a_3,b_3)$>

- Se convierten los ejemplos de entrenamiento en vectores de características y se aplica el algoritmo de aprendizaje sin valores desconocidos (conteo).

- Se aplica la red bayesiana aprendida para emparejar un par de tuplas (a,b) comparando

$$\mathbf{P(M=t|a,b)} = \Pi\, P(S_i|M=t)\, P(M=t)$$

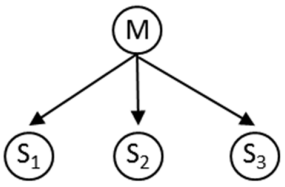
y $$\mathbf{P(M=f|a,b)} = \Pi\, P(S_i|M=f)\, P(M=f)$$

Métodos probabilísticos

Naïve Bayes

CPTs SIN CONJUNTO DE ENTRENAMIENTO

(a_4, b_4)	$<?, s_1(a_4,b_4), s_2(a_4,b_4), s_3(a_4,b_4)>$
(a_5, b_5)	$<?, s_1(a_5,b_5), s_2(a_5,b_5), s_3(a_5,b_5)>$
(a_6, b_6)	$<?, s_1(a_6,b_6), s_2(a_6,b_6), s_3(a_6,b_6)>$



Convertimos los pares que deseamos emparejar en vectores de características con valores desconocidos (la etiqueta es, en este caso, el valor desconocido):

- Aplicamos el algoritmo EM para aprender tanto las CPTs como las etiquetas desconocidas a la vez.

Métodos probabilísticos

Ideas clave

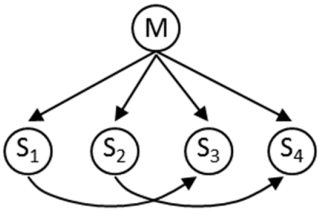
- El diseñador proporciona la estructura de la red bayesiana, un grafo dirigido acíclico (p.ej. modelo Naïve Bayes).

- Podemos utilizar la red bayesiana tanto si disponemos de un conjunto de datos de entrenamiento (conteo) como si no (algoritmo EM).

NOTA: El mismo procedimiento puede generalizarse para redes bayesianas arbitrarias (no sólo Naïve Bayes).

Métodos probabilísticos

Naïve Bayes asume que no existen correlaciones entre $S_1..S_n$ pero podemos incluir dichas correlaciones en nuestro modelo construyendo redes bayesianas más complejas:



Problema:

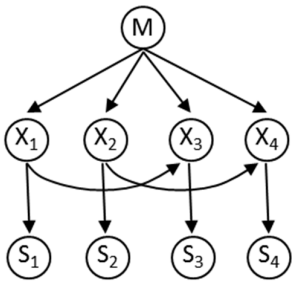
Aumenta el número de parámetros de las CPTs.

N características, P padres por nodo, D valores por nodo

$O(ND^P)$ frente a 2DN del modelo Naïve Bayes

Métodos probabilísticos

Una posible solución:



- Tuplas con k atributos.
- k características $S_1..S_k$ para comparar los valores de cada atributo por separado.
- Añadimos k variables binarias X_i que nos indican si los atributos deberían coincidir si las tuplas casan.

Modelando las correlaciones al nivel de las variables X_i se reduce el número de parámetros de las CPTs:

$O(k2^P+2kD)$ frente a $O(ND^P)$ del modelo más general.

Métodos probabilísticos

En la práctica

- Si el modelo probabilístico tiene demasiados parámetros, se necesitan muchos datos para ajustar esos parámetros y el aprendizaje puede ser ineficiente
  ➔ Suposiciones de independencia
  para reducir el número de parámetros del modelo
  y optimizaciones del algoritmo EM (p.ej. inicialización).

- Una vez aprendido el modelo,
  la inferencia puede ser computacionalmente costosa
  ➔ Algoritmos aproximados y simplificaciones.

Emparejamiento colectivo

Las técnicas vistas hasta ahora toman decisiones independientes a la hora de realizar emparejamientos:

Se decide si el par (a,b) casa independientemente de si existe otro par (c,d) que case y pueda estar relacionado con (a,b).

En la práctica, las decisiones de emparejamiento están a menudo correlacionadas y aprovechar esas correlaciones puede mejorar la precisión de nuestros algoritmos…

Emparejamiento colectivo

EJEMPLO

Coautores de artículos

W. Wang, C. Chen, A. Ansari, A mouse immunity model
W. Wang, A. Ansari, Evaluating immunity models
L. Li, C. Chen, W. Wang, Measuring protein-bound fluxetine
W. J. Wang, A. Ansari, Autoimmunity in biliary cirrhosis

	First initial	Middle inititial	Last name
a_1	W		Wang
a_2	C		Chen
	...	...	...
a_9	W	J	Wang
a_{10}	A		Ansari

Objetivo:
Reconciliar referencias a los autores de los artículos.

Una primera solución:
Extraer los nombres de los autores y crear una tabla sobre la que aplicar las técnicas ya vistas…

No funciona :-(

… al no aprovechar las relaciones entre coautores

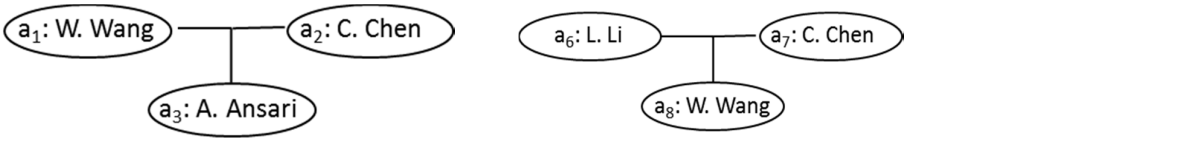
Emparejamiento colectivo

EJEMPLO

Coautores de artículos

W. Wang, C. Chen, A. Ansari, A mouse immunity model
W. Wang, A. Ansari, Evaluating immunity models
L. Li, C. Chen, W. Wang, Measuring protein-bound fluxetine
W. J. Wang, A. Ansari, Autoimmunity in biliary cirrhosis

Hipergrafo

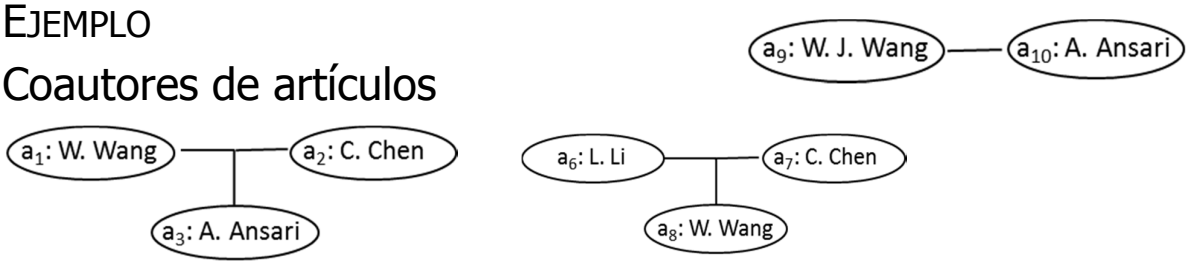


Si hemos emparejado a_3 con a_5, intuitivamente es más probable que a_1 y a_4 emparejen (mismo nombre + relación de coautor)

Emparejamiento colectivo

EJEMPLO

Coautores de artículos



Una "segunda" solución":

Añadir coautores a las tuplas

p.ej. a_1.coauthors = {C.Chen, A.Ansari}

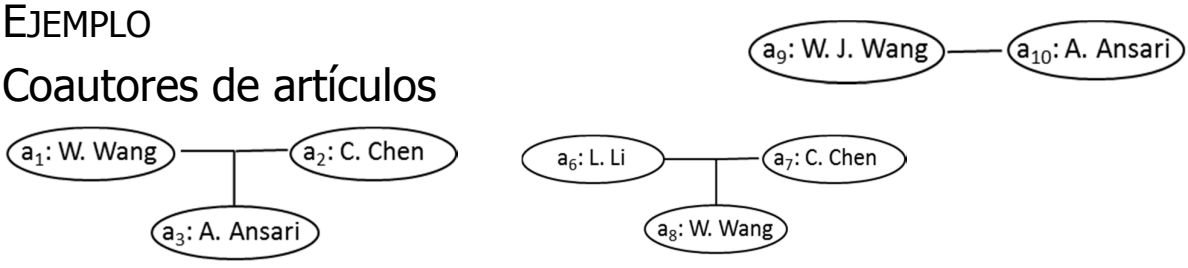
Emparejar usando las técnicas ya conocidas

p.ej. Coeficiente de Jaccard sobre A.coauthors

Emparejamiento colectivo

EJEMPLO

Coautores de artículos



Problema:

Si dos autores distintos a_3 y a_5 comparten el mismo nombre, los emparejaríamos aumentando la probabilidad de que a_1 y a_4 también acaben emparejándose.

Solución:

Realizar los emparejamientos de forma colectiva.

Emparejamiento colectivo

Un algoritmo de emparejamiento colectivo:

Método de agrupamiento jerárquico aglomerativo en el que la medida de similitud la modificamos para considerar correlaciones:

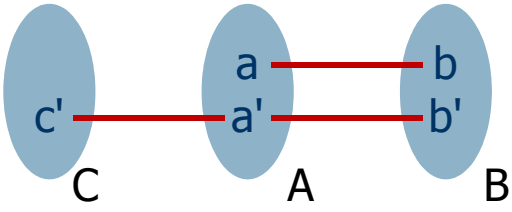
$sim(A,B) = \alpha\ sim_{atributos}(A,B) + (1-\alpha)\ sim_{vecinos}(A,B)$

- $sim_{atributos}(A,B)$ sólo utiliza los atributos de A y B (p.ej. single-link, complete-link, average-link…).
- $sim_{vecinos}(A,B)$ considera las correlaciones…

Emparejamiento colectivo

$sim_{vecinos}(A,B)$

N(A): Identificadores de los clusters de los nodos relacionados con algún nodo de A.



N(A)={B,B,C}

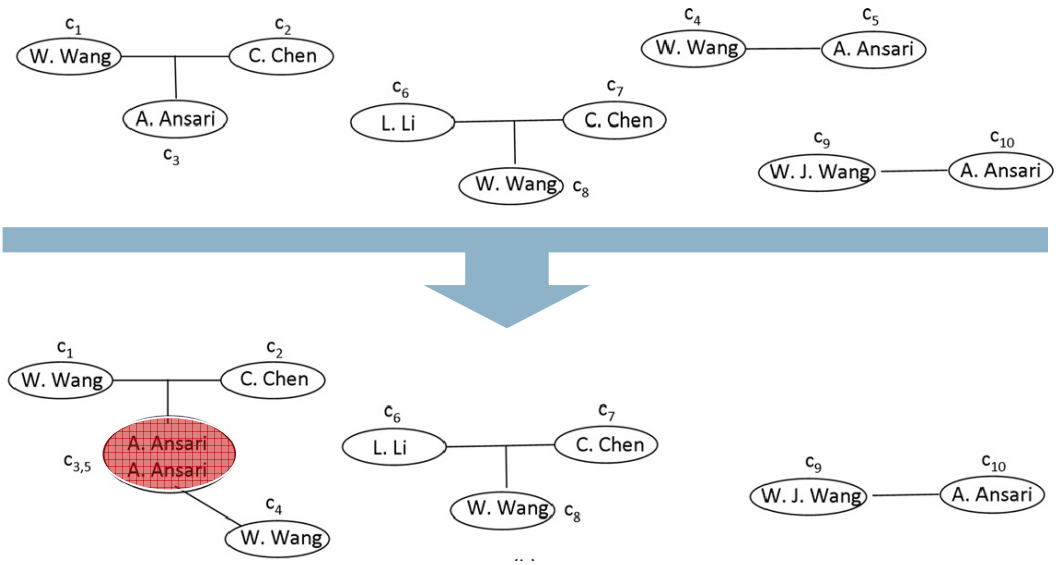
$sim_{vecinos}(A,B)$ = Jaccard(N(A),N(B))

= $|N(A) \cap N(B)| \ / \ |N(A) \cup N(B)|$

Emparejamiento colectivo

Agrupamiento jerárquico aglomerativo

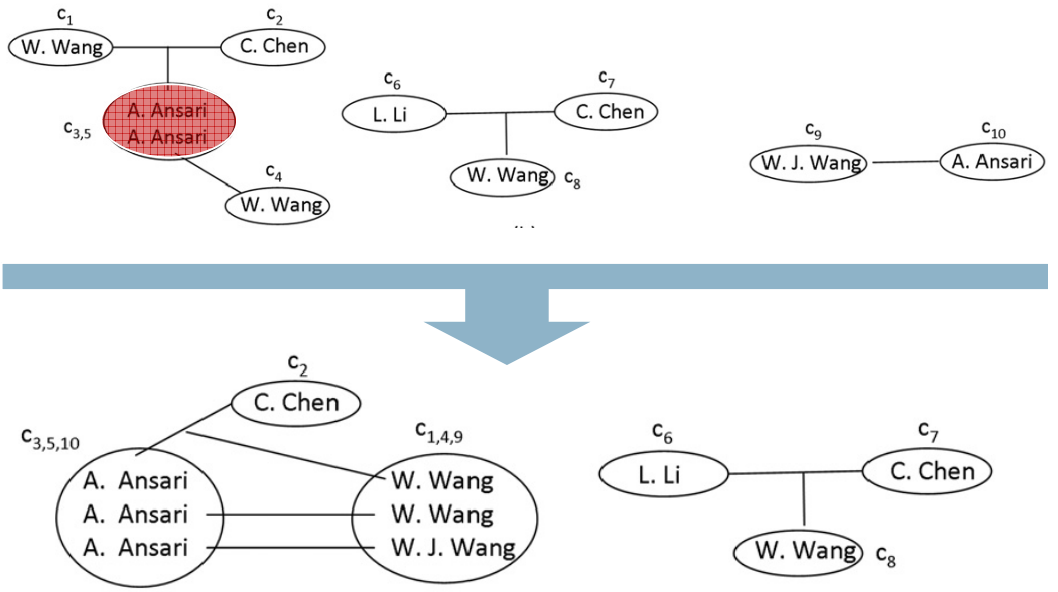
$sim(A,B) = \alpha\ sim_{atributos}(A,B) + (1-\alpha)\ sim_{vecinos}(A,B)$



Emparejamiento colectivo

Agrupamiento jerárquico aglomerativo

$sim(A,B) = \alpha\ sim_{atributos}(A,B) + (1-\alpha)\ sim_{vecinos}(A,B)$



Escalabilidad

¿Cómo mejorar la eficiencia
en el proceso de integración de datos?

- Reduciendo el número de comparaciones necesario (el número de pares de tuplas que se evalúan).
- Minimizando el tiempo empleado en cada comparación (la eficiencia de la evaluación de cada posible par).

Escalabilidad

Reducción del número de comparaciones

- **Hashing** (distribuir las tuplas en cubetas [buckets] y comparar sólo las tuplas dentro de cada cubeta).

  p.ej. por códigos postales

- **Ordenación** (ordenar las tuplas y comparar cada tupla sólo con las W-1 anteriores, donde W es el tamaño de ventana).

  p.ej. IDs (DNI, SSN…), apellidos, Soundex…

Escalabilidad

Reducción del número de comparaciones

- **Indexación** (indexar las tuplas de forma que se pueda utilizar el índice para que, dada una tupla, se pueda obtener un pequeño conjunto de tuplas potencialmente similares).

  p.ej. índice invertido por nombres

- **"Canopies" ["doseles"]**, a.k.a. "conjuntos paraguas" (usando una medida de similitud sencilla, se agrupan las tuplas en grupos solapados, para luego utilizar una medida más sofisticada para comparar tuplas del mismo dosel) ≈ hashing con solapamiento.

Escalabilidad

Reducción del número de comparaciones

- **Representantes:**

  Se construyen grupos de tuplas que emparejan dentro del grupo pero no con tuplas de otros grupos y se crea un representante del grupo, ya sea una tupla del grupo o un "prototipo" mezclando tuplas del grupo.

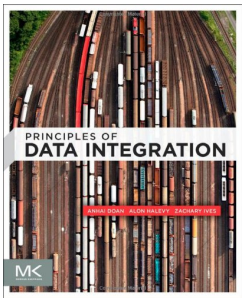
  Llegada una nueva tupla, se compara sólo con los representantes de los grupos.

Escalabilidad

Procesamiento paralelo

Bibliografía recomendada

- Hai Doan, Alon Halevy & Zachary Ives:
  Principles of Data Integration
  Morgan Kaufmann, 1st edition, 2012.
  ISBN 0124160441
  http://research.cs.wisc.edu/dibook/



Chapter 7: Data Matching