Casamento aproximado de dados e esquemas

Carlos A. Heuser

Universidade Federal do Rio Grande do Sul

Instituto de Informática

Porto Alegre - Brazil

Roteiro

- Introdução ao casamento de dados
- Casamento de instâncias
- Casamento de esquemas

Problemas de pesquisa em integração de dados

- Integração de esquemas
- Casamento de esquemas
- Mapeamentos entre esquemas
- **Casamento de instâncias**

Casamento aproximado de dados

- Objetivo

Determinar se
dois objetos de dados diferentes
representam
o mesmo objeto da vida real

Exemplo – casamento de *strings*

- Objetivo:

  Determinar se duas **cadeias de caracteres** diferentes representam o mesmo objeto da vida real

- Possíveis cadeias de caracteres:

```
UFRS

UFRGS

Univ. Fed. do Rio Grande do Sul

Universidade Federal do Rio Grande do Sul
```

Exemplo casamento de registros

- Determinar se dois **registros** representam o mesmo objeto da vida real:

Instituição	Cidade	Data Inscrição
Universidade Federal do Rio de Janeiro (UFRJ)	Rio deJaniero	23.jan.2003

Instituição	Cidade	Data Inscrição
Universidade federal do Rio de Janeiro	Rio de Janeiro	2003.01.23

Aplicações de casamento de dados

- **Integração de dados** oriundos de fontes diferentes
- **Limpeza de dados (*data cleaning*)** coletados pela WEB
  - (por exemplo, dados de clientes)
- **Consultas aproximadas** a bancos de dados
- Ponto comum:
  - entidades não têm uma chave primária para união ou junção.

Casamento aproximado de dados

- Tema antigo de pesquisa:
  - banco de dados
  - aprendizagem de máquina
  - recuperação de informações
- Termos:
  - *instance matching*
  - *object matching*
  - *merge/purge problem*
  - *fuzzy match*
  - *data linkage*
  - *record linkage*
  - *Entity resolution*
  - *....*

Onde casamento de instâncias é usado

Integração de dados

- Determinar se duas instâncias de dados representam o mesmo objeto da realidade.

- Dois tipos:
  - Casamento de **instâncias**
  - Casamento de **esquemas**

Consultas por similaridade

- Encontrar todas instâncias de dados que representam o mesmo objeto na vida real

Roteiro

- Introdução ao casamento de dados
- **Casamento de instâncias**
- Casamento de esquemas

Função de similaridade

- Casamento é computado por uma **função de similaridade**:

$$f(v_1,v_2) \rightarrow s$$

- O **escore de similariade** s define a similaridade entre duas instâncias de dados v_1 e v_2
- Escores mais altos signficam similaridade maior
- Aqui, assumimos $s \in [0..1]$

Limiar

- Casamento de instâncias é um problema **booleano** (falso/verdadeiro):
  - Casamento tem que resultar em **falso** (não casam) ou **verdadeiro** (casam).

- Mas, funções de similaridade **retornam valores numéricos** entre 0 e 1

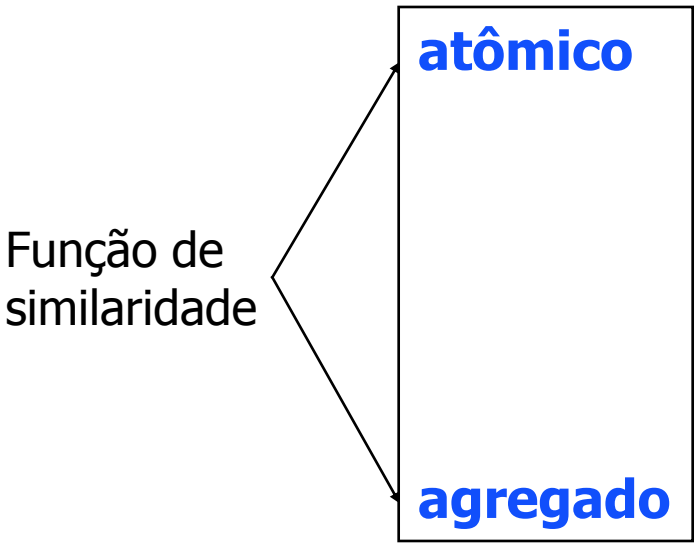
- Escore tem que ser transformado em booleano

- Um **limiar (*threshold*)** t tem que ser definido

  - $s \geq t \rightarrow$ instâncias casam
  - $s < t \rightarrow$ instâncias não casam

Taxonomia de funções de similaridade

- Funções de similaridade podem ser classificadas de acordo com o tipo de instâncias de dados que casam



Funções de similaridade de atômicos (nomes, strings)

- Há um grande número de funções na literatura

- Podem ser classificadas em:

  - Funções baseadas em **caracteres**

    Comparam **caracteres individuais** que compõe as cadeias

  - Funções baseadas em **termos**

    Comparam os **termos** (**palavras**) que compõe as cadeias

Funções baseadas em caracteres

- Distância de **edição** (**Levenshtein**)
  - É uma função de **distância** não de similaridade
- A distância entre dois strings é o número de operações de :
  - exclusão,
  - inserção, or
  - substituição

  de um caractere requeridas para transformar um string no outro

Exemplo de distância de edição

Calcular a distância entre as cadeias:

Koffi Anan

e

Kofi Annan

Edit distance - example

- f_{edit}(Koffi Anan , Kofi Annan)

v_1	v_2	Edit operation
Koffi_Anan	K	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy
Koffi_Anan	Koffi_An	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy
Koffi_Anan	Koffi_An	copy
Koffi_Anan	Koffi_Ann	**Insert n**

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy
Koffi_Anan	Koffi_An	copy
Koffi_Anan	Koffi_Ann	**Insert n**
Koffi_Anan	Koffi_Anna	copy

Edit distance - example

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy
Koffi_Anan	Koffi_An	copy
Koffi_Anan	Koffi_Ann	**Insert n**
Koffi_Anan	Koffi_Anna	copy
Koffi_Anan	Koffi_Annan	copy

Edit distance - example

- f_{edit}(Koffi Anan , Kofi Annan) = 2

v_1	v_2	Edit operation
Koffi_Anan	K	copy
Koffi_Anan	Ko	copy
Koffi_Anan	Kof	copy
Koffi_Anan	Kof	**Delete f**
Koffi_Anan	Kofi	copy
Koffi_Anan	Kofi_	copy
Koffi_Anan	Koffi_A	copy
Koffi_Anan	Koffi_An	copy
Koffi_Anan	Koffi_Ann	**Insert n**
Koffi_Anan	Koffi_Anna	copy
Koffi_Anan	Koffi_Annan	copy

Função de distância vs. função de similaridade

- Resultado de uma função de distância tem que ser transformado em escore de similaridade

- Exemplo:
  - distância: 2 operações
  - total de caracteres dos strings: 20 caracteres
  - simialridade: 18/20 = 0.9

Funções baseadas em termos

- **Jaccard**
  - Número de palavras comuns entre dois strings dividido pelo número total de palavras
- **TF-IDF**
  - Função comum em recuperação de informações
  - Termos recebem pesos de forma a ponderar menos palavras mais comuns

Funções baseadas em caracteres x Funções baseadas em termos

- Baseadas em **caracteres**
  - Manipulam bem erros de ortografia
- Baseadas em **termos**
  - Aceitam ordem diferente de termos

Funções atômicas

- Um grande número de funções já foi definido

- Bibliotecas de software (Java):

  - William Cohen's SecondString library:
    http://secondstring.sourceforge.net/

  - SimMetric library:
    http://sourceforge.net/projects/simmetrics/

Aplicando funções de similaridade - problemas

- Funções de similaridade são **dependentes de domínio**:
  - Nomes de pessoas
  - Nomes de organizações a acrônimos
  - Cadeias com uma palavra
  - Cadeias com muitas palavras e erros de ortografia
  - ...
- Como definir o **limiar**?
  - Distribuição de valores de escore varia de uma função para a outra

Casamento de instâncias - Conclusões

- Há muitas aplicações de integração de dados
- Nem sempre a técnica de junção clássica de bases de dados relacionais resolve
  - falta de chave primária
- Técnicas baseadas em similaridade são necessárias
- Já existem muitas funções disponíveis

Casamento de esquemas

- Modelos de dados que distinguem **esquema** de **instâncias**:
  - XML, relacional, ...

- Distingue-se:
  - casamento de esquemas de
  - casamento de instâncias.

- Objetivo do casamento de esquemas:
  - Apoiar a definição de **mapeamentos** entre esquemas
  - Mapeamento:

    Especifica como sub-conjuntos de elementos de um esquema S_1 correspondem a sub-conjuntos de S_2

Roteiro

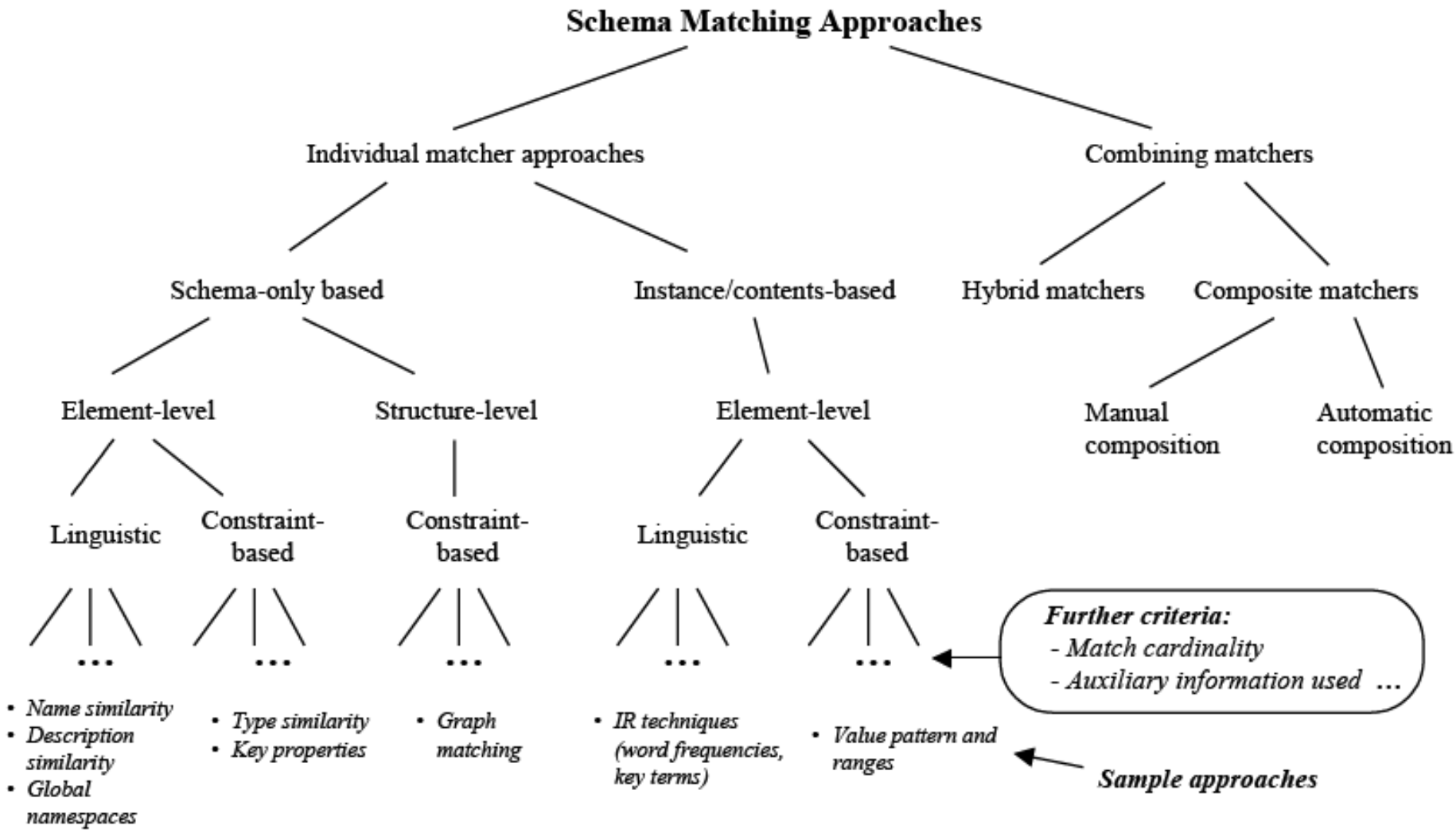
- Introdução ao casamento de dados
- Casamento de instâncias
- **Casamento de esquemas**

Exemplo de casamentos

Table 2. Full vs partial structural match (example)

S1 elements	S2 elements	
Address	CustomerAddress	full structural match of
Street	Street	Address and CustomerAddress
City	City	
State	USState	
ZIP	PostalCode	
AccountOwner	Customer	partial structural match of AccountOwner and
Name	Cname	Customer
Address	CAddress	
Birthdate	CPhone	
TaxExempt		

Taxonomia de casadores de esquema



Esquema vs. instância

- Alguns casadores não consideram dados (instâncias), apenas o esquema.

- Outros casadores usam os valores dos dados
  - Exemplo: duas colunas de nome preço que têm distribuição de valores bem diferente, podem não representar o mesmo atributo

Elementos vs. estutura

- Falando de forma geral, um esquema pode ser considerado como sendo um **grafo** formado por **elementos** (por exemplo, nomes de tabelas e nomes de colunas no relacional) e **arcos** (**ligações estruturais** entre elementos – por exemplo, as ligações de um nome de tabela com os nomes de suas colunas).
- Um casador pode considerar somente os elementos ou pode considerar também relações estruturais

Lingüístico vs. restrições

- Um casador pode levar em conta **somente aspectos lingüísticos**, isto é considerar apenas os nomes dos elementos

- Um casador pode levar em conta **restrições de integridade**:
  - Casar colunas que têm o mesmo tipo;
  - Casar colunas que forma a chave primária,...

Cardinalidade do resultado

- Casadores podem ser classificados de acordo com o número de elementos que são casados a cada elemento de S_1 ou S_2.
  - 1:1
  - 1:n
  - n:1
  - n:n

Entrada considerada pelo casador

- Além dos esquemas propriamente ditos (e instâncias) casadores podem receber como entrada outros dados:
  - dicionários,
  - ontologias,
  - casamentos anteriores,
  - ...
  - entrada de um usuário.

Combinando casadores

- Para a maioria das aplicações, um casador somente não produz resultados adequados.

- Vários casadores são combinados:

  - Casador **híbrido**: um casador é construído combinando as técnicas de vários casadores simples em um novo casador (uma única passada nos dados)

  - Casador **composto**: **os resultados** de vários casadores individuais **são combinados**.

Conclusões – casadores de esquemas

- Há um grande número de casadores propostos.
- Não há soluções genéricas.
- Casamento de qualidade provavelmente envolve intervenção humana (pode haver aprendizado)