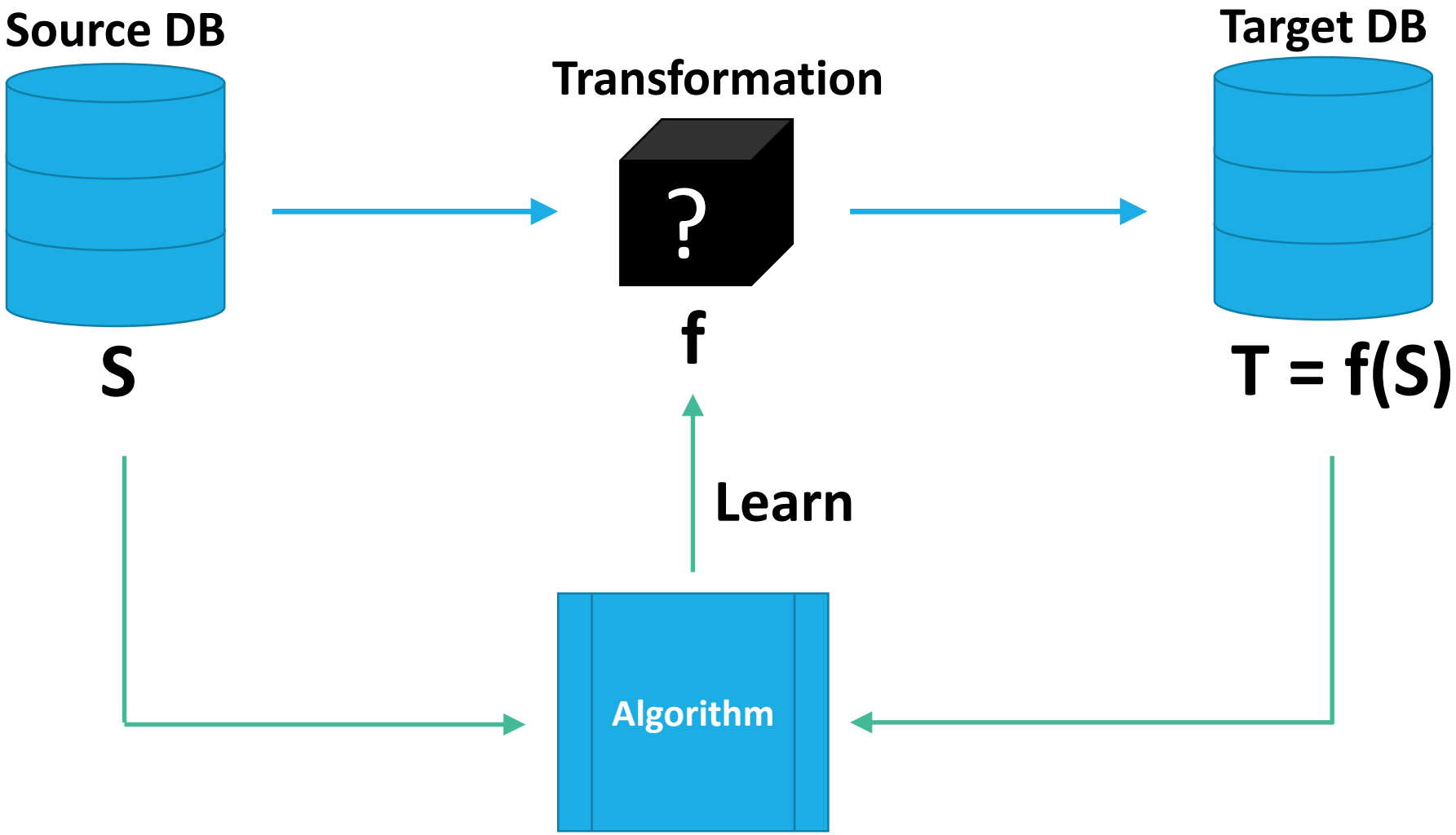
Schema Mapping Discovery From Example Data Using ILP

Manuel Fink and Heiner Stuckenschmidt
Data and Web Science Group
University of Mannheim

Problem Description



Example 1 (**GAV** Mappings)

S

Vegetable

ID	Name	NID
017	Mushroom (white)	11
032	Mushroom (brown)	11
067	Potato	24

AnimalProduct

ID	Name	Type	NID
7254	Beef (Minced)	meat	34
8696	Chicken (Breast)	meat	65
8920	Egg (Chicken)	egg	67

NutritionalValue

ID	KCal	Protein	CarboHydrates	Fat
11	16	2.7g	0.6g	0.2g
24	76	1.9g	15.6g	0g
34	208	20.5g	0g	14g
65	102	23g	0g	0.7g
67	137	11.9g	1.5g	9.3g

T

VegetarianIngredient

ID	Name	E	C	F	P
017	Mushroom (white)	16	0.6g	0.2g	2.7g
032	Mushroom (brown)	16	0.6g	0.2g	2.7g
067	Potato	76	15.6g	0g	1.9g
8920	Egg (Chicken)	137	1.5g	9.3g	11.9g

$$\text{Vegetable}(ID, Name, NID) \wedge \text{NutrionalValue}(NID, E, P, C, F) \rightarrow \text{VegetarianIngredient}(ID, Name, E, C, F, P)$$

$$\text{AnimalProduct}(ID, Name, \mathbf{egg}, NID) \wedge \text{NutrionalValue}(NID, E, P, C, F) \rightarrow \text{VegetarianIngredient}(ID, Name, E, C, F, P)$$

Problem In ILP Framework

Components

- Background Knowledge *B*: Tuples from relations of source DB
  - (Constraints like Key Relationships, Column Data Types/Domain)
- Positive Examples E^+: Tuples from relation(s) in target DB
- Negative Examples E^-: Any Tuples not in relations of target DB (CWA)
- Hypothesis H: Set of Schema Mapping Rules

B = {*Vegetable(„017“, „Mushroom (white)“, „11“),*
NutritionalValue(„11“, „16“, „2.7g“, „0.2g“, „2.7g“), …}

$\mathbf{E^+}$ = {*VegetableIngredient(„017“, „Mushroom (white)“, „16“, „0.6g“, „0.2g“, „2.7g“),* …}

$\mathbf{E^-}$ = {*VegetableIngredient(„017“, „Beef (minced)“, „208“, „20.5g“, „0g“, „14g“),*
VegetableIngredient(„032“, „Mushroom (white)“, „16“, „0.6g“, „0.2g“, „2.7g“), …}

Problem In ILP Framework

Components

- Background Knowledge *B*: Tuples from relations of source DB
- Positive Examples E^+: Tuples from relation(s) in target DB
- Negative Examples E^-: Any Tuples not in relations of target DB
- Hypothesis H: Set of Schema Mapping Rules

B = {*Vegetable(„017“, „Mushroom (white)“, „11“), NutritionalValue(„11“, „16“, „2.7g“, „0.2g“, „2.7g“), …*}

E^+ = {*VegetableIngredient(„017“, „Mushroom (white)“, „16“, „0.6g“, „0.2g“, „2.7g“), …*}

E^- = {*VegetableIngredient(„017“, „Beef (minced)“, „208“, „20.5g“, „0g“, „14g“), VegetableIngredient(„032“, „Mushroom (white)“, „16“, „0.6g“, „0.2g“, „2.7g“), …*}

Requirements

- ✓ Prior Satisfiability: $B \wedge E^- \not\models$ false
- ✓ Prior Necessity: B $\not\models E^+$

Solution Constraints

- ✓ Posterior Sufficiency: $B \wedge$ H $\models E^+$
  (Produce *all* target tuples)
- ✓ Posterior Satisfiability: $B \wedge$ H $\wedge E^- \not\models$ false
  (Produce *only* target tuples)

Problem In ILP Framework

Components

- Background Knowledge *B*: Tuples from relations of source DB
- Positive Examples E^+: Tuples from relation(s) in target DB
- Negative Examples E^-: Any Tuples not in relations of target DB
- Hypothesis H: Set of Schema Mapping Rules

Language Bias

- GLAV (Global-And-Local-As-View) Schema Mappings (Source-to-Target Tuple-Generating Dependencies)

$$\forall \bar{x}\, \varphi(\bar{x}) \;\rightarrow\; \exists \bar{y}\, \psi(\bar{x}, \bar{y})$$

Example 2 (**GLAV** Mapping)

Person

ID	Name	CityID	EmpID
0	Chris	5	13
1	Liz	4	25

City

ID	Name	LatLong
3	New York	40.7128N 74.0060W
4	Philadelphia	39.9526N 75.1652W
5	Baltimore	39.2904N 76.6122W

Employer

ID	Name	CityID
13	Comcast	4
25	HBO	3

S

Commute

ID	Employee	Start	Stop
1	Chris	1	2
2	Liz	2	3

GeoLocation

ID	LatLong
1	39.2904N 76.6122W
2	39.9526N 75.1652W
3	40.7128N 74.0060W

T

Person(ID,PName,CID1,EID) ∧ **Employer**(EID,EName,CID2
∧ **City**(CID1,CName1,Loc1) ∧ **City**(CID2,CName2,Loc2)
→ ∃ CoID,GID1,GID2 :
Commute(CoID,PName,GID1,GID2)
∧ **GeoLocation**(GID1,Loc1) ∧ **GeoLocation**(GID2,Loc2)

Interesting Problem Dimensions (Future Work)

- Learn Mappings with Constants in Body:
  - $\text{AnimalProduct}(\text{ID}, \text{Name}, \mathbf{egg}, \text{NID}) \wedge \text{NutrionalValue}(\text{NID}, \text{E}, \text{P}, \text{C}, \text{F}) \rightarrow \text{VegetarianIngredient}(\text{ID}, \text{Name}, \text{E}, \text{C}, \text{F}, \text{P})$
- Learn GLAV (not GAV) Mappings:
  - $\text{Person}(\text{ID},\text{PName},\text{CID1},\text{EID}) \wedge \text{Employer}(\text{EID},\text{EName},\text{CID2}) \wedge \text{City}(\text{CID1},\text{CName1},\text{Loc1}) \wedge \text{City}(\text{CID2},\text{CName2},\text{Loc2}) \rightarrow \exists\ \text{CoID},\text{GID1},\text{GID2}: \text{Commute}(\text{CoID},\text{PName},\text{GID1},\text{GID2}) \wedge\ \text{GeoLocation}(\text{GID1},\text{Loc1}) \wedge \text{GeoLocation}(\text{GID2},\text{Loc2})$
- Learn Mappings with Functions:
  - $\text{Vegetable}(\text{ID}, \text{Name}, \text{NID}) \wedge \text{NutrionalValue}(\text{NID}, \text{E}, \text{P}, \text{C}, \text{F}) \rightarrow \text{VegetarianIngredient}(\text{ID}, \text{Name}, \mathbf{f}(\mathbf{E}), \text{C}, \text{F}, \text{P}),\ f: energy \mapsto energy \circ\ 'kcal'$

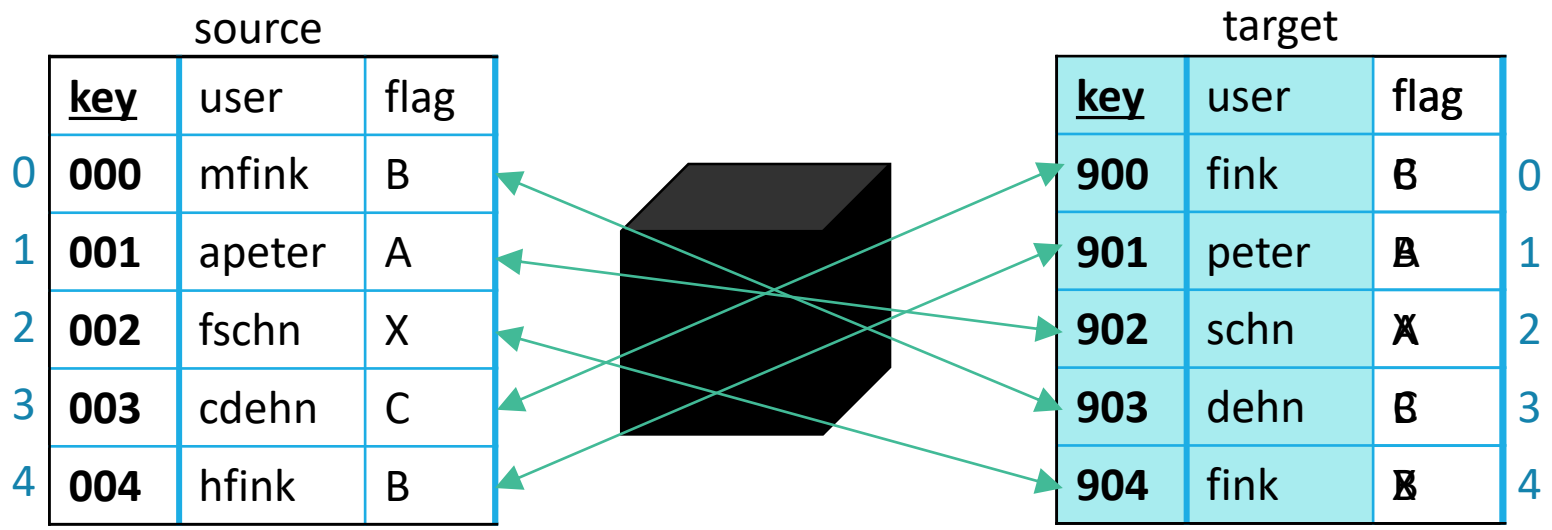
Challenges and Aspects

- Scalability
  - At least O(100) different Tables on both sides
  - Broad Tables/Predicates (e.g. 80 Columns…)
  - Long Tables (GigaBytes of Data)
- Instance Ambiguity
  - Instances are String/Number constants
- Language Bias
  - Atypical for ILP
    - Existential Quantifier in Head
    - Multiple Predicates in Head
    - Functions?

Work So Far

Variation of Problem

- Fixed Schema
  - $\forall x_1 \dots x_n$: $T^{Source}(x_1, \dots, x_n) \mapsto T^{Target}(f_1(x_1), \dots, f_n(x_n))$
- Transformation Functions on Values (Prefixing, Scaling, Trimming etc.)



Work So Far

Variation of Problem

- Fixed Schema
- Offset Columns
  (Violation of First Normal-Form)

Table^{Source}

ID	TABNAME	TABKEY	FLAG
38500	VBAP	8000000001908000001	V
42100	VBAP	8000000001908000003	V
1780	MARA	8000000054765	X
179	MARA	80000000000000000015	Y

Table^{Target}

ID	TABNAME	TABKEY	FLAG
385	VBAP	9047700001908000001	V
421	VBAP	9047700001908000003	V
178	MARA	90499000054675	X
179	MARA	90400000000000000015	Y

Domain Rules:

[MANDT] 800 -> 904
[VBELN] .{2}x -> 77y
[KUNNR] .{2}x -> 99x

Field Rules:

Table.ID x{0}* -> x

Offset Rules:

Table.TABKEY[TABNAME/FLAG]
VBAP / V:
MANDT[0-2]|VBELN[3-12]
MARA / X:
MANDT[0-2]|KUNNR[3-12]
MARA / X:
MANDT[0-2]

Work So Far

Data Set

- Artifical Transformation
  - 2 different Transformations defined on 12 domains
- 168MB
- 84 Tables with up to 28000 lines
- Solvable in ~1 Minute
  - 26 Domain rules
    - 14 Key Mappings
    - 4 Fixed Value Transformations
    - 1 Trimming Transformation
    - 7 Mask Overlay Transformations
  - 1 Field rule
    - Disable domain rule
  - 5 Offset Rules
    - 1 Control Column
    - Up to 4 different Segmentations with up to 2 transformed domains

Work So Far

Example Solution – Domain Rules:

KNUMB:	IdTableTransformation	
KUNNR:	IntegerAdditionTransformation.{NUMERIC} :	x -> [number + 9900000000]
TDOBNAME:	FrontMaskTransformation.{NUMERIC} :	xxy" -> "99y"
ATINN:	IdTableTransformation	
CLINT:	IdTableTransformation	
ADRNR:	IdTableTransformation	
SYBIN2:	IdTableTransformation	
SYBIN1:	FixedValueTransformation.{NUMERIC,TEXT} :	x -> "0"
USNAM:	FixedValueTransformation.{NUMERIC,TEXT} :	x -> "ANONYM"
J_OBJNR:	FrontMaskTransformation.{NUMERIC,TEXT} :	xxxxy" -> "xx99y„
AD_SO_KEY:	IdTableTransformation	
FPLNR:	IdTableTransformation	

Work So Far

Example Solution – Domain Rules (cont):

INT2:	RearTrimTransformation.{NUMERIC,TEXT} :	xxx[.]{3} -> xxx
AD_ADDRNUM:	IdTableTransformation	
VBELN:	FrontMaskTransformation.{NUMERIC} :	xxy" -> "99y"
PARNR:	IdTableTransformation	
CK_KALNR:	IdTableTransformation	
PERNR:	IdTableTransformation	
XUBNAME:	FixedValueTransformation.{NUMERIC,TEXT} :	x -> "ANONYM"
MANDT:	FixedValueTransformation.{NUMERIC,TEXT} :	x -> "904"
NA_OBJKEY:	IntegerAdditionTransformation.{NUMERIC} :	x -> [number + 9900000000]
AD_PERSNUM:	IdTableTransformation	
QMNUM:	IdTableTransformation	
NA_PARNR:	IntegerAdditionTransformation.{NUMERIC} :	x -> [number + 9900000000]
KUNDE:	IntegerAdditionTransformation.{NUMERIC} :	x -> [number + 9900000000]

Work So Far

Example Solution (Offset Rules)

- CDHDR.OBJECTID:

[OBJECTCLAS] -> OBJECTID:

[DEBI]	KUNNR[0-10) | UnchangesSegments(11,..)
[VERKBELEG]	VBELN[0-10) | UnchangesSegments(11,..)
[BANK]	MANDT[0-3) | UnchangesSegments(4,..)
[MATERIAL]	UnchangedSegments

Questions / Ideas?