Contents

01 Introduction
02 Automatic Data Warehouse Design
03 Data Warehouse Merging
04 Data Warehouse Imputation
05 Implementation
06 Conclusion

01 Introduction

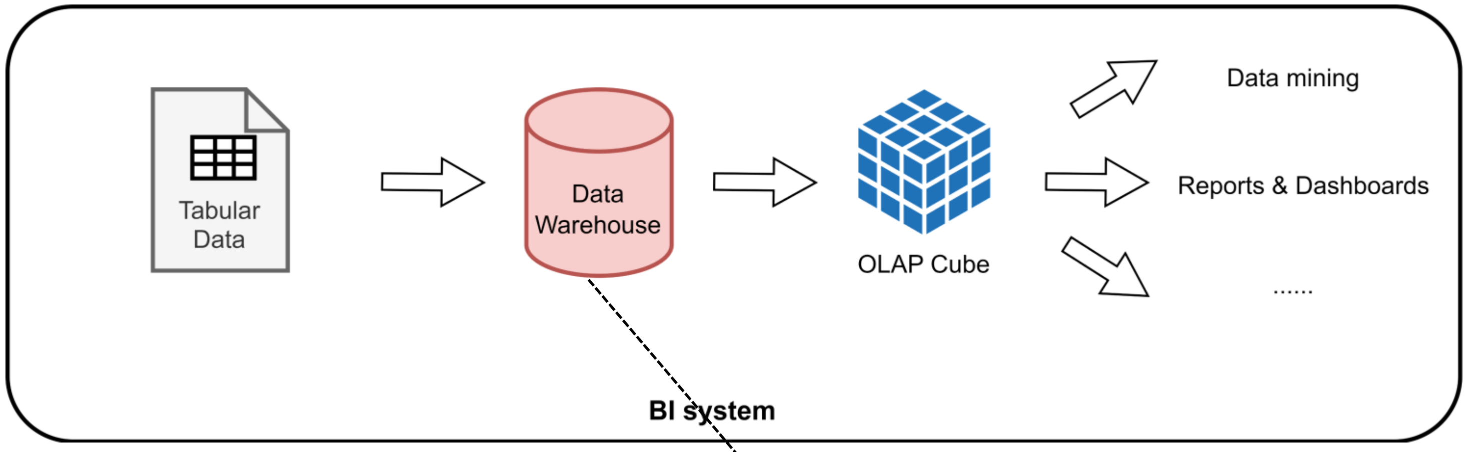
Research Context

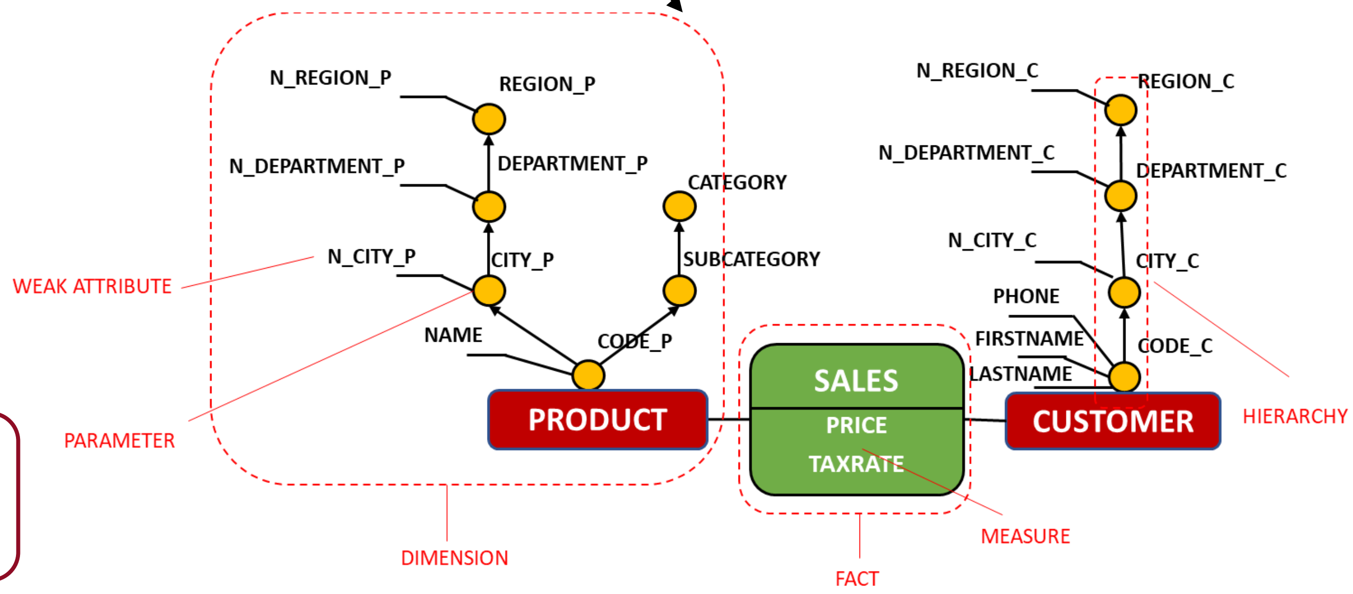
Large Companies ✓

Small Entities ✗ Lack of budget and experts ✓

BI4people (Business intelligence for the people)

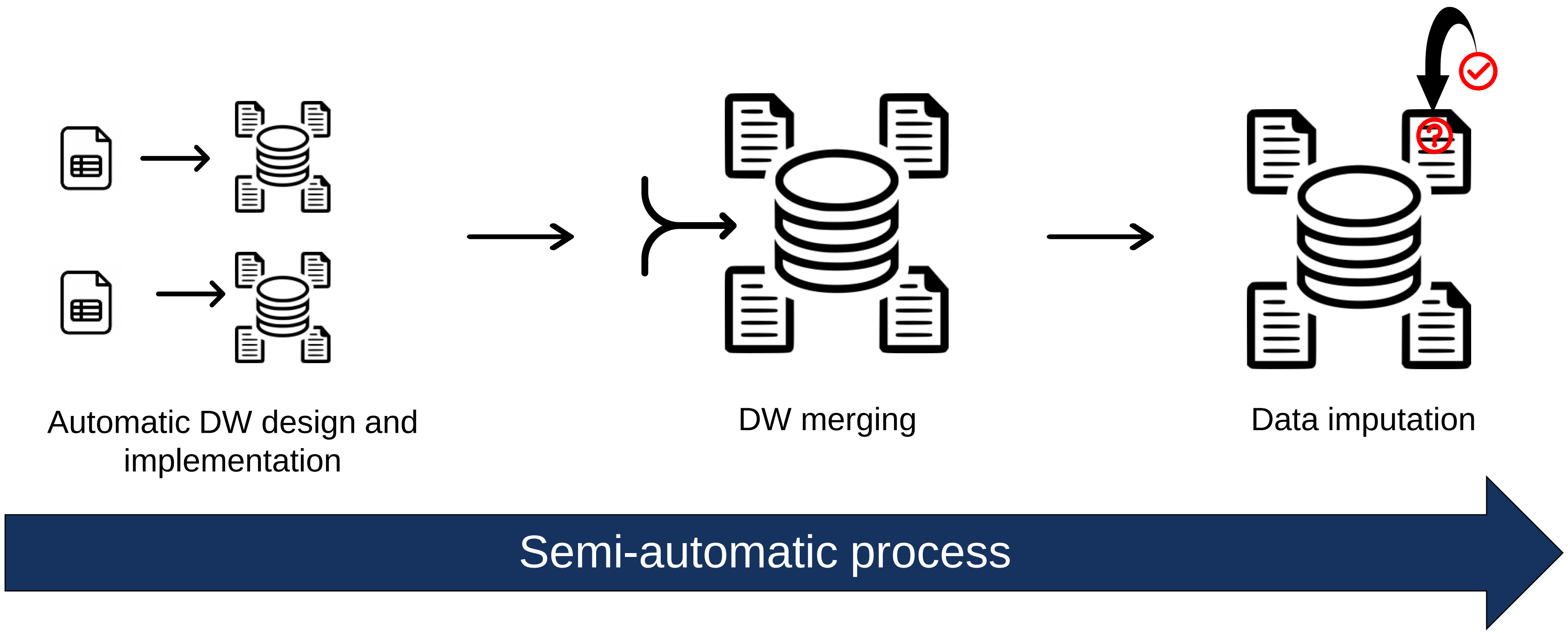
How can we automatically integrate tabular data integration in multidimensional data warehouses?





01 Introduction

Solution



Contents

01 Introduction

02 **Automatic Data Warehouse Design**

03 Data Warehouse Merging

04 Data Imputation

05 Implementation

06 Conclusion

02 Automatic DW Design

Problems

How to automatically generate a DW from tabular data?

- **Lack of schema**
- **Complex DW structure**

Data-driven Approach	Source Type	Schema
Boehnlein and Ulbrich-vom Ende (1999)	Relational database	ER/SER (Structured Entity Relationship)
Moody and Kortink (2000)		ER (Entity Relationship)
Phipps and Davis (2002)		
I.-Y.Song et al. (2007)		
Jensen et al. (2004)		Not mentionned
Elamin et al. (2017)		Logical schema
Sautot et al. (2015)	Data warehouse	Star/Constellation schema without hierarchy
Golfarelli et al. (2001)	XML file	DTD (Document Type Definition)
Vrdoljak et al. (2003)		XML schema
Ouaret et al. (2014)		
Romero and Abello (2007)	Ontology	OWL (Wel Ontology Language)
Usman et al. (2010, 2013)	Flat data	-
Sanprasit et al. (2021)		

Data source with schema : 11/13

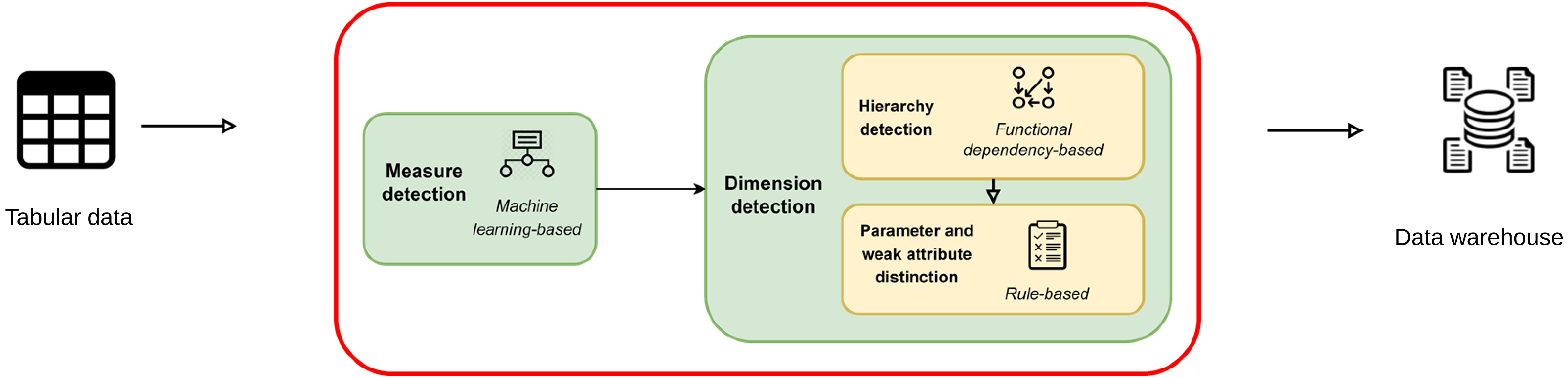
Data source without schema : 2/13

Potential incorrect hierarchy generation

Requirement of domain ontology

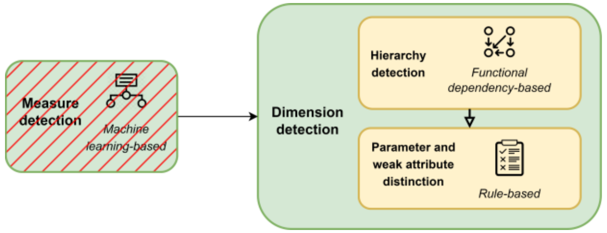
02 Automatic DW Design

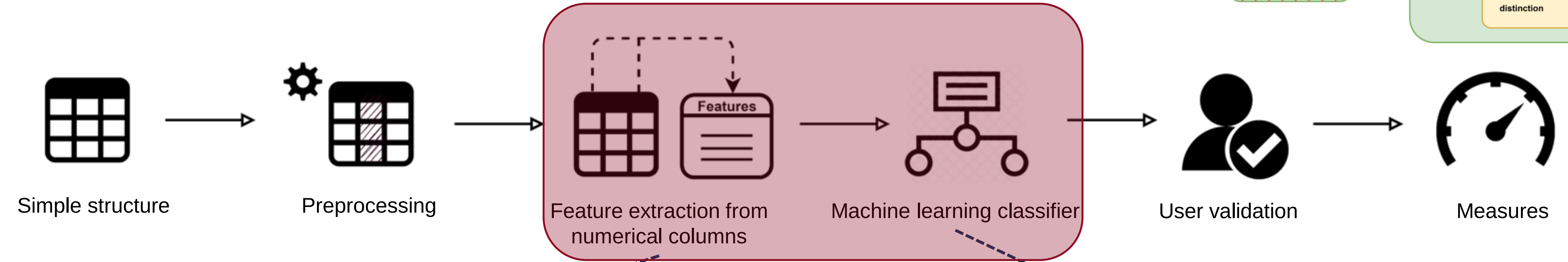
Contribution



02 Automatic DW Design

Measure Detection



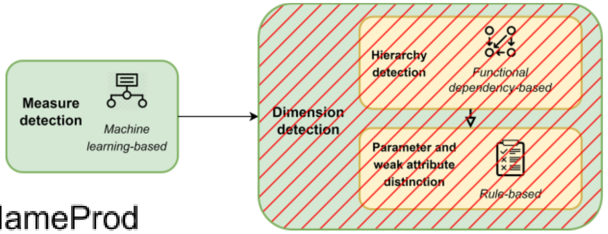


Feature Category	Feature
General feature	Data Type
	Positive/Negative/Zero value ratio
	Unique value ratio
	Same digital number
Statistical feature	Average/Minimum/Maximum/Median/Upper quartile/Lower quartile values
	Coefficient of variation
	Range ratio
Inter-column feature	Location ratio
	Numerical column ratio
	Multiple functional dependencies
	Numerical neighbor

- **Support vector machine (SVM)**
- **Decision tree (DT)**
- **Random forest (RF)**
- **K-nearest neighbors (KNN)**

02 Automatic DW Design

Dimension Detection



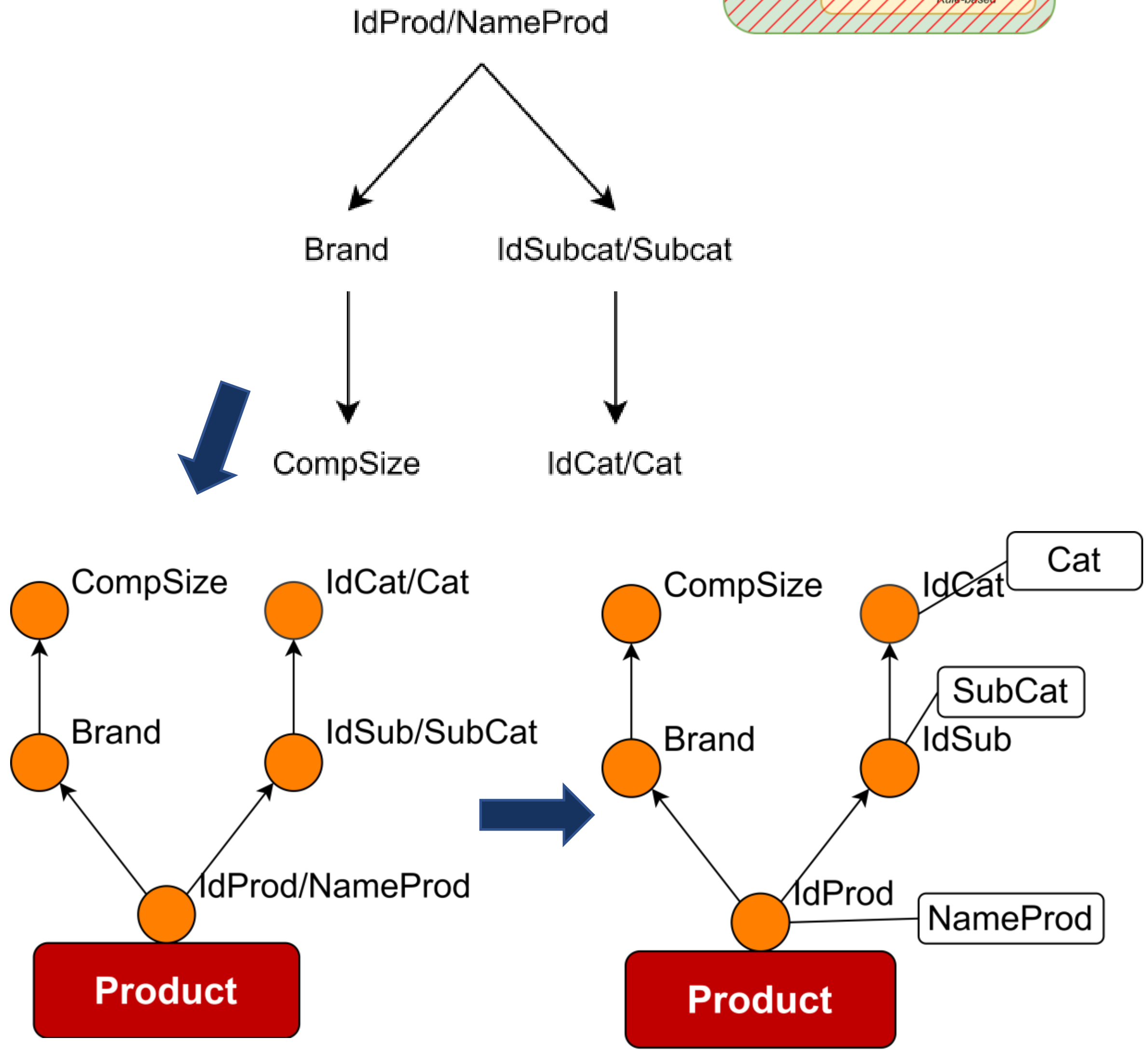
Hierarchy detection

Algo. 2 getHierarchies, p38

- Functional dependency extraction
- Transitivity and equivalent attribute processing
- Hierarchy extraction from functional dependency trees

Distinction of parameters and weak attributes

- Group of equivalent attributes – syntactic rules
- Highest level attribute – semantic rules



02 Automatic DW Design

Experimental Assessment - Measure Detection

Objectives

- What is the **effectiveness** of measure detection by **machine learning algorithms**?
- Are the proposed **features effective**?
- Is the trained **model generic**?

Datasets

- 346 tables
- 1382 numerical columns
- 9 sites of open data
- 6 domains (economy, health, government, environment society)

Metrics

- Recall (**R**) = $\frac{N_{mm}}{N_{mm} + Nmn}$
- Precision (**P**) = $\frac{N_{mm}}{N_{mm} + Nnm}$
- F-score (**F**) = $\frac{2PR}{P + R}$

Algorithms

- Support vector machine (**SVM**)
- Decision tree (**DT**)
- Random forest (**RF**)
- K-nearest neighbors (**KNN**)
- Typology-based (**TP**) (Alobaid et al.,2019)
- Functional dependency-based (**FDB**)

02 Automatic DW Design

Experimental Assessment - Measure Detection

- What is the **effectiveness** of measure detection by **machine learning algorithms**?
- Are the proposed **features effective**?
- Is the trained **model generic**?

	Baselines		Machine learning			
Metric	TP	FDB	RF	SVM	DT	KNN
R(%)	80.05	75.43	**96.64**	94.77	94.08	90.16
P(%)	73.57	77.50	**90.89**	78.44	88.44	87.61
F(%)	76.67	76.45	**93.65**	85.76	91.12	88.78

17.2%

ML Algorithms	Metrics	GE	ST	IC	GE+ST	GE+IC	ST+IC	ALL
RF	R(%)	88.10	94.27	92.68	95.30	93.67	91.93	**96.64**
	P(%)	83.59	86.28	80.91	88.21	86.13	91.14	**90.89**
	F(%)	85.69	90.01	86.37	91.57	89.67	91.50	**93.65**
SVM	R(%)	92.20	93.96	88.89	94.07	92.86	93.70	**94.77**
	P(%)	74.45	76.80	75.47	76.85	76.90	76.71	**78.44**
	F(%)	82.32	84.35	81.63	84.45	84.47	84.23	**85.76**
DT	R(%)	89.05	89.16	89.90	89.97	88.47	89.12	**91.20**
	P(%)	78.53	86.24	83.62	**89.22**	88.26	87.15	89.17
	F(%)	83.29	87.59	86.54	89.55	88.28	88.07	**90.12**
KNN	R(%)	84.13	91.95	92.07	85.56	**92.57**	92.08	90.16
	P(%)	83.73	82.45	81.48	86.06	84.14	83.65	**87.61**
	F(%)	83.82	86.90	86.42	85.68	88.11	87.59	**88.78**

Random forest

- Best F-score
- Stable distribution

Each feature category is useful

Each feature has a contribution

Generic

- Source
- Domain

02 Automatic DW Design

Experimental Assessment - Dimension Detection

- What is the **effectiveness** of the dimension detection approach?

Dataset	Element	Precision (%)	Recall (%)	F-score (%)
Example	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
	Attribute (D2)	100.00	100.00	100.00
	Attribute (D3)	100.00	100.00	100.00
Sales1	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
Sales2	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
	Attribute (D2)	100.00	100.00	100.00
	Attribute (D3)	100.00	100.00	100.00
	Attribute (D4)	100.00	100.00	100.00
	Attribute (D5)	100.00	100.00	100.00
DevApp	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
Countries	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
Covid	Dimension ID	100.00	100.00	100.00
	Attribute (D1)	100.00	100.00	100.00
	Attribute (D2)	100.00	100.00	100.00

Correctly detect dimensions

Dataset	Element	Precision (%)	Recall (%)	F-score (%)
Example	Hierarchical	100.00	100.00	100.00
	Same level	100.00	100.00	100.00
Sales1	Hierarchical	100.00	100.00	100.00
	Same level	50.00	66.67	57.14
Sales2	Hierarchical	87.50	100.00	93.33
	Same level	66.67	80.00	72.73
DevApp	Hierarchical	83.33	100.00	90.91
	Same level	100.00	83.33	90.91
Countries	Hierarchical	100.00	100.00	100.00
	Same level	100.00	100.00	100.00
Covid	Hierarchical	57.14	80.00	66.67
	Same level	100.00	75.00	85.71

Correctly detect most hierarchies

Contents

01 Introduction

02 Automatic Data Warehouse Design

03 **Data Warehouse Merging**

04 Data Imputation

05 Implementation

06 Conclusion

03 DW Merging

Problems

How to merge two DWs having commun elements?

- **Merging at both schema and instance levels**
- **Generation of different types of schema (star, constellation)**

DW matching ✓

DW merging

	Merging Level		Schema Type	Multidimensional Element			
	Schema	Instance	Output	Fact	Dimension	Hierarchy	Weak Attribute
Feki et al. (2005)	√	-	UML class diagram	√	√	√	√
Torlone (2008)	√	√	Constallation schema	-	√	√	-
Kwakye et al. (2013)	√	√	Star schema	√	√	-	-
Olaru and Vincini (2014)	√	√	Star schema dimension	-	√	√	-

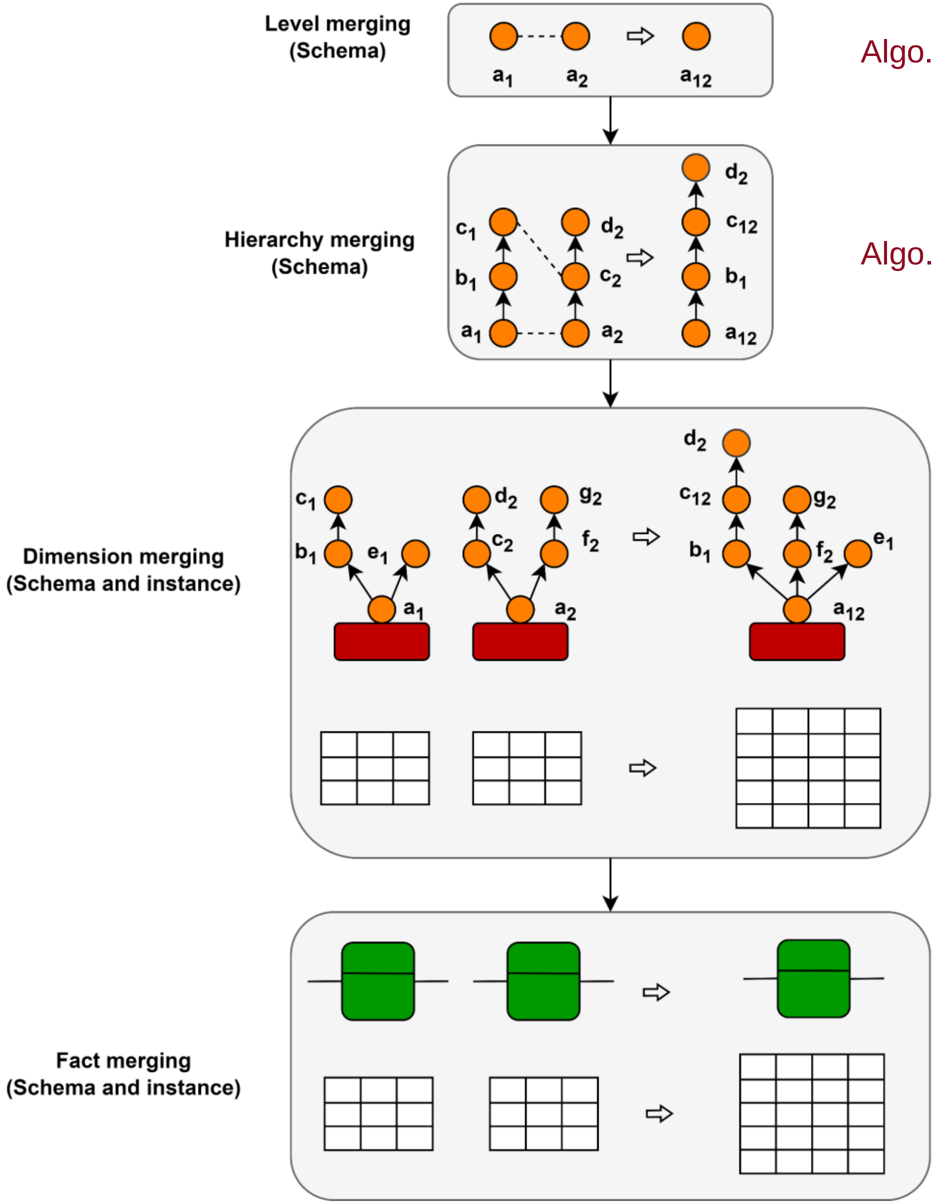
Not all merging levels

Single schema type output

Not all multidimensional elements

03 DW Merging

Contribution



Algo. 5 mergeHierarchies, p68

Algo. 5 mergeHierarchies, p68

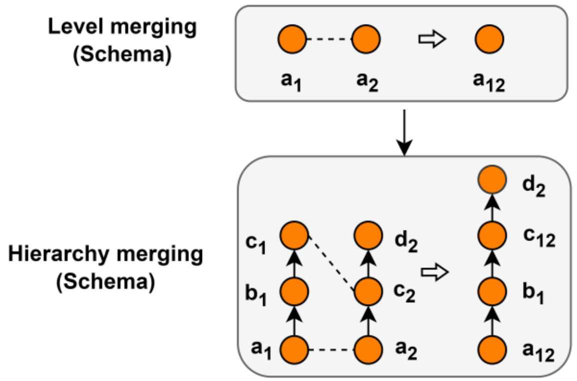
Algo. 6 mergeDimensions, p73

Algo. 8 mergeStars, p78

03

DW Merging

Level Merging



Algo. 5 mergeHierarchies, p68

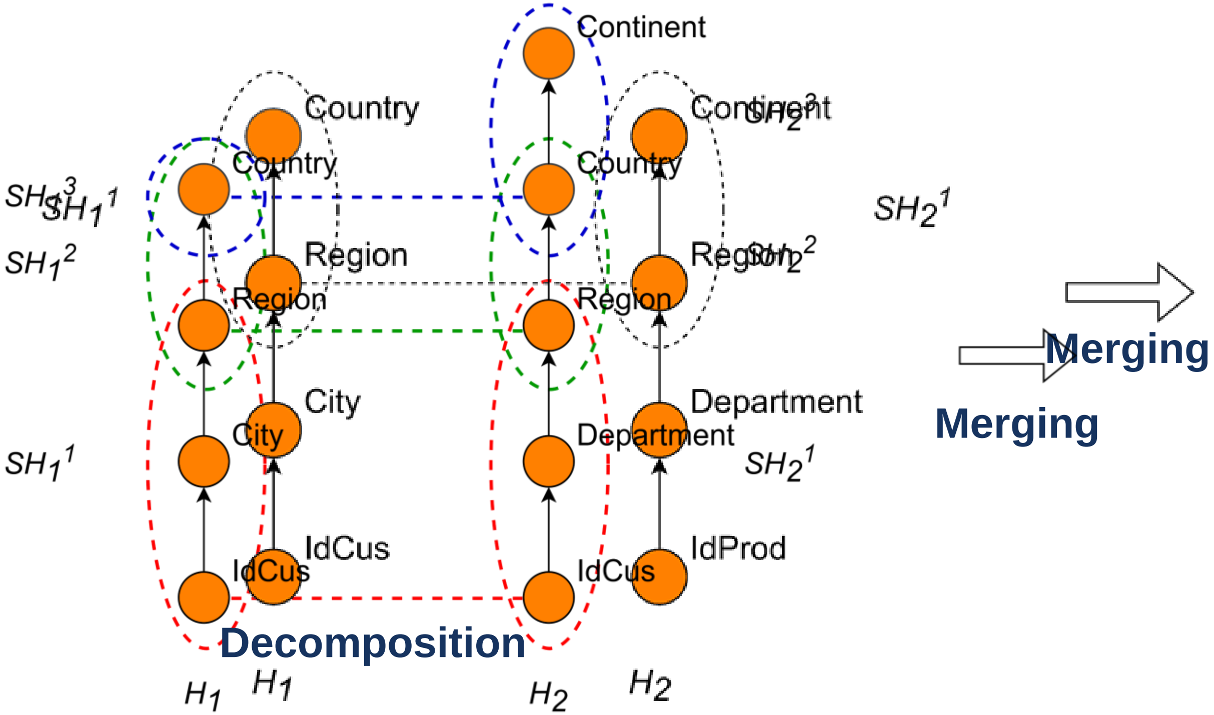
Algo. 5 mergeHierarchies, p68

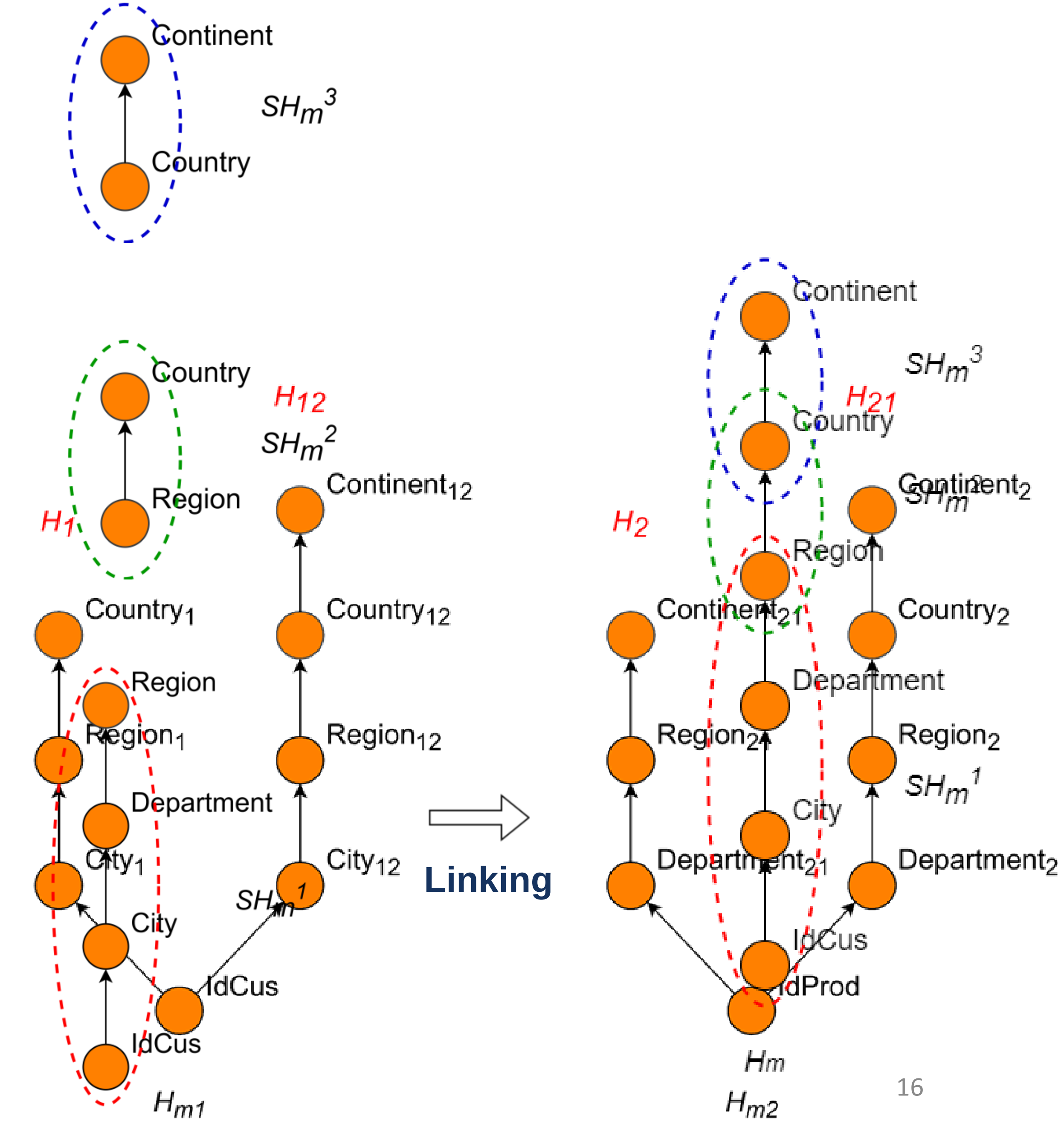
03 DW Merging

Hierarchy Merging

Hierarchy merging

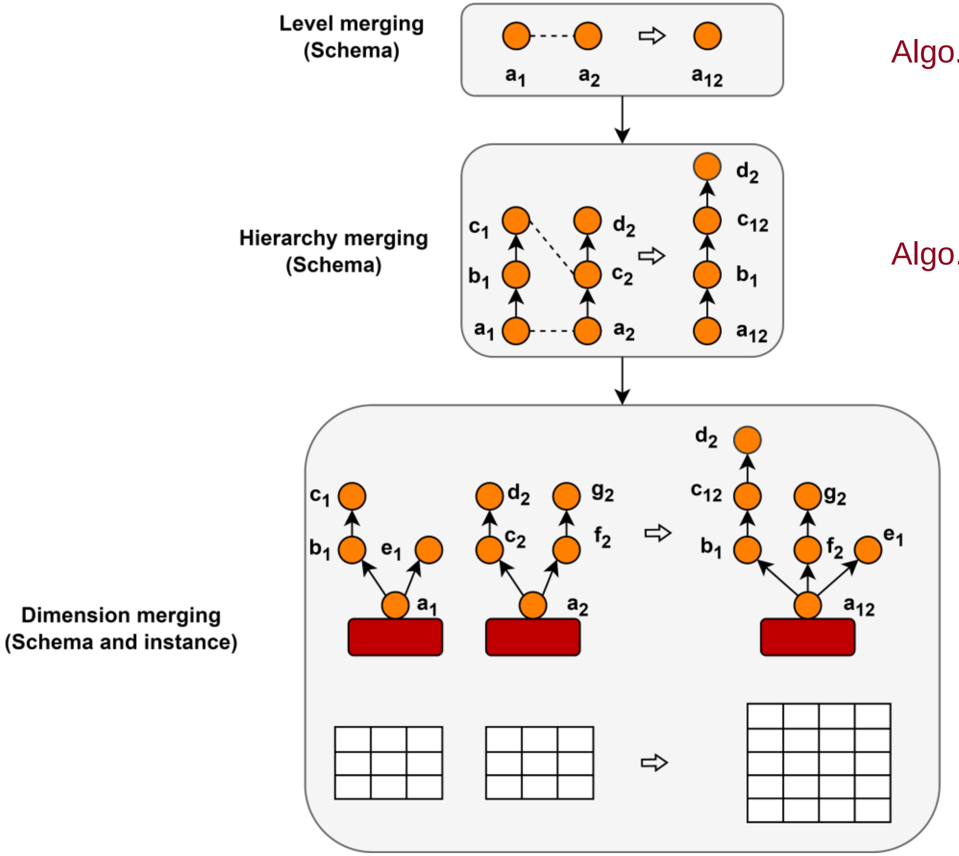
- Decompose hierarchies into sub-hierarchies
- Merging sub-hierarchies by functional dependencies
- Linking merged sub-hierarchies





03 DW Merging

Dimension Merging



Algo. 5 mergeHierarchies, p68

Algo. 5 mergeHierarchies, p68

Algo. 6 mergeDimensions, p73

03 DW Merging

Dimension Merging

Dimension merging

- Merging all hierarchies of two dimensions

IdCus	NameCus	Age	Email	City	Region	Country	MemLevel
C1	N1	34	C1@e.com	CT1	R1	CN1	L1
C2	N2	53	C2@e.com	CT2	R1	CN1	L2
C3	N3	66	C3@e.com	CT2	R1	CN1	L1
C4	N1	26	C4@e.com	CT3	R3	CN2	L1
C5	N5	45	C5@e.com	CT5	R2	CN1	L3
C6	N6	32	C6@e.com	CT4	R3	CN2	L2
C7	N7	41	C7@e.com	CT3	R3	CN2	L3

D_1

IdCus	NameCus	Phone	Department	Region	Country	Population	Continent	MemSubLevel
C1	N1	012345	D1	R1	CN1	1000000	CTN1	SL1
C2	N2	123456	D3	R1	CN1	1000000	CTN1	SL3
C3	N3	234567	D3	R1	CN1	1000000	CTN1	SL1
C4	N1	345678	D4	R3	CN2	1200000	CTN1	SL2
C6	N6	456789	D4	R3	CN2	1200000	CTN1	SL3
C8	N8	567890	D2	R2	CN1	1000000	CTN1	SL4
C9	N9	678901	D5	R4	CN3	500000	CTN2	SL5

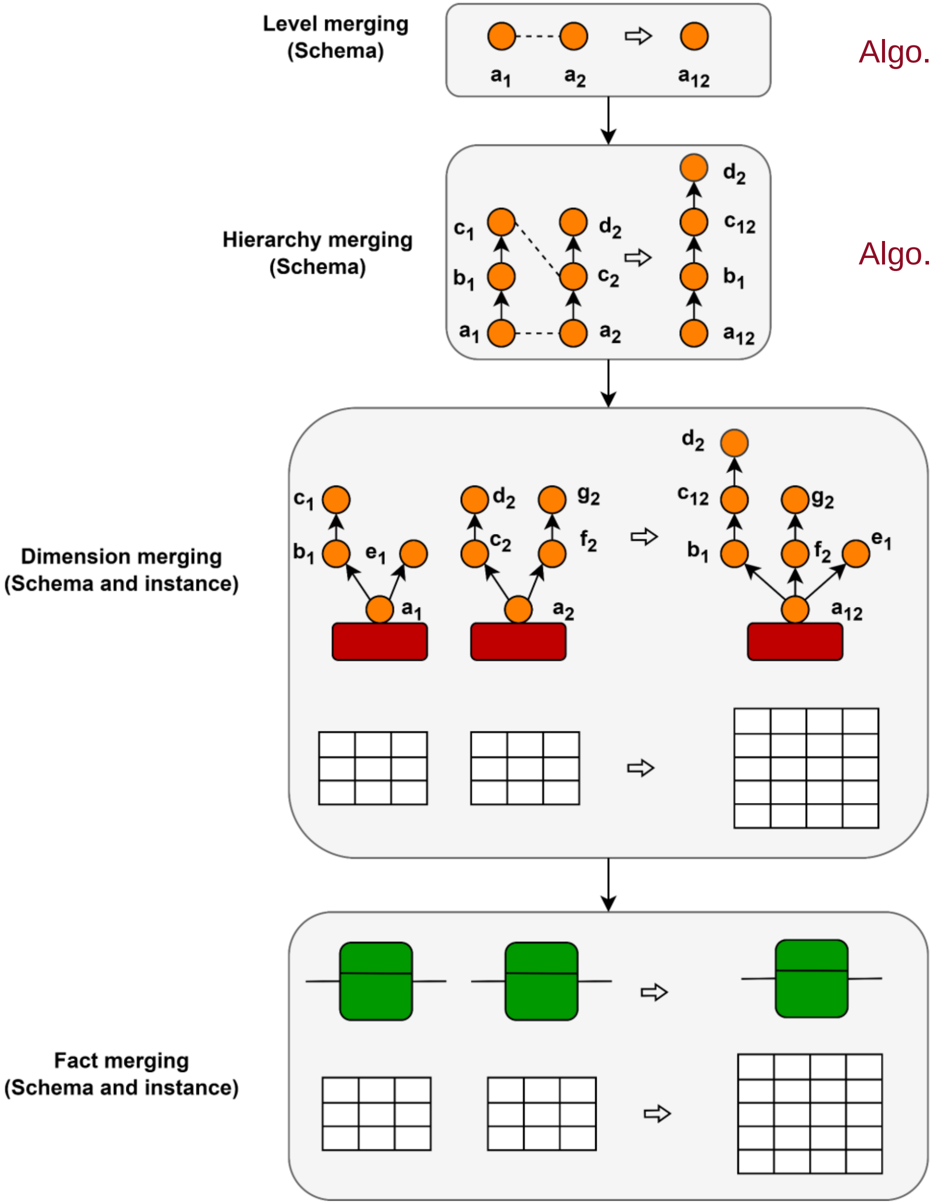
D_2

	IdCus	NameCus	Age	Email	Phone	City	Department	Region	Country	Population	Continent	MemSubLevel	MemLevel
$D_1 + D_2$	C1	N1	34	C1@e.com	012345	CT1	D1	R1	CN1	1000000	CTN1	SL1	L1
	C2	N2	53	C2@e.com	123456	CT2	D3	R1	CN1	1000000	CTN1	SL3	L2
	C3	N3	66	C3@e.com	234567	CT2	D3	R1	CN1	1000000	CTN1	SL1	L1
	C4	N1	26	C4@e.com	345678	CT3	D4	R3	CN2	1200000	CTN1	SL2	L1
D_1	C5	N5	45	C5@e.com	NULL	CT5	NULL	R2	CN1	NULL	NULL	NULL	L3
$D_1 + D_2$	C6	N6	32	C6@e.com	456789	CT4	D4	R3	CN2	1200000	CTN1	SL3	L2
D_1	C7	N7	41	C7@e.com	NULL	CT3	NULL	R3	CN2	NULL	NULL	NULL	L3
D_2	C8	N8	NULL	NULL	567890	NULL	D2	R2	CN1	1000000	CTN1	SL4	NULL
	C9	N9	NULL	NULL	678901	NULL	D5	R4	CN3	500000	CTN2	SL5	NULL

D_{12}

03 DW Merging

Fact Merging



Algo. 5 mergeHierarchies, p68

Algo. 5 mergeHierarchies, p68

Algo. 6 mergeDimensions, p73

Algo. 8 mergeStars, p78

Objectives

- Is the process able to merge DWs at the **schema level**?
- Is the process able to merge DWs at the **instance level**?
- Is the process able to generate a **star or constellation schema**?

Datasets

- TPC-H benchmark data
- 100M data files

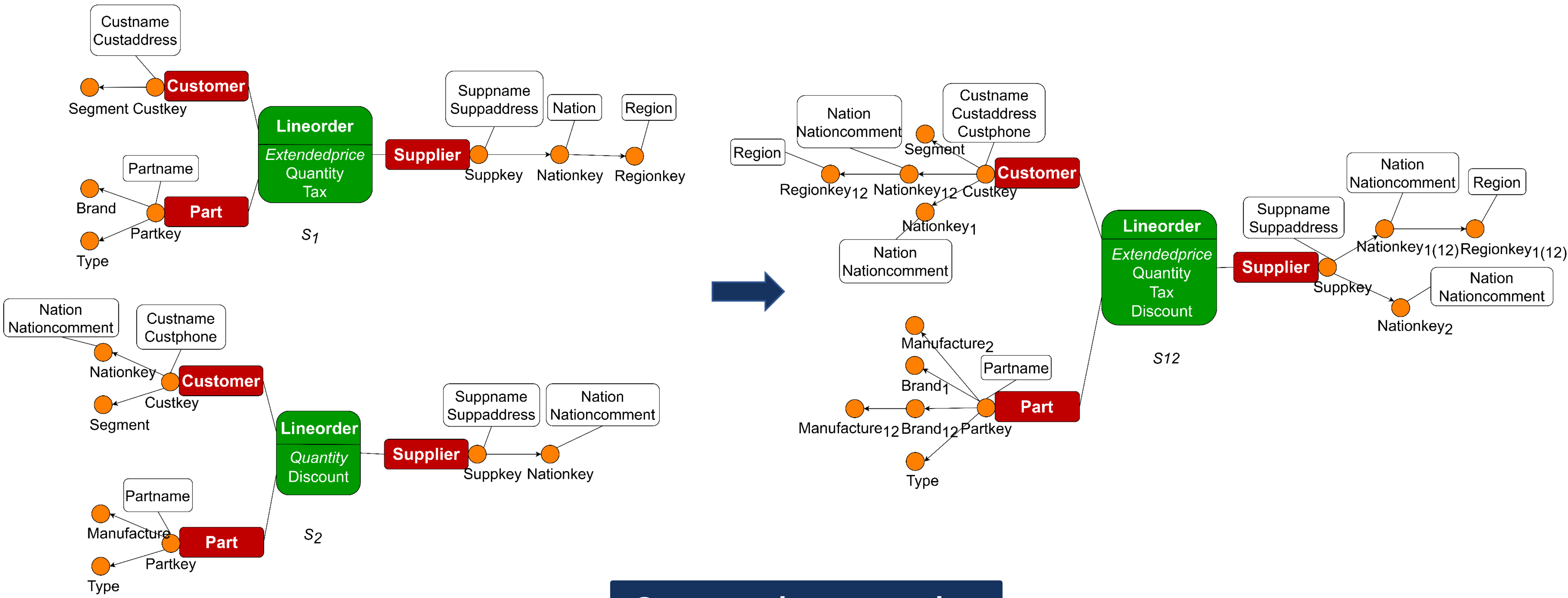
Strategy

- Two cases (star, constellation)
- Create two DWs for each case
- Select randomly ¾ of the data

03 DW Merging

Experimental Assessment – Star Schema Generation

- Is the process able to merge DWs at the **schema level**?
- Is the process able to generate a **star** or constellation **schema**?



Correct schema merging

03 DW Merging

Experimental Assessment – Star Schema Generation

Is the process able to merge DWs at the **instance level**?

Dimension /Fact	Attribute	N_1	N_2	$N_\cap$	N_m
Customer	**Custkey**	11250	11250	8426	14074
	Custname	11250	11250		14074
	Custaddress	11250	0		11250
	Custphone	0	11250		11250
	Nationkey	0	11250		11250
	Nation	0	11250		11250
	Nationcomment	0	11250		11250
	Segment	11250	11250		14074
Supplier	**Suppkey**	750	750	570	930
	Suppname	750	750		930
	Suppaddress	750	750		930
	Nationkey	750	750		930
	Nation	750	750		930
	Nationcomment	0	750		750
	Regionkey	750	0		750
	Region	750	0		750
Part	**Partykey**	15000	15000	11215	18785
	Partname	15000	15000		18785
	Brand	15000	0		15000
	Manufacture	0	15000		15000
	Type	15000	15000		18785
Lineorder	**Custkey Partkey Suppkey**	253423	253782	107736	399469
	Quantity	253423	253782		399469
	Extendedprice	253423	0		253423
	Tax	253423	0		253423
	Discount	0	253782		253782

Correct instance merging

Keys: $N_m = N_1 + N_2 - N_\cap$

Commun attributes: $N_m = N_m$ of the key

Distinct attributes: $N_m = N_{1/2}$

N_1: number of values in DW_1
N_2: number of values in DW_2
$N_\cap$: number of commun values between DW_1 and DW_2
N_m: number of values in the merged DW

Content

01 Introduction

02 Automatic Data Warehouse Design

03 Data Warehouse Merging

04 Data Imputation

05 Implementation

06 Conclusion

04 Data Imputation

Problems

How to carry out data imputation to ensure consistent analysis?

- **Dimension**
- **Categorical**

- **Deducted values**
- **Predicted values**

	Number	Categorical Data
Statistical-based	13	54%
Mode/Mean	2	100%
EM Algorithm	3	0%
Regression	5	40%
Hot/Cold Deck	3	100%
Machine Learning-based	33	36%
KNN	7	100%
Clustering	7	0%
Other supervised learning	13	31%
Neural Network	6	17%
Rule-based	8	100%
Editing Rule	1	100%
Dependency Rule	3	100%
Association Rule	4	100%
External Source-based	9	100%
Crowdsourcing	3	100%
Web Information	3	100%
Knowledge Base	3	100%
Hybrid	8	75%

Biased, based on numerical data

Appropriate sources and queries

Data Imputation

Contribution

Hie - OLAPKNN

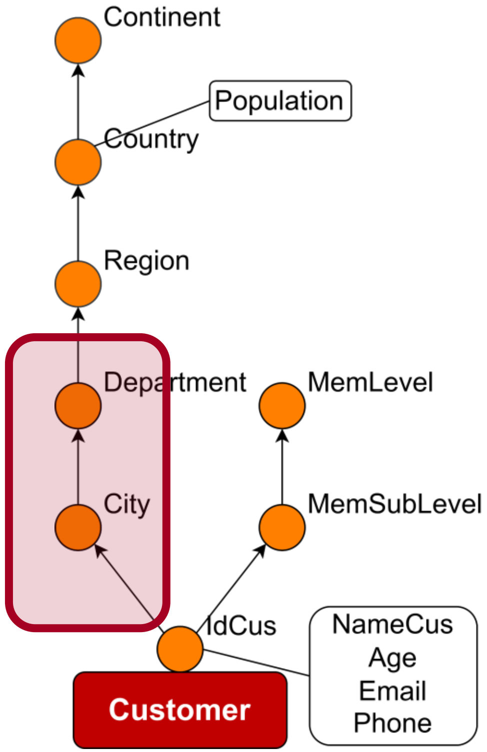
	Hierarchical imputation →	**OLAPKNN imputation**
Why?	Functional dependencies in hierarchies	Non-parametric and instance-based Suitable for different types of data Relatively high accuracy
Specifity	✓ Deducted values ✗ Limited	✓ Specific distance metric ✓ Consideration of dependencies

Data Imputation

Hierarchical Imputation

Hierarchical imputation

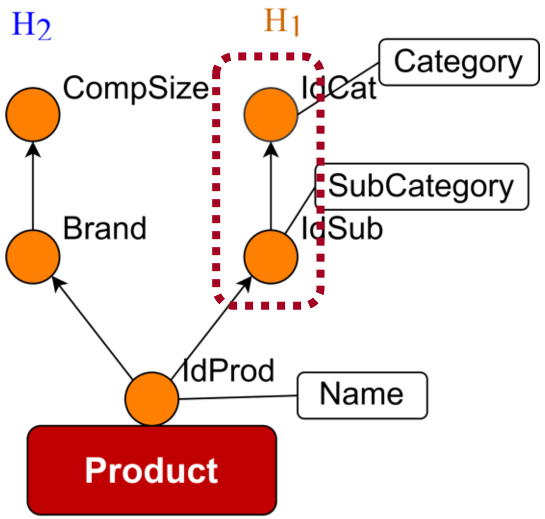
(Algo. 9 IntraImputation, p103
Algo. 10 InterImputation, p104)



IdCus	NameCus	Age	Email	Phone	City	Department	Region	Country	Population	Continent	MemSubLevel	MemLevel
C1	N1	34	C1@e.com	012345	CT1	D1	R1	CN1	1000000	CTN1	SL1	L1
C2	N2	53	C2@e.com	123456	CT2	D3	R1	CN1	1000000	CTN1	SL3	L2
C3	N3	66	C3@e.com	234567	CT2	D3	R1	CN1	1000000	CTN1	SL1	L1
C4	N1	26	C4@e.com	345678	CT3	D4	R3	CN2	1200000	CTN1	SL2	L1
C5	N5	45	C5@e.com	NULL	CT5	NULL	R2	CN1	NULL $NULL \mapsto 1000000$	NULL $NULL \mapsto CTN_1$	NULL	L3
C6	N6	32	C6@e.com	456789	CT4	D4	R3	CN2	1200000	CTN1	SL3	L2
C7	N7	41	C7@e.com	NULL	CT3	NULL $NULL \mapsto D_4$	R3	CN2	NULL $NULL \mapsto 1200000$	NULL $NULL \mapsto CTN_1$	NULL	L3
C8	N8	NULL	NULL	567890	NULL	D2	R2	CN1	1000000	CTN1	SL4	NULL
C9	N9	NULL	NULL	678901	NULL	D5	R4	CN3	500000	CTN2	SL5	NULL

04 Data Imputation

OLAPKNN



OLAPKNN

- Creation of candidate list
  (Algo. 13 getCanList, p114)
- Replaced value weight map
  (Algo. 14 getVWeightMap, p115)
- Replacement of the values
  (Algo. 15 replaceNoPlow, p116;
  Algo. 16 replacePLow, p117)

Instance	Brand	CompanySize	Id	Name	Id_Sub	Subcategory	Id_Cat	Category
i_1	Apple	Large	p1	Iphone 13	?	?	Tn	Technology
i_2	Samsung	Large	p2	Gaxaly S10	Ph	Phones	Tn	Technology
i_3	Homefine	Medium	p3	Office Table	Tb	Table	Of	Office
i_4	Homefine	Medium	p4	Computer Table	Tb	Table	Of	Office
i_5	Sumsang	Large	p5	Galaxy Book	Pc	Laptop	Tn	Technology
i6	HP	Large	p6	Envy 17	Pc	Laptop	Tn	Technology

04 Data Imputation

OLAPKNN

Distance of 4 levels

- Hierarchy level weight
  - Cardinality-based
- Hierarchy weight
  - Dependency degree

H_2 H_1

Instance	Brand	CompanySize	Id	Name	Id_Sub	Subcategory	Id_Cat	Category
i_1	Apple	Large	p1	Iphone 13	?	?	Tn	Technology
i_2	Samsung	Large	p2	Gaxaly S10	Ph	Phones	Tn	Technology
i_3	Homefine	Medium	p3	Office Table	Tb	Table	Of	Office
i_4	Homefine	Medium	p4	Computer Table	Tb	Table	Of	Office
i_5	Sumsang	Large	p5	Galaxy Book	Pc	Laptop	Tn	Technology

$\Delta(i_1,i_2)$ $\Delta p_2^{H1}(i_1,i_2)$ $\Delta p_3^{H1}(i_1,i_2)$ $\Delta_{(W1,H2)}$ $\Delta p_2^{H2}(i_1,i_2)$ $\Delta p_3^{H2}(i_1,i_2)$

Dimension Instance Distance

$\Delta(i_1,i_2) = W_h(H_1,H_2)\times\Delta_{(H1,H2)}(i_1,i_2) + W_h(W_1,H_2)\times\Delta_{(W1,H2)}(i_1,i_2) + W_h(H_1,H_1)\times\Delta_{(H2,H2)}(i_1,i_2)$

Hierarchy Instance Distance

$\Delta_{H2}(i_1,i_2) = W_l(p_2^{H2})\times\Delta p_2^{H2}(i_1,i_2) + W_l(p_3^{H2})\times\Delta p_3^{H2}(i_1,i_2))$

$\Delta_{W1}(i_1,i_2) = \Delta(i_{1,Name}, i_{2,Name})$

$\Delta_{H1}(i_1,i_2) = W_l(p_3^{H1})\times\Delta p_3^{H1}(i_1,i_2)$

Hierarchy Level Instance Distance

$\Delta p_2^{H2}(i_1,i_2)= \Delta(i_{1,Brand}, i_{2,Brand})$

$\Delta p_3^{H2}(i_1,i_2)= \Delta(i_{1,CompanySize}, i_{2,CompanySize})$

$\Delta p_3^{H1}(i_1,i_2)= (\Delta(i_{1,Id\ Cat}, i_{2,Id\ Cat}) + \Delta(i_{1,Category}, i_{2,Category}))/2$

Attribute Distance

$\Delta(i_{1,Brand}, i_{2,Brand})$ $\Delta(i_{1,CompanySize}, i_{2,CompanySize})$ $\Delta(i_{1,Name}, i_{2,Name})$ $\Delta\Delta(i_{1,Id\_Sub}, i_{2,Id\_Sub})$ $\Delta(i_{1,Subcategory}, i_{2,Subcategory})$ $\Delta\Delta(i_{1,Id\_Cat}, i_{2,Id\_Cat})$ $\Delta\Delta(i_{1,Category}, i_{2,Category})$

04 DW Imputation

Experimental Assessment

Objectives

- What is the **effectiveness** of Hie-OLAPKNN?
- What is the **efficiency** of Hie-OLAPKNN?
- If the imputed DW respect the hierarchy **strictness**?

Datasets

- 5 real-world datasets from relational dataset repository site
- Mono-attribute
- Multi-attribute
- 1%, 5%, 10%, 20%, 30%, 40%

Metrics

- Recall (**R**) = $\frac{\{Imputed\} \cap \{True]}{\{True\}}$
- Precision (**P**) = $\frac{\{Imputed\} \cap \{True]}{\{True\}}$
- F-score (**F**) = $\frac{2PR}{P + R}$
- Run time
- Strictness degree

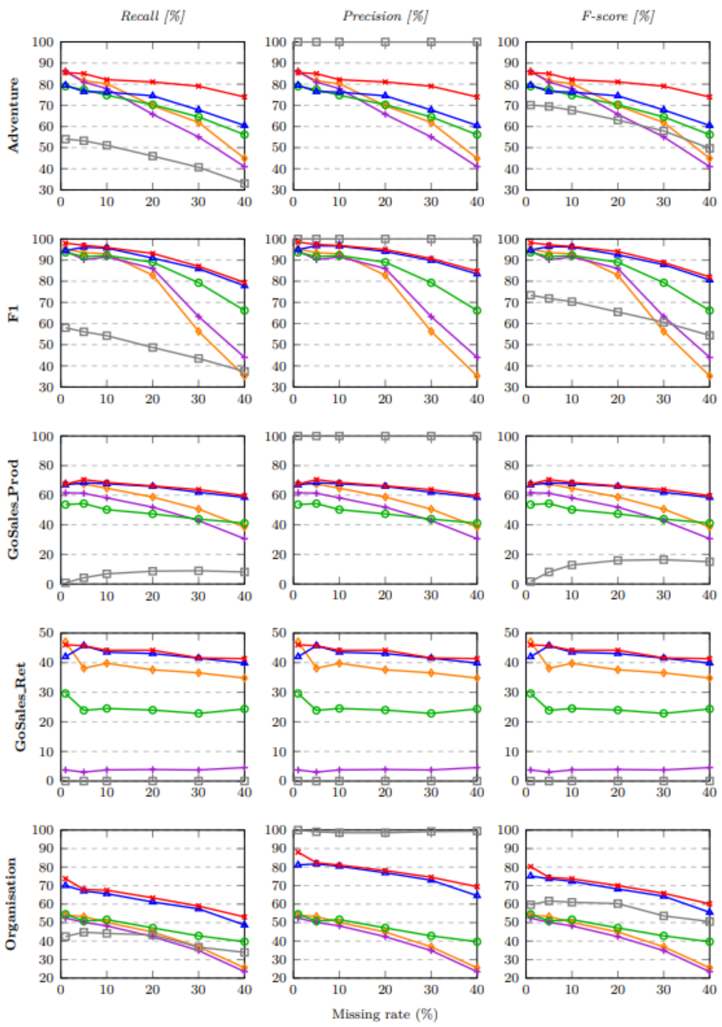
Algorithms

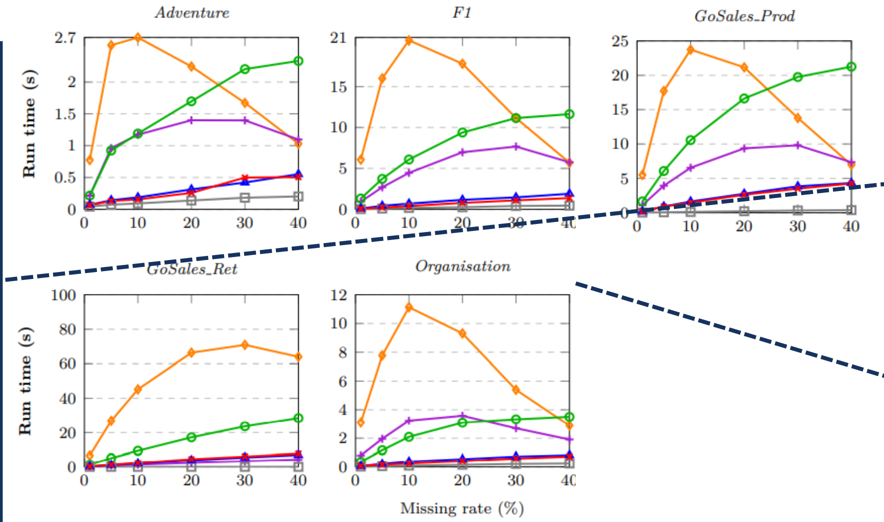
- Hie-OLAPKNN
- Hierachical imputation (**Hie**)
- OLAPKNN
- KNN (Domeniconi and Yan, 2004)
- NB (Garcia and Hruschka, 2005)
- MIBOS (Wu et al., 2012)

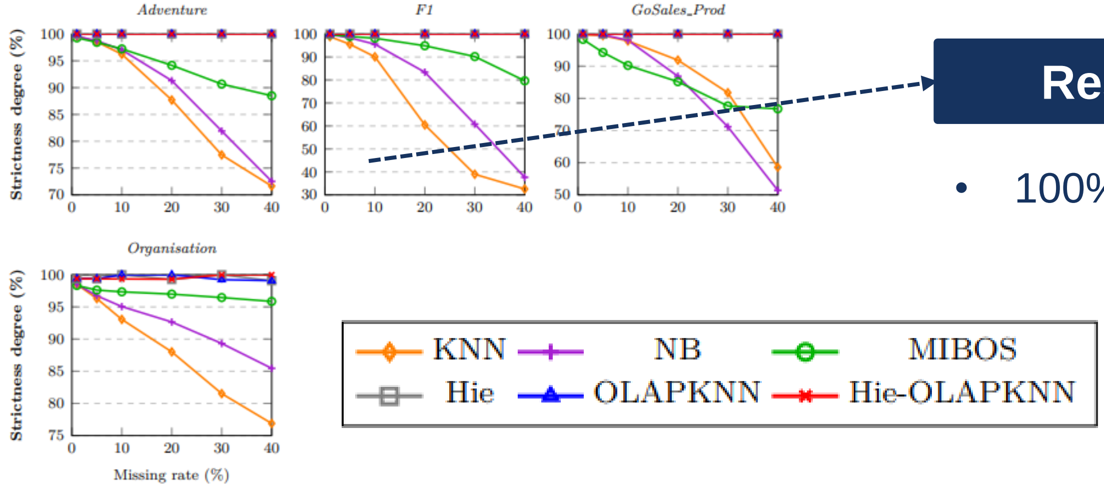
04 DW Imputation

Effectiveness Results

- What is the **effectiveness** of Hie-OLAPKNN?
- What is the **efficiency** of Hie-OLAPKNN?
- If the imputed DW respect the hierarchy **strictness**?







More effective

- + 44.8%

More efficient

- +19 times

Respect strictness

- 100%

Content

01 Introduction

02 Automatic Data Warehouse Design

03 Data Warehouse Merging

04 Data Imputation

05 **Implementation**

06 Conclusion

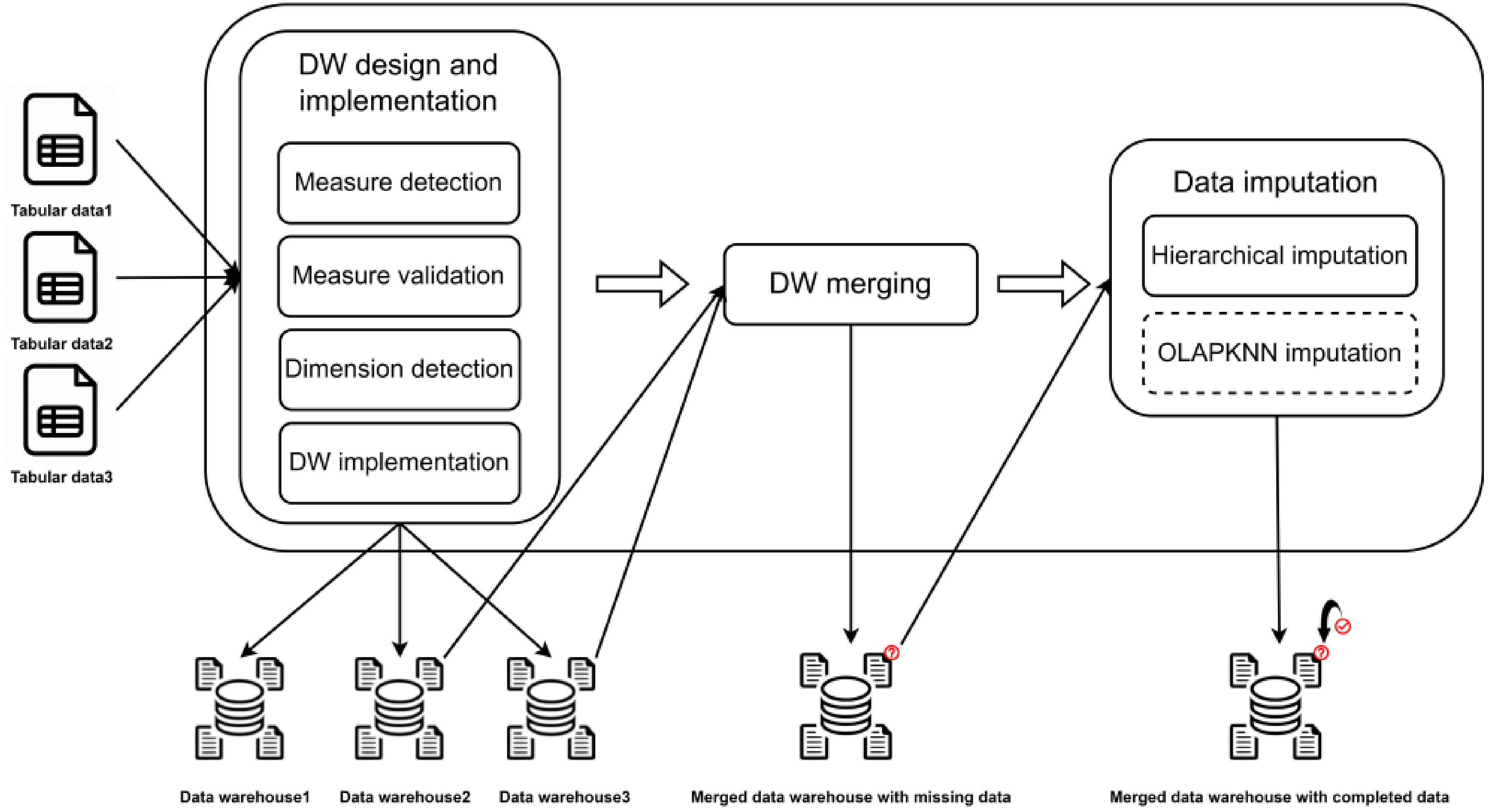
05 Implementation

Functional architecture

User-friendly interface

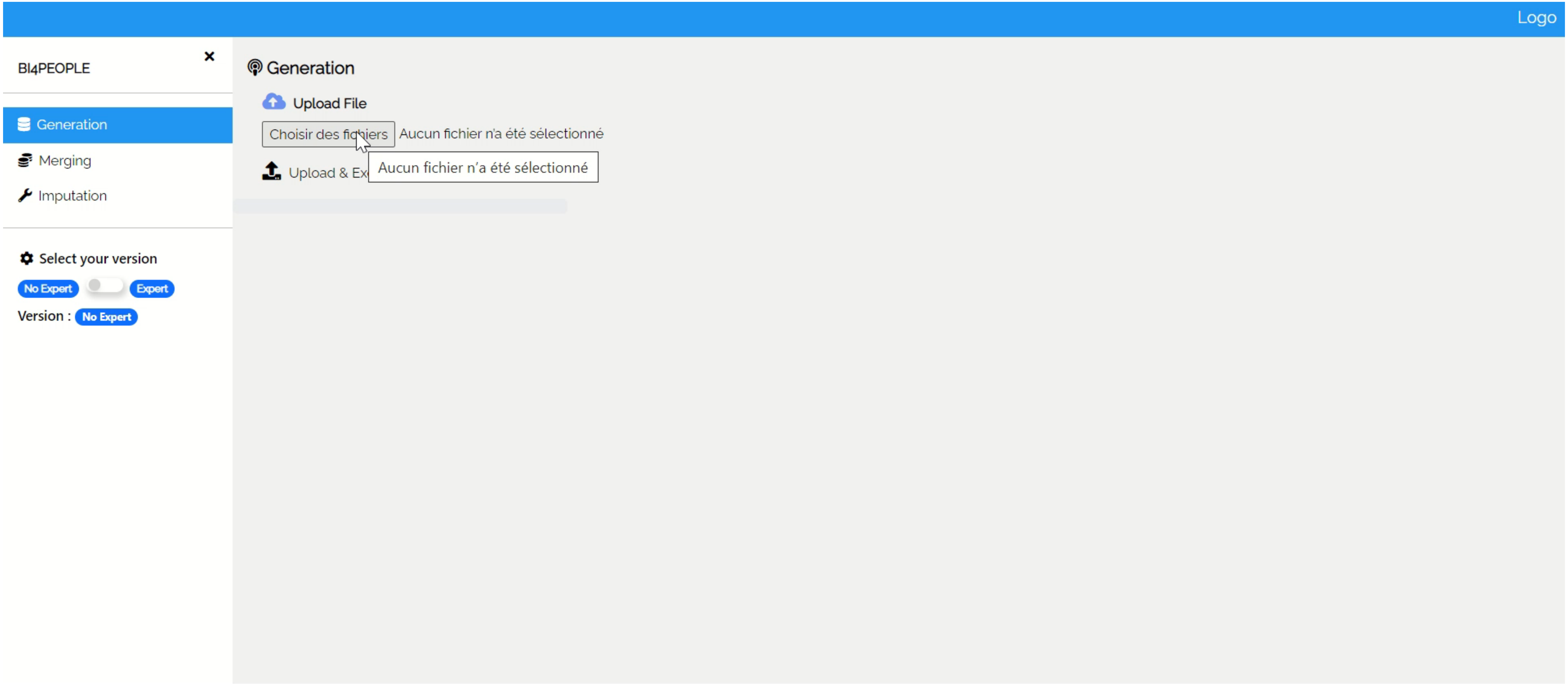
Version non-expert

Version expert



05 Implementation

Demo



Content

01 Introduction
02 Automatic Data Warehouse Design
03 Data Warehouse Merging
04 Data Imputation
05 Implementation
06 **Conclusion**

Contributions on Automatic DW Design from Tabular Data

- **Mesure Detection**
  - *Machine learning classification*
  - *Random forest : +17%*
  - *Relevant features*
  - *Generic model*

- **Dimension Detection**
  - *Hierarchy: functional dependency*
  - *Parameter and weak attribute: rules*
  - *Dimension: 100%*
  - *Hierarchy: 67% - 100%*

Contributions on Automatic DW Merging

- **DW merging**
  - *Schema and instance*
  - *Generation of star or constellation schema*

Contributions on Data Imputation

- **Hie-OLAPKNN**
  - *Hierarchical imputation*
  - *OLAPKNN: specific distance*
  - *Effective : + 45%*
  - *Efficient :* -+19 times

Contributions on Tabular Data Integration Application

- **Application**
  - *3 fonctionnalities*
  - *User-friendly interface*
  - *Non-expert and expert version*

06 Conclusion

Future work

Short-term plan

- Automatic DW Design Approach Extension by Ontologies
- Imputation by External Sources

Mid-term plan

- Schema Evolution of Sources

Long-term plan

- Other Semi/Non-structured Data in Data Lakes

Thank you!