ML to Data Management: A Round Trip

Laure Berti-Equille

Aix-Marseille University, CNRS, LIS
France
laure.berti-equille@univ-amu.fr

Angela Bonifati

University Claude Bernard Lyon 1
France
angela.bonifati@univ-lyon1.fr

Tova Milo

Tel Aviv University
Israël
milo@cs.tau.ac.il

ICDE PARIS 2018

34th IEEE International Conference on Data Engineering
April 16th – 19th

Who We Are

Laure Berti-Equille

Aix-Marseille University
France
laure.berti-equille@univ-amu.fr

Angela Bonifati

University Claude Bernard Lyon 1
France
angela.bonifati@univ-lyon1.fr

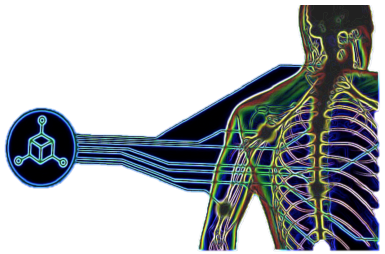
Tova Milo

Tel Aviv University
Israël
milo@cs.tau.ac.il

ML Revolutionizes Industry

Security and Surveillance
Facial and character recognition, automatic fraud detection, plagiarism detection, DDoS detection, etc.

eHealth
Automate screening tool for medical imagery diagnostics, bio-augmentation, etc.



Manufacturing
optimizing fab operations, automating quality testing, inventory, asset, and supply chain management, predictive maintenance, etc.

Machine Learning Applications

Autonomous vehicles

Smart eCommerce
Product recommendations, demand forecasting, search, classification, matching, etc.

Digital Marketing
User conversion prediction, Ad scoring, customer targeting, brand tracking, viral marketing analysis, etc.

Personal assistant
Predictive help, automatic speech recognition, dialog management, etc.

Hot Topic for DB community

[VLDB'17 Keynote]

Deep Learning (m)eats Databases

(shortened)

Jens Dittrich

[SIGMOD Record 2016]

Machine Learning and Databases: The Sound of Things to Come or a Cacophony of Hype?

Divy Agrawal, QCRI, dagrawal@qf.org.qa
Magdalena Balazinska, University of Washington, magda@cs.washington.edu
Michael Cafarella, University of Michigan, michjc@michigan.edu
Michael Jordan, UC Berkeley, jordan@cs.berkeley.edu
Tim Kraska, Brown University, tim_kraska@brown.edu
Raghu Ramakrishnan, Microsoft, raghu@microsoft.com
Christopher Ré, Stanford, chrismre@cs.stanford.edu

Categories and Subject Descriptors
H.2.0 [**Information Systems**]: Database Management

General Terms
Database Research, Machine Learning

Keywords
Database Research, Machine Learning, Panel

1. INTRODUCTION

Machine learning seems to be eating the world with a new breed of high-value data-driven applications in image analysis, search, voice recognition, mobile, and office productivity products. To paraphrase Mike Stonebraker, *machine learning is no longer a zero-billion-dollar business*. As the home of high-value, data-driven applications for over four decades, a natural question for database researchers to ask is: *what role should the database community play in these new data-driven machine-learning-based applications?*

The last few years have seen increasing crossover between database research and machine learning. But is this crossover a wise choice for database research? What are the opportunities and the costs of this approach to industry, to the future of database research, and to academics? Do database researchers have something to contribute to this trend? These two areas have dissimilar traditions in both research, intellectually, and in industry, so bridging the gap between the fields is likely to require considerable effort. *Is it worth it?*

2. QUESTIONS TO CONSIDER

We consider how, why, and in what way the database community could make contributions at the intersection of machine learning and databases.

What are the research opportunities and pitfalls for database researchers in these machine-learning applications?

- What are the most interesting research problems at this intersection? Are there core intellectual problems in machine learning that can only be solved with researchers from both sides? Or Are the problems all data-janitor work? If it is data janitor work, is it sufficiently interesting janitorial work to examine in research?
- Is there anything fundamentally different about building database systems that use machine learning or are designed to support machine learning? Or are these new systems just the same old thing rebranded with sexier packaging?
- To attract partners in the machine learning side of the world, we need to be viewed as providing intellectual value. What do database people know that is useful to machine learning? At which level is our knowledge useful? Should we regard machine learning as a black box? Should we apply our ideas inside the black box? Should we build systems that make the black box happy? Where is the most bang for the buck?
- Do we need a new conference on ML+Databases? Or is SIGMOD or KDD the right place?
- What is the risk to the database community if database people build machine learning tools? Could this lead to us becoming a "me-too" community, i.e., a lagging—rather than a leading—indicator? Or is this risk higher if we don't jump on the machine learning bandwagon like other fields, notably NLP and Computer Vision?
- Can we teach old dogs new tricks? Does working at the intersection of machine learning and databases require that database researchers learn an entirely new set of skills? In contrast, while Database research is applied to and often driven by business, there are few

[SIGMOD'17 Tutorial]

Database Meets Deep Learning: Challenges and Opportunities

Wei Wang[†], Meihui Zhang[‡], Gang Chen[§], H. V. Jagadish[#], Beng Chin Ooi[†], Kian-Lee Tan[†]
[†]National University of Singapore [‡]Singapore University of Technology and Design
[§]Zhejiang University [#]University of Michigan
[†]{wangwei, ooibc, tankl}@comp.nus.edu.sg [‡]meihui_zhang@sutd.edu.sg
[§]cg@zju.edu.cn [#]jag@umich.edu

ABSTRACT

Deep learning has recently become very popular on account of its incredible success in many complex data-driven applications, including image classification and speech recognition. The database community has worked on data-driven applications for many years, and therefore should be playing a lead role in supporting this new wave. However, databases and deep learning are different in terms of both techniques and applications. In this paper, we discuss research problems at the intersection of the two fields. In particular, we discuss possible improvements for deep learning systems from a database perspective, and analyze database applications that may benefit from deep learning techniques.

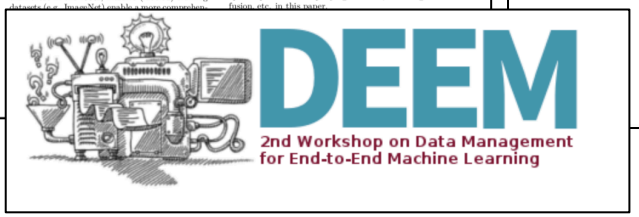
1. INTRODUCTION

In recent years, we have witnessed the success of numerous data-driven machine-learning-based applications. This has prompted the database community to investigate the opportunities for integrating machine learning techniques in the design of database systems and applications [29]. A branch of machine learning, called deep learning [22, 18], has attracted worldwide interest in recent years due to its excellent performance in multiple areas including speech recognition, image classification and natural language processing (NLP). The foundation of deep learning was established about twenty years ago in the form of neural networks. Its recent resurgence is mainly fueled by three factors: immense computing power, which reduces the time to train and deploy new models, e.g. Graphic Processing Unit (GPU) enables the training systems to run much faster than those in the 1990s; massive (labeled) training datasets (e.g., ImageNet) enable a more comprehen-

tem optimization and large scale data-driven applications since 1970s, which are closely related to the first two factors. It is natural to think about the relationships between databases and deep learning. First, are there any insights that the database community can offer to deep learning? It has been shown that larger training datasets and a deeper model structure improve the accuracy of deep learning models. However, the side effect is that the training becomes more costly. Approaches have been proposed to accelerate the training speed from both the system perspective [5, 19, 9, 28, 11] and the theory perspective [45, 12]. Since the database community has rich experience with system optimization, it would be opportune to discuss the applicability of database techniques for optimizing deep learning systems. For example, distributed computing and memory management are key database technologies. They are also central to deep learning.

Second, are there any deep learning techniques that can be adapted for database problems? Deep learning emerged from the machine learning and computer vision communities. Recently, it has been successfully applied to other domains, like NLP [13]. However, few studies have been conducted using deep learning techniques for database problems. This is partially because traditional database problems — like indexing, transaction and storage management — involve less uncertainty, whereas deep learning is good at predicting over uncertain events. Nevertheless, there are problems in databases like knowledge fusion [10] and crowdsourcing [27], which are probabilistic problems. It is possible to apply deep learning techniques in these areas. We will discuss specific problems like querying interface, knowledge fusion, etc. in this paper.

[workshop@SIGMOD]



[SIGMOD'15 Panel]

Data Management in Machine Learning: Challenges, Techniques, and Systems

Arun Kumar, UC San Diego, La Jolla, CA, USA
Matthias Boehm, IBM Research – Almaden, San Jose, CA, USA
Jun Yang, Duke University, Durham, NC, USA

ABSTRACT

Large-scale data analytics using statistical machine learning (ML), popularly called advanced analytics, underpins many modern data-driven applications. The data management community has been working for over a decade on tackling data management-related challenges that arise in ML workloads, and has built several systems for advanced analytics. This tutorial provides a comprehensive review of

focus is on analyzing the technical challenges and on explaining the key ideas, architecture, strengths, and limitations of major systems that address these challenges. This tutorial aims to provide data management researchers and systems developers with a survey of effective techniques and open issues, and to help identify systems they could build upon or compare with. It could also help data scientists understand the assumptions, pros, and cons of different systems and make more informed choices for their applications.

[SIGMOD Blog, Feb. 2018]

ACM SIGMOD Blog

Azza Abouzied and Paolo Papotti

FEBRUARY 14, 2018

COURTING ML: WITNESSING THE MARRIAGE OF RELATIONAL & WEB DATA SYSTEMS TO MACHINE LEARNING

Big Data , Databases , Machine Learning No Comment

The web is an ever-evolving source of information, with data and knowledge derived from it powering a great range of modern applications. Accompanying the huge wealth of information, web data also introduces numerous challenges due to its size, diversity, volatility, inaccuracy, and contradictions. This year's **WebDB 2018** theme emphasizes the challenges and opportunities that arise at the intersection of web data and machine learning research. On one hand, a large portion of web data fuels ML, with novel applications such as predictive analytics, Q&A chat bots, and content generation. On the other hand, the new wave of ML technology found its way into traditional Web data challenges, with contributions such as web data extraction with deep learning, and using ML to optimize data processing pipelines.

To kick start the conversation on research at the cross hairs of ML and data, we interviewed **Luna Dong** (Amazon Research), **Alkis Polyzotis** (Google), **Jens Dittrich** (Saarland University), **Arun Kumar** (University of California, San Diego) and **Peter Bailis** (Stanford University). Below you will find their bios. We selected this diverse set of academic and industrial, systems and theoretical researchers to better understand the quickly evolving research field of Machine Learning and Database Systems. We asked them about their motivation for working in this field, their current work and their view on the future. We summarize our interviews along the following four questions.

Introduction

Many problems in data management need precise knowledge and reasoning about information content and linkage for tasks as:

- Information and structure extraction
- Data curation
- Data integration
- Querying & DB administration
- Privacy preservation
- Data storage

Our focus

Many DM tasks can be reformulated as a classification or an optimization problem.

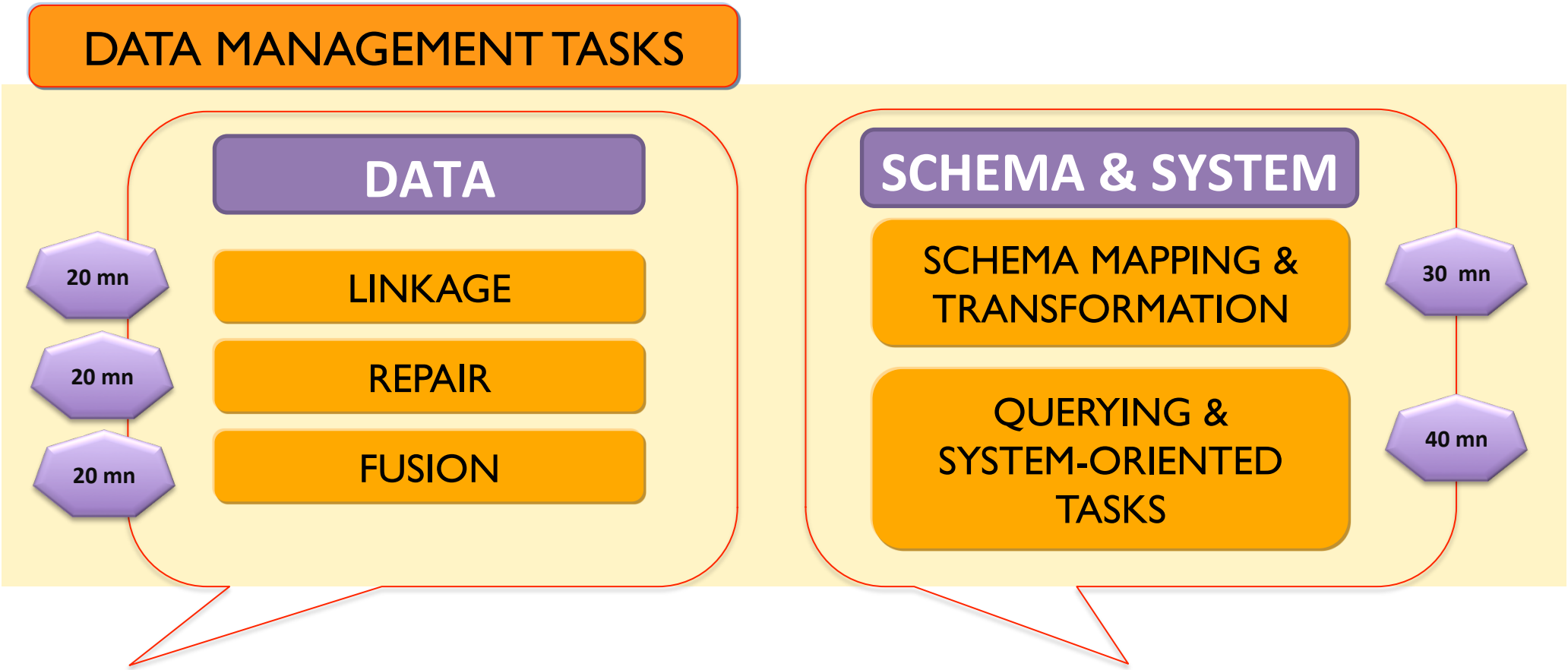
Tutorial Goals

- Offer a comprehensive review of ML applications to specific areas of data management: data curation, integration, querying, and DB tuning
- Analyze when and how ML might be leveraged for developing new areas of data management
- Analyze how data management could help ML workflows and data pipelines and contribute to ML advances

Our Tutorial is NOT

- A tutorial on ML pipelines, systems or techniques
  ➔ [Kumar, Boehm, Yang, Tutorial SIGMOD'17]
  [Polyzotis et al., Tutorial SIGMOD'17]
- Not trying to cover all domain-specific methods
- Not specific to data integration or curation
  ➔ [Dong, Rekatsinas, *coming* Tutorial SIGMOD'18]
- Not specific to Deep Learning
- Not exhaustive for the sake of conciseness

Our Focus: ML applications to DM



Tutorial Part I (morning)

Tutorial Part II (afternoon)

Main Takeaways

- Roadmap of existing ML-powered data management solutions
- Overview of open research problems
- Directions for cross-fertilization in ML and DB

ML for Data Management: A Round Trip

PART I

Laure Berti-Equille

Outline

Introduction

- Motivations
- SWOT Analysis

Part 1- ML-Powered Data Curation

- Record Linkage, Deduplication, Entity Resolution
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

Outline

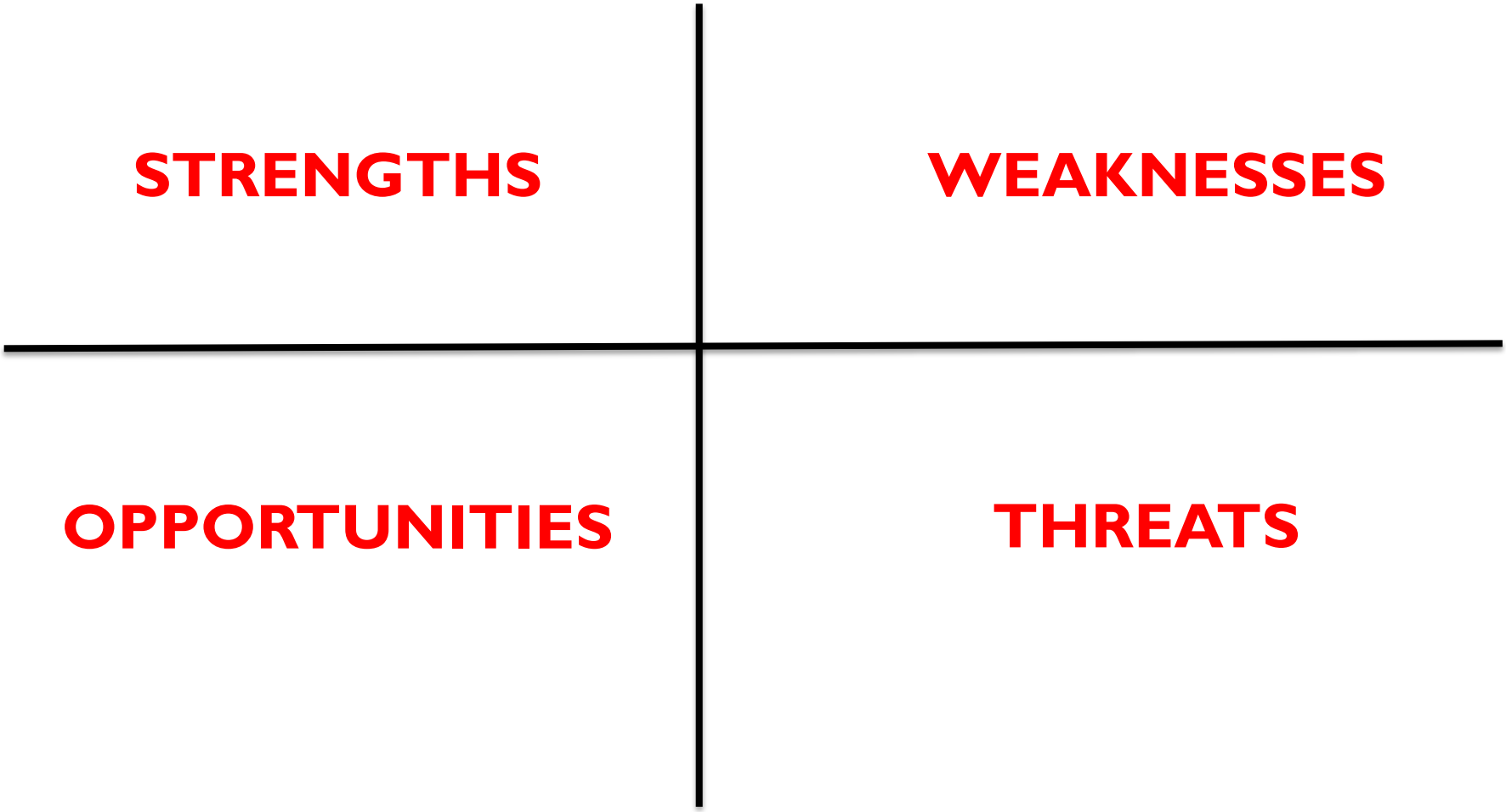
Introduction

- Motivations
- **SWOT Analysis**

Part 1- ML-Powered Data Curation

- Record Linkage, Deduplication, Entity Resolution
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

SWOT Analysis (1)



SWOT Analysis (2)

STRENGTHS

1. Leverage diverse signals/ data with semantically rich representations

2. Various techniques for learning representations

EXAMPLES

To manage multimedia and cross-modal data:

- Information extraction, Slot Filling, KB Construction [Shin et al., 2015][Wu et al., SIGMOD'18]
- Cross-modal information retrieval
- Complex event summarization
- Cross-modal synthesis of medical images
- Automatic image/video labeling

Embeddings, multiple views, hierarchical representations

- Large-scale networks representation [Tang, KDD'17 tutorial]
- Text representation and classification
- Recommendation
- Link prediction
- Visualization

SWOT Analysis (3)

STRENGTHS

3. Optimization

4. Cost reduction

5. Good alternative to heuristics

EXAMPLES

To deduplicate, repair, or fuse data:

- SCARE [Yakout et al., 2013]
- HoloClean [Rekatsinas et al., 2017]
- SLiMFast [Jogleakr et al., 2017]

To build large-scale knowledge graph:

- ML-based relation extraction can automatically generate large amount of annotated data and extract features via distant supervision [Mintz et al., 2009] reducing annotating cost

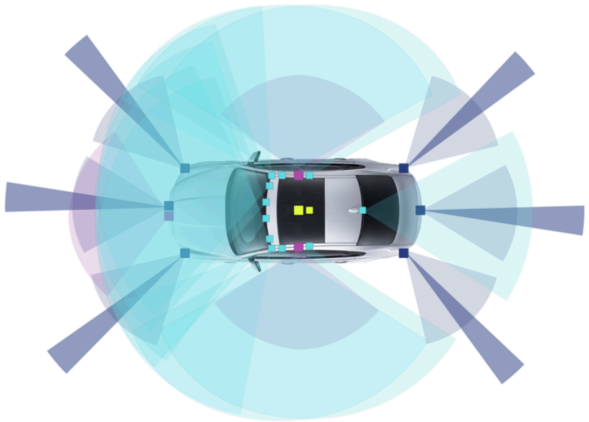
To optimize queries & tune DB: (cf. Part II)

- Complicated heuristics for estimating selectivity and query plan cost could be replaced and learn dynamically
- Regression-based automatic profiling/tuning (demo Dione [Zacheilas et al., ICDE'18]

SWOT Analysis (4)

WEAKNESSES

1. Obtaining training data is costly



EXAMPLES

- **Data annotation and preprocessing bottlenecks:** For self-driving cars, 3 million miles of driving data have to be annotated.

Assumptions	Very Conservative estimate
Fleet size	100
Duration of data collection	1 working year / 8h
Volume of data generated by a single car	1TB / h
Data reduction due to preprocessing	0.0005
Research team size	30
Proportion of the team submitting jobs	20%
Target training time	7 days
Number of epochs required for convergence	50
Calculations	
Total raw data volume	203.1 PB
Total data volume after preprocessing	104 TB
Training time on a single DGX-1 Volta system (8 GPUs)	166 days (Inception V3) 113 days (ResNet 50)21 days (AlexNet)
Number of machines (DGX-1 with Volta GPUs) required to achieve target training time for the team	142 (Inception V3) 97 (ResNet 50) 18 (AlexNet)

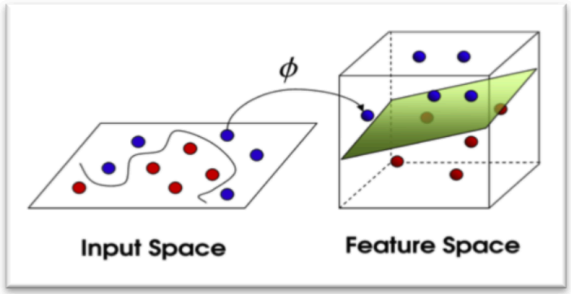
https://devblogs.nvidia.com/training-self-driving-vehicles-challenge-scale

SWOT Analysis (5)

WEAKNESSES

1. **Obtaining training data is costly**

2. **Finding or coding evidences into features is hard**



3. **Scaling to Terabytes-size datasets with millions of variables is not easy**

4. **Model interpretability is limited**

EXAMPLES

- ***Data annotation and preprocessing bottlenecks***
  - *Training data generation:* Snorkel [Ratner et al., NIPS'17] (cf. Part II)
  - *Crowdsourcing automation for labeling training data* suffers from inconsistent quality because expertise is hard to get.
  - *Data integration and curation* are required but generally ad-hoc to get clean training data with well-defined features relevant for the ML models.
- ***Deep model training is computationally-expensive.*** Techniques for "Learning to learn", and hyper-parameter optimization can multiply training computation by 5-1000X. [Marcus, Arxiv, 2018]
- ***Understand the decisions of Convolutional Neural Network is not straightforward***

  Human beings usually cannot fully trust a network, unless it can explain its logic for decisions (NIPS 2017 Interpretable ML Symposium: http://interpretable.ml/)

SWOT Analysis (6)

OPPORTUNITIES

1. **Revisit DBMS design, techniques and the whole "DBMS abstraction"** [Dittrich, Keynote VLDB'17]

> *"ML hardware is at its infancy."*
>
> [Dean, NIPS 2017]
>
> http://learningsys.org/nips17/assets/slides/dean-nips17.pdf
>
> What about ML DBMS?

2. **Apply core-DB technologies to ML workloads**

EXAMPLES

To improve components of a DB system:

- Learned Index structure [Kraska et al., 2017]
- NoDBA project [Sharma et al., 2018] using reinforcement learning to tune a database as a virtual database administrator

Automated testing of DB applications:

ETL regression testing [Dzakovic, XLDB'18]

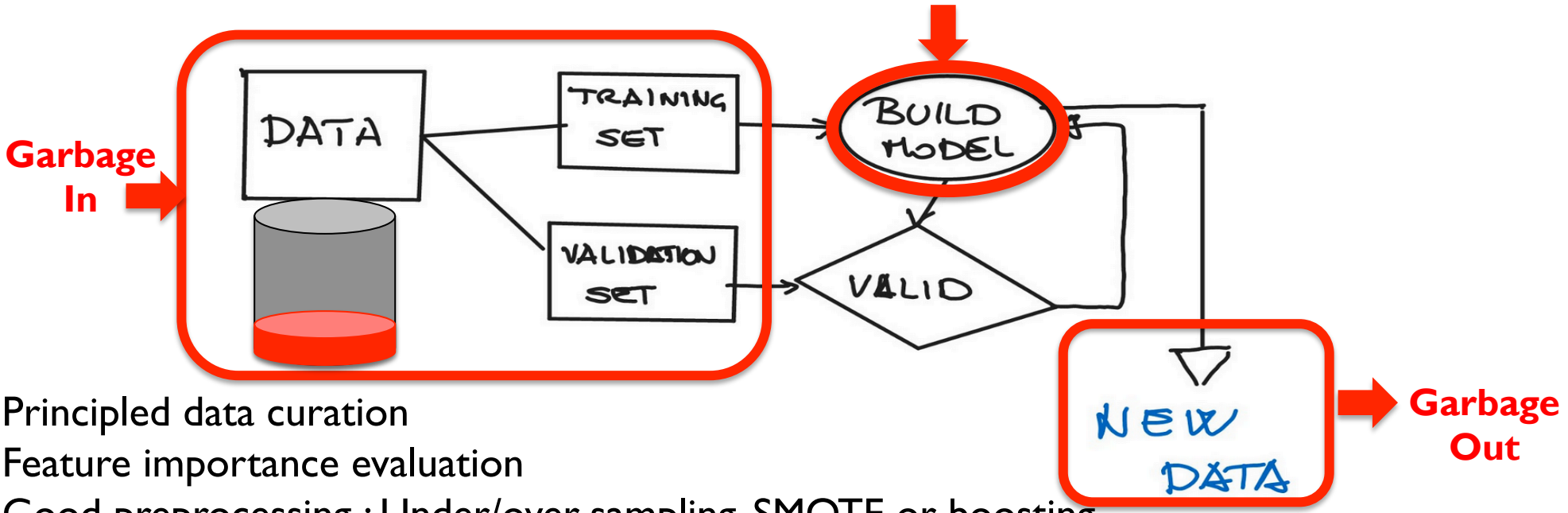
When releasing ETL upgrades, the stakes are high: a single defect can spoil the data in the DB, and the worst-case recovery from a backup would take days

Principled data curation and preprocessing for ML

SWOT Analysis (7)

THREATS

1. **Learning from dirty data is risky**
2. **Bad feature engineering**
3. **Minority class problem in unbalanced dataset**



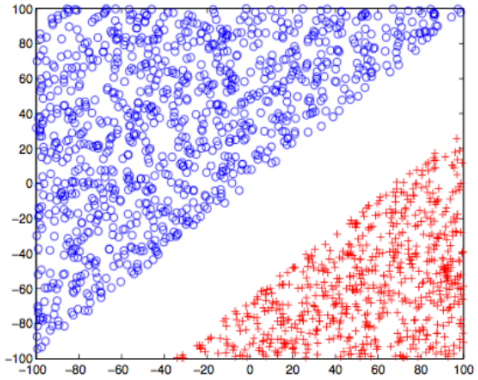
➔Principled data curation
➔Feature importance evaluation
➔Good preprocessing : Under/over-sampling, SMOTE or boosting

SWOT Analysis (8)

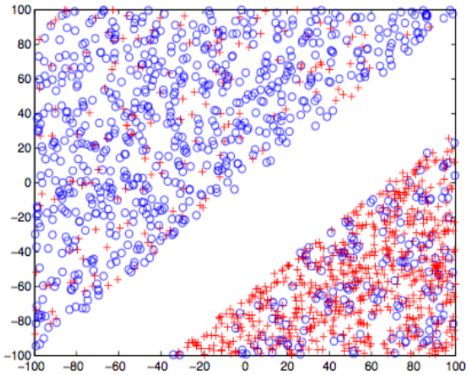
Learning from noisy labels is a hot topic in ML

[Natarajan et al., NIPS'13]

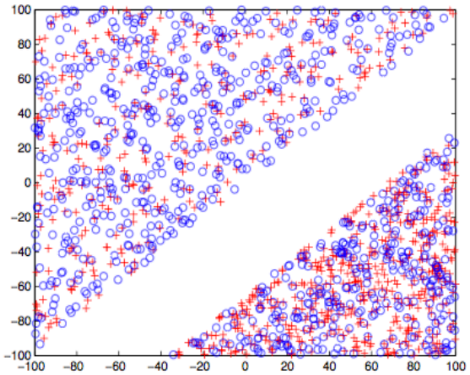
noise-free data



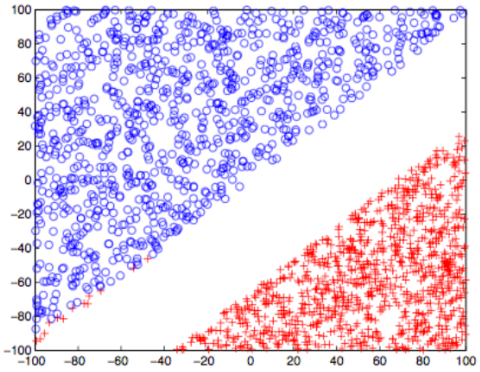
corrupted data ρ=.2



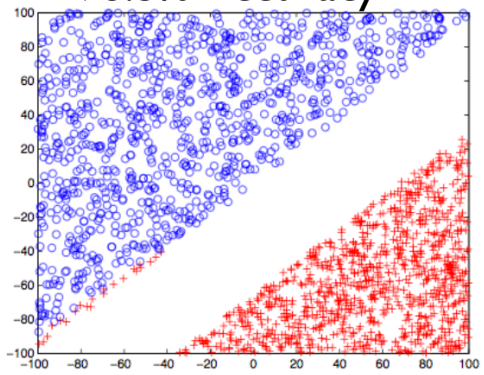
ρ=.4 rate par class



C-SVM Results



98.5% Accuracy



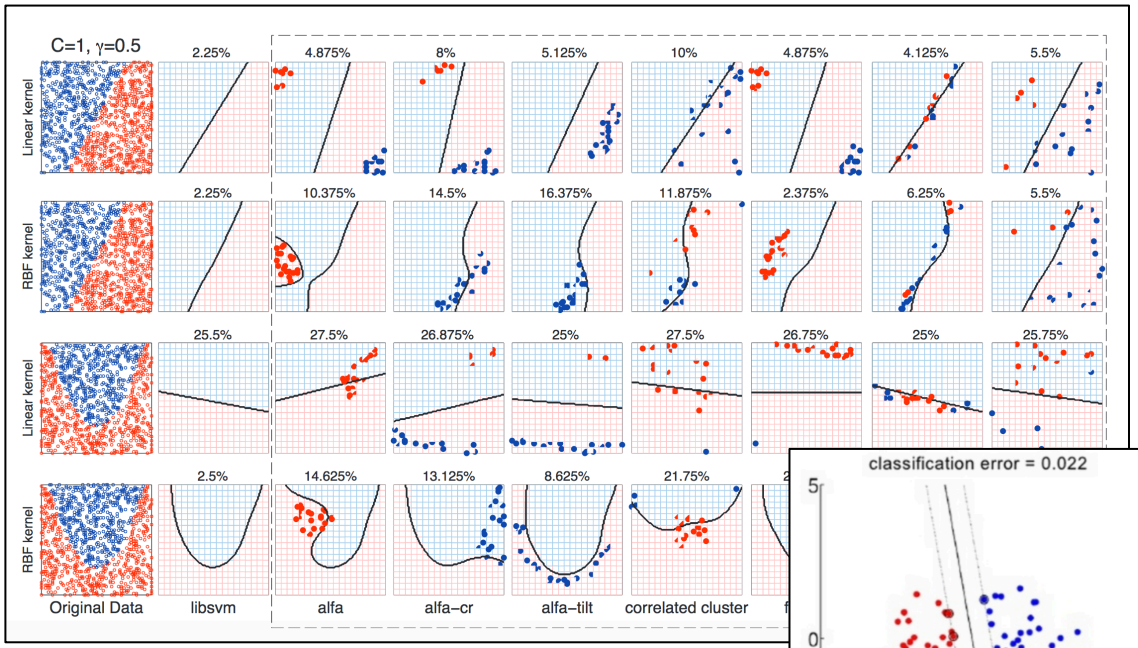
SWOT Analysis (9)

THREATS

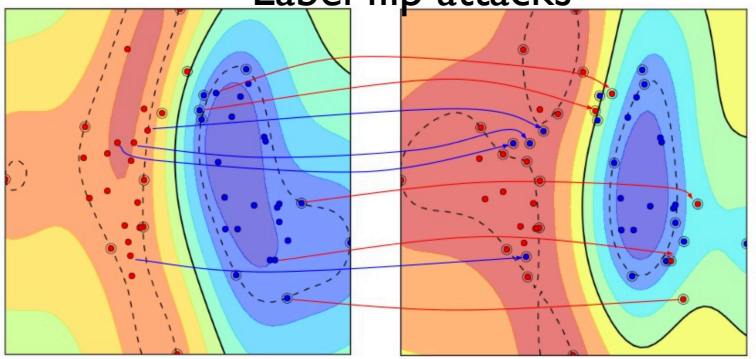
4. Adversarial Learning

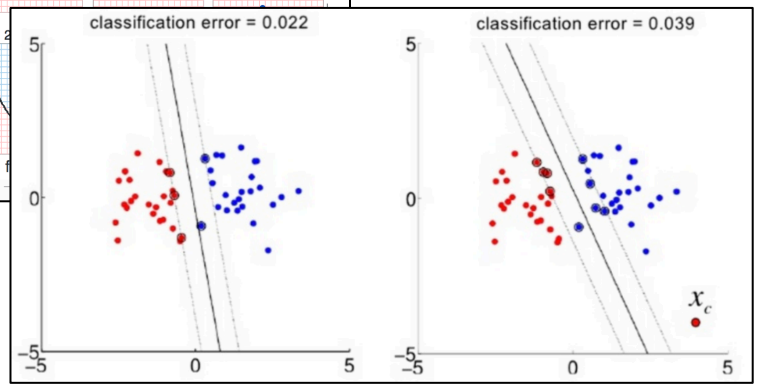
[Xiao et al., Neurocomputing 2014][Biggio et al., ICML'12]

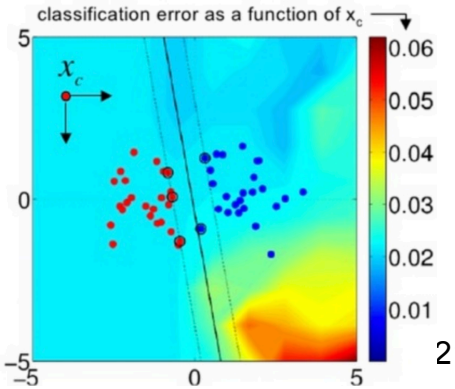
Poisoning Attacks on SVM



Label flip attacks







SWOT Analysis: A Summary (10)

STRENGTHS

1. Leverage diverse signals/data with semantically rich representations
2. Various techniques for learning representations
3. Good alternative to heuristics
4. Optimization with objective functions
5. Reduction of annotating cost

WEAKNESSES

1. Training data annotation and preprocessing is costly
2. Finding/coding evidences into features is hard
3. Scaling to TB-size datasets with millions of variables is challenging
4. Model interpretability can be limited

OPPORTUNITIES

1. Revisit design, techniques, and "DBMS abstraction"
2. Apply core-DB technologies to ML workloads

THREATS

1. Learning from dirty data is risky
2. Bad feature engineering
3. Minority class problem in unbalanced dataset
4. Adversarial Learning

Outline

Introduction

- Motivations
- SWOT Analysis

Part 1- ML-Powered Data Curation

- Record Linkage, Entity Resolution, Deduplication
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

Record Linkage (RL): Generic Workflow

Database R

Name	SSN	Addr
Will Forth	354-564-339	Ada Bd
Jacky Khan	435-232-129	Marple Street
Dom Hack	235-575-689	Main Street
…	…	…

Database S

Name	SSN	Addr
Jack Khan	435-223-129	Marple St
Hans Ford	354-564-339	Clover Bd
Tom Hack	235-557-689	Main St
…	…	…

R X S

Cleaning Standardization

Attribute selection

Blocking

- Hashing
- Sorted keys
- Sorted NN
- (Multiple) Windowing
- Clustering

[Fellegi, Sunter, 1969]
[Christen, 2012]

{pairs}

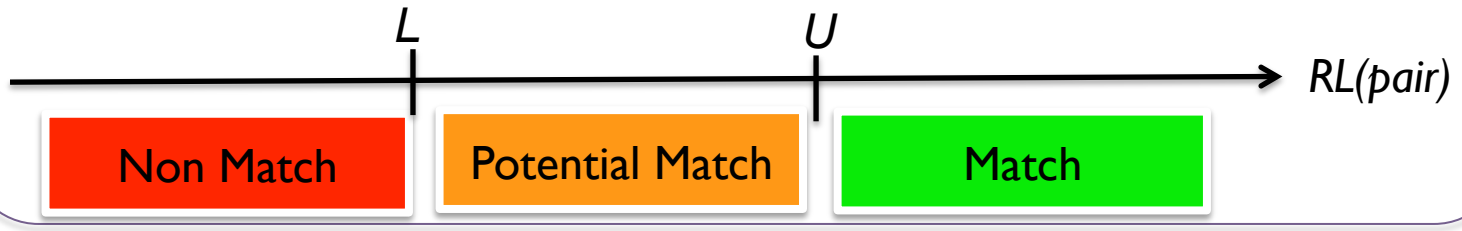
Record pair comparison

- Token-based : N-grams…
- Distance-based: Jaro, Edit, Levenshtein, Soundex
- Domain-dependent

{comparison vectors }

Decision Model

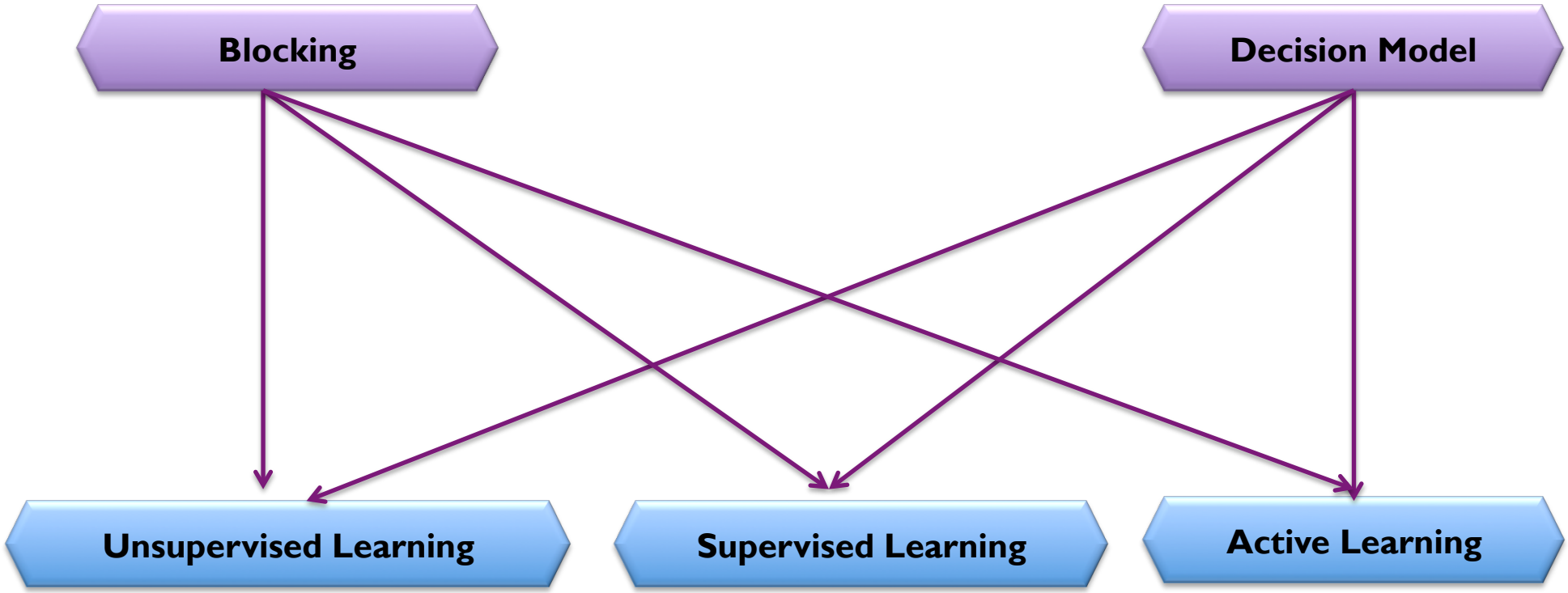
Linkage decision: $RL(pair) = \frac{P(vector \mid pair \in Match)}{P(vector \mid pair \in Non\ Match)}$



ML for Entity Resolution (ER)

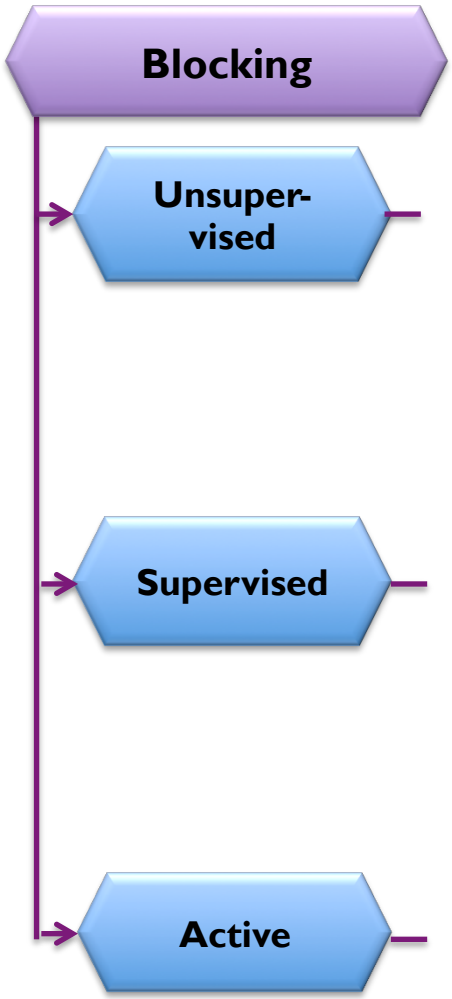
[Getoor, Machanavajjhala, Tutorial VLDB'12]

[Christen, 2012]



ML-based ER approaches (1)

[Papadakis, Palpanas, Tutorial ICDE'16]



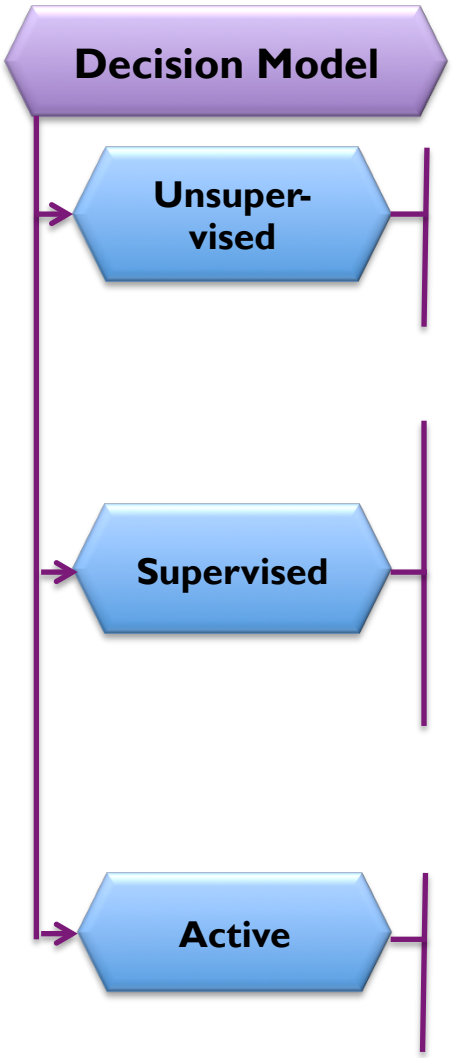
Attribute-clustering blocking [Papadakis et al., TKDE 2013]

Block-clustering [Fisher et al., KDD'15]

Learning blockers [Bilenko et al., ICDM'06]

Crowdsourcing [Wang et al., VLDB'12][Gokhale et al., SIGMOD'14]

ML-based ER approaches (2)



ML Variants of the Fellegi-Sunter model

Clustering [Chaudhuri et al., ICDE'05][Hassanzadeh et al., PVLDB'09]

Collective ER [Battacharya, Getoor, TKDD'07]

Regression Classification [Hu et al, 2017]

Support Vector Machines [Bilenko, Mooney, KDD'03]

Decision Trees [Chaudhuri et al., VLDB'07]

Conditional Random Fields [Singla, Domingos, PKDD'05] [Gupta, Sarawagi, VLDB'09]

Committee of classifiers [Sarawagi, Bhamidipaty, KDD'02]

Ensemble of classifiers [Chen et al., SIGMOD'09] [Bilenko, Mooney, KDD'03] [Tejada et al. KDD'02]

Pioneer ML-based Deduplication

[Sarawagi, Bhamidipaty, KDD'02]
[Koudas, Srivastava, Sarawagi, Tutorial SIGMOD'06]

Training examples

Customer 1	D
Customer 2	
Customer 1	N
Customer 3	
Customer 4	D
Customer 5	

f_1	f_2	...	f_n	
1.0	0.4	...	0.2	1
0.0	0.1	...	0.3	0
0.3	0.4	...	0.4	1

← *Similarity distance functions*

Unlabeled list

Customer 6
Customer 7
Customer 8
Customer 9
Customer 10
Customer 11

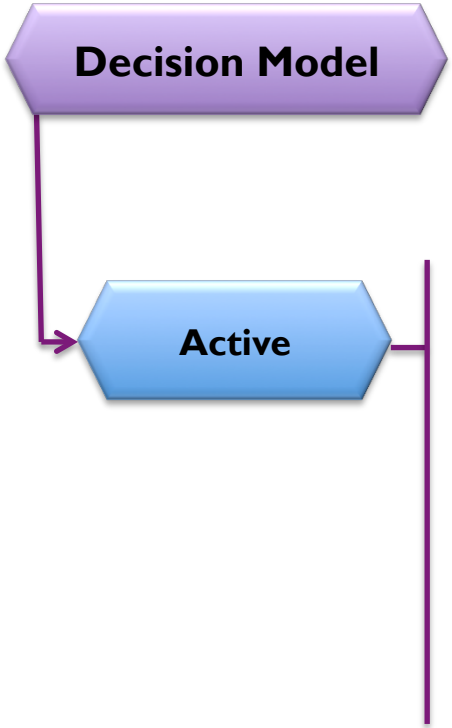
0.0	0.1	...	0.3	?
1.0	0.4	...	0.2	?
0.6	0.2	...	0.5	?
0.7	0.1	...	0.6	?
0.3	0.4	...	0.4	?
0.0	0.1	...	0.1	?

Classifier

Learnt Rule: All-Ngrams*0.4
+ CustomerAddressNgrams*0.2
– 0.3EnrollYearDifference
+ 1.0*CustomerNameEditDist
+ 0.2*NumberOfAccountsMatch – 3 > 0

Learners:
SVMs: high accuracy with limited data [Christen, 2008]
Decision trees: interpretable, efficient to apply
Perceptrons: efficient incremental training
[Bilenko et al., 2005]

ML-based ER approaches (3)



Learning similarity functions and thresholds

Sampling and labeling

- Active sampling/learning [Qian et al., CIKM'17] [Arasu et al., SIGMOD'10] [Bellare et al., KDD'12]

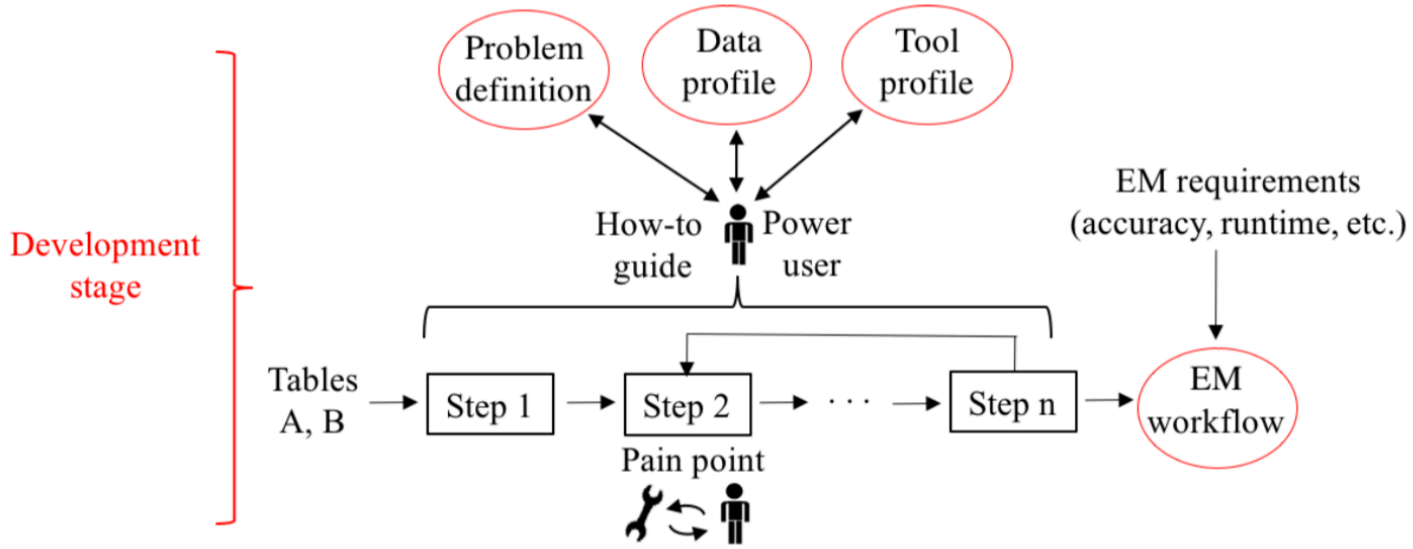
Crowdsourced ER

- Crowdsourcing algorithms for ER [Vesdapunt et al., VLDB'14]
- CrowdER [Wang et al., VLDB'12] [Wang et al., SIGMOD'13']
- Corleone [Gokhale, et al., SIGMOD'14]

Human-In-The Loop for Entity Matching

[Doan et al., HILDA@SIGMOD'17]

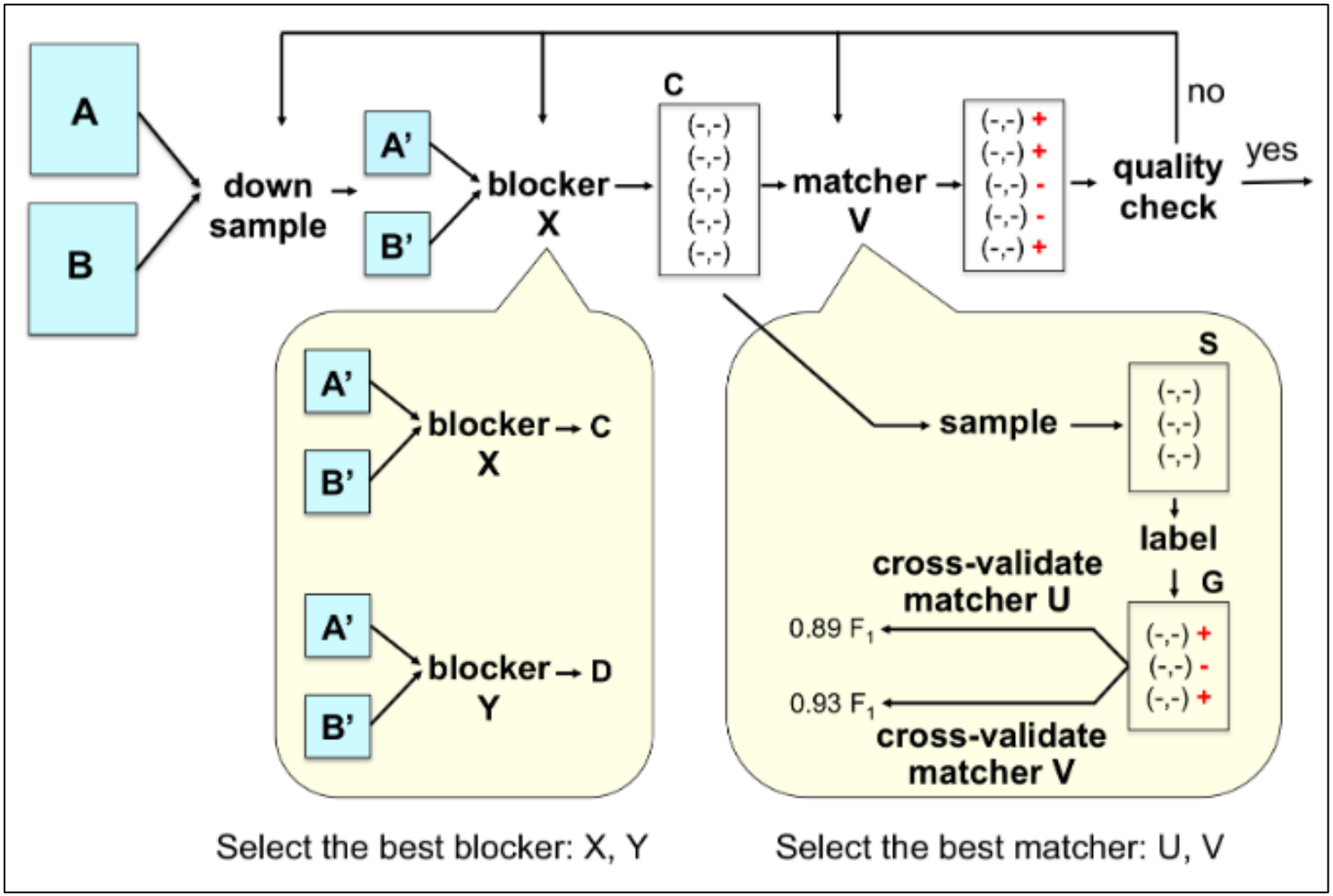
Magellan project: Lessons learnt for How-to Guide for EM



Human-In-The Loop for Entity Matching

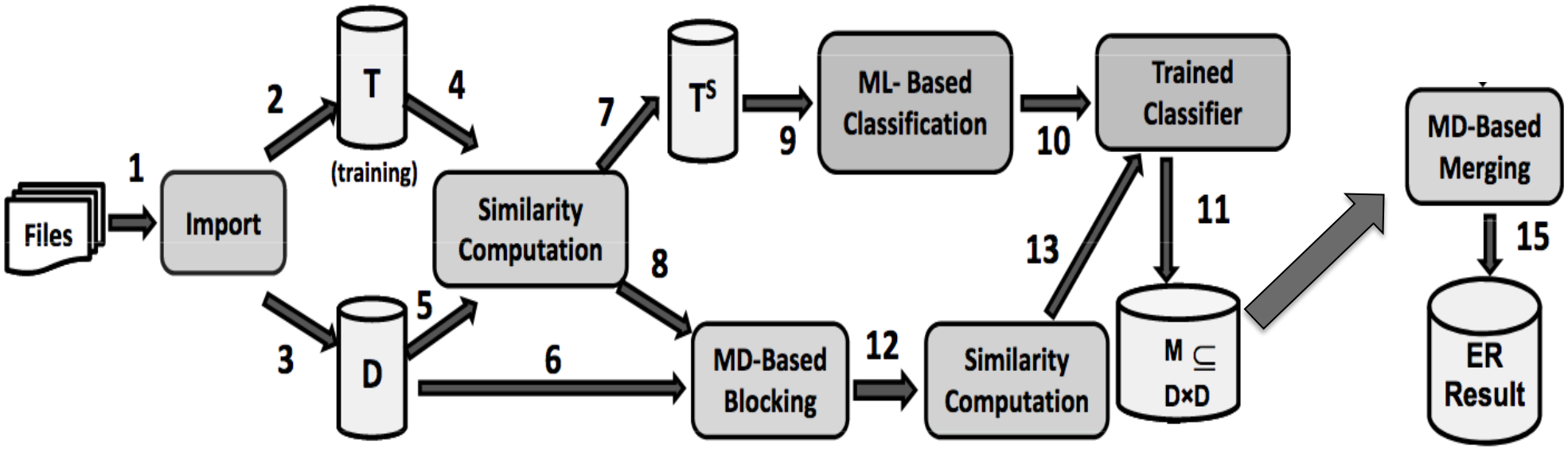
[Doan et al., HILDA@SIGMOD'17]

Magellan project: Lessons learnt for How-to Guide for EM



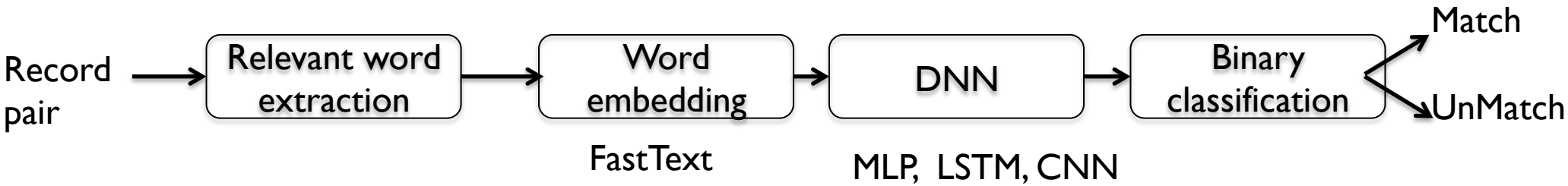
ERBlox with ML and Matching Dependencies

[Bahmani et al., SUM'15]

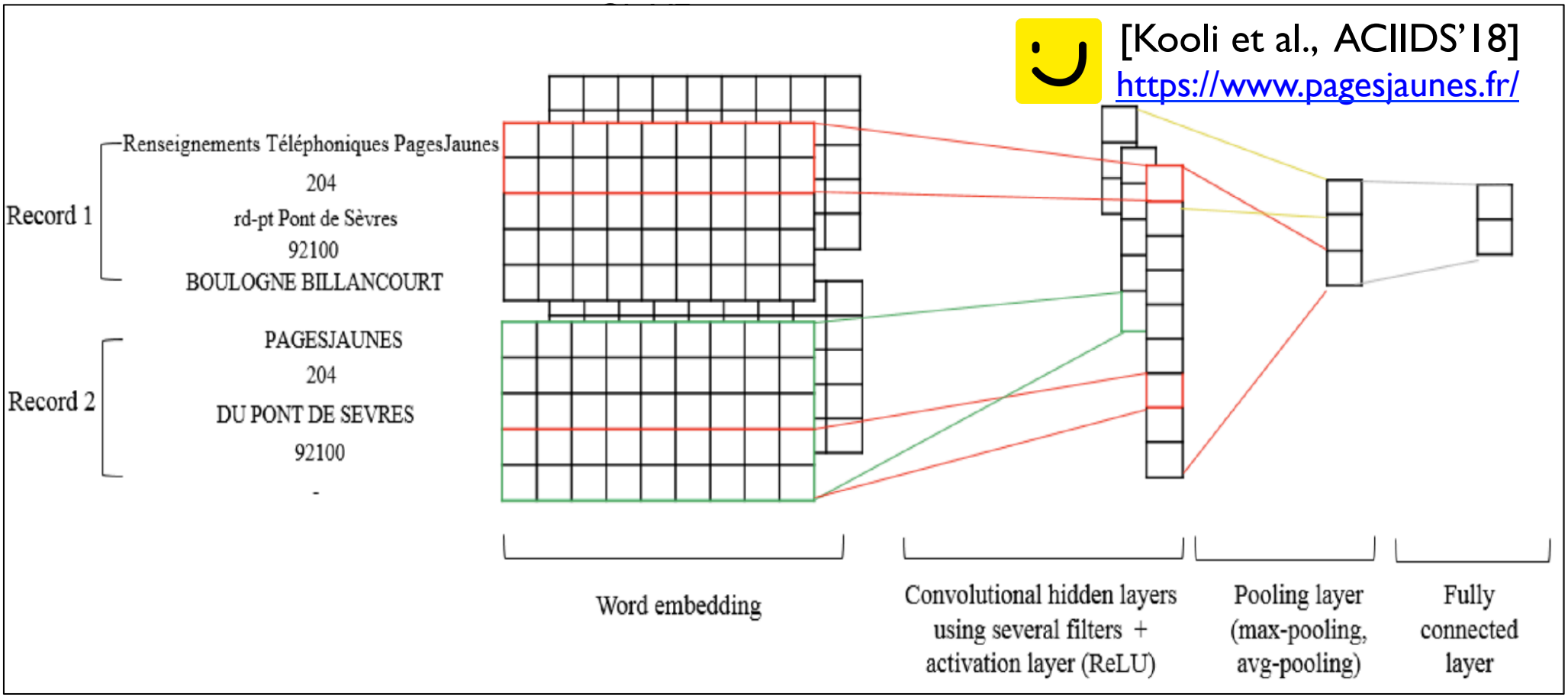


Matching dependency φ for R1 and R2 : $\wedge_{j\in[1,k]}(R_1[X_1[j]] \approx_j R_2[X_2[j]]) \rightarrow R_1[Z_1] \rightleftharpoons R_2[Z_2],$

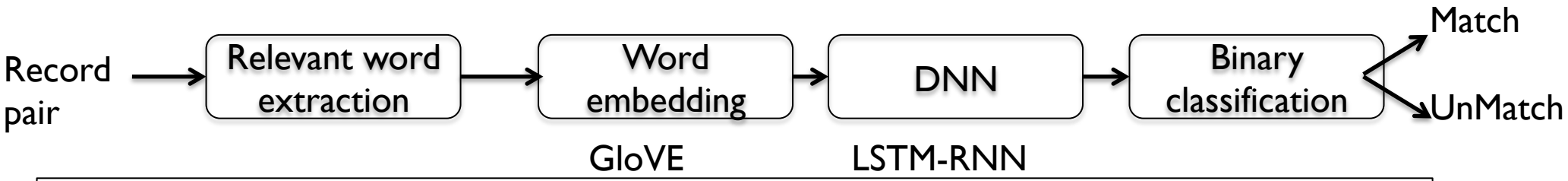
Deep learning for ER (1)

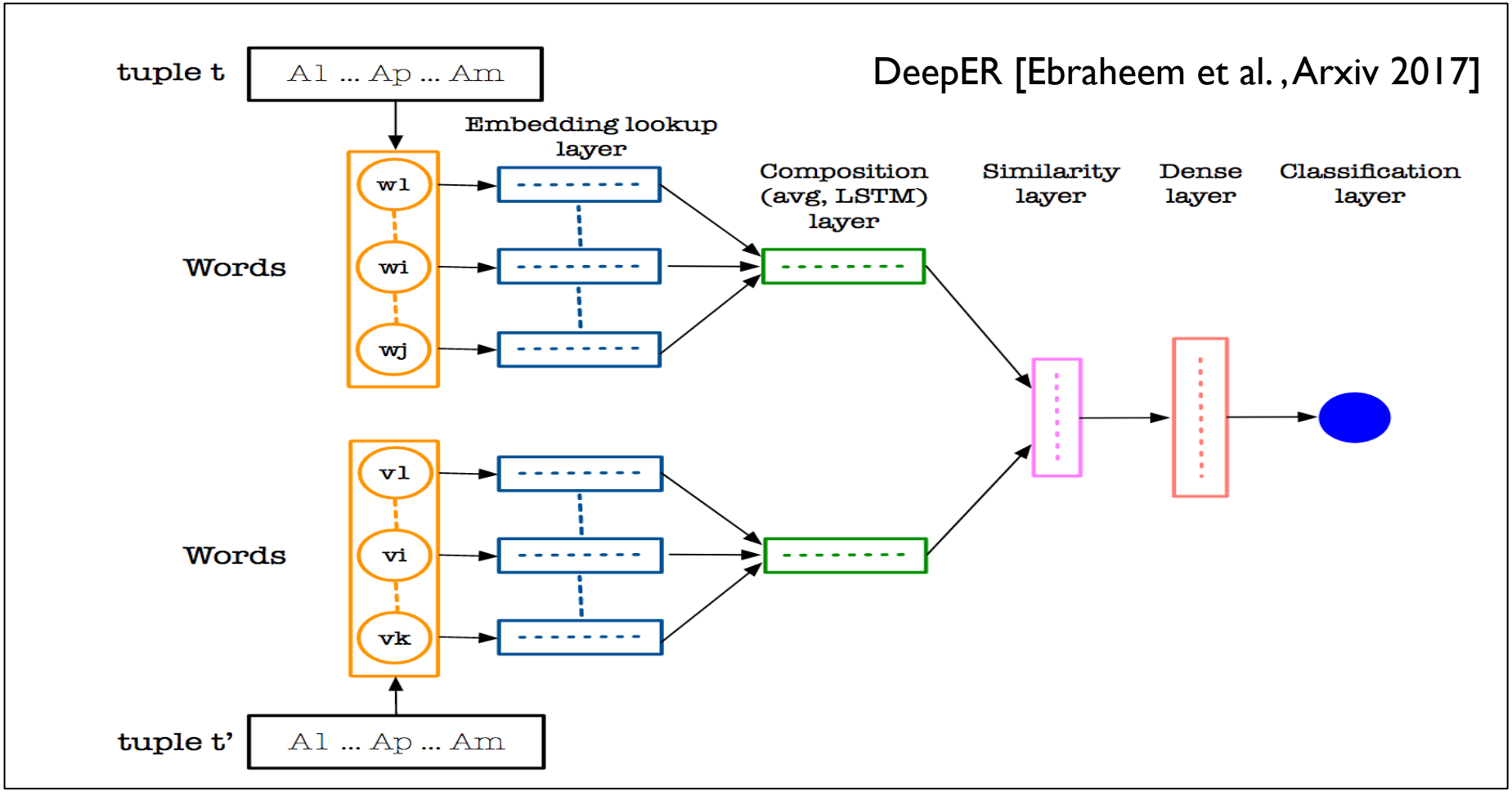


[Kooli et al., ACIIDS'18]
https://www.pagesjaunes.fr/



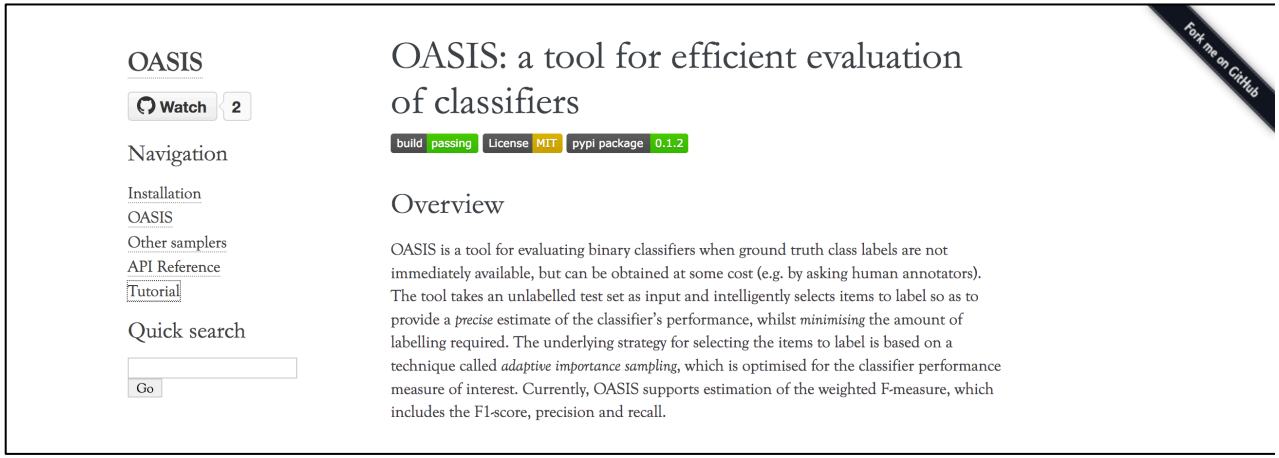
Deep learning for ER (2)



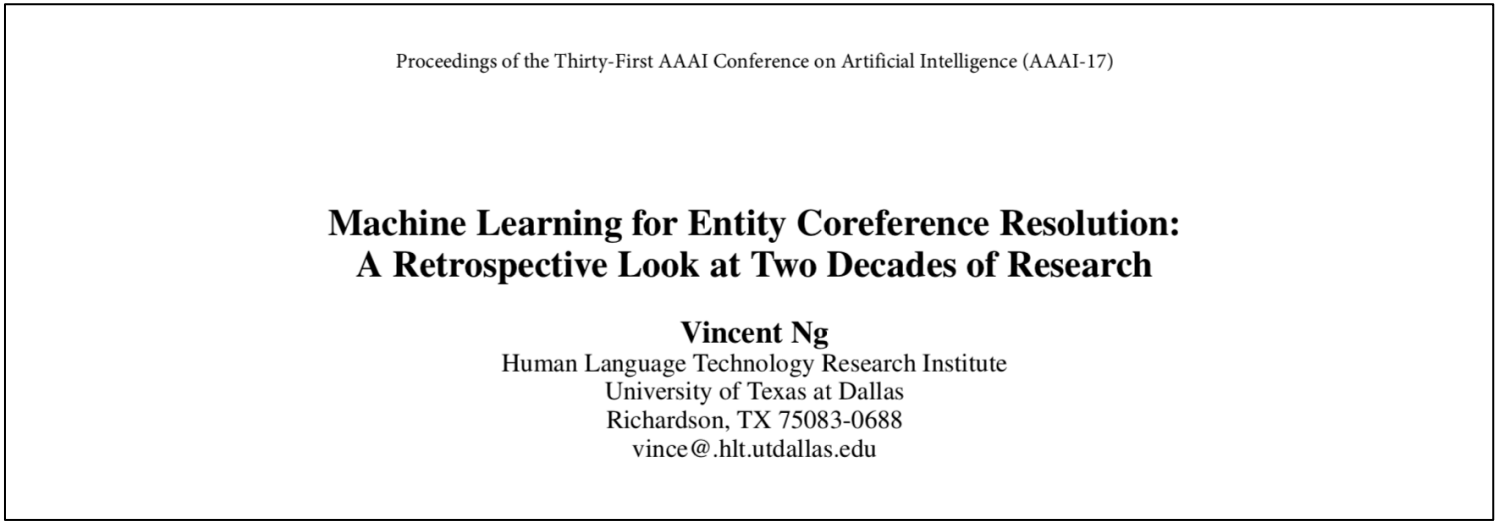


Recent Results

- Evaluation of ER with adaptive importance sampling
  [Marchand, Rubinstein, VLDB'17]



- Outside the DB sphere:



Outline

Introduction

- Motivations
- SWOT Analysis

Part 1- ML-Powered Data Curation

- Record Linkage, Entity Resolution, Deduplication,
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

ML-Based Repairing

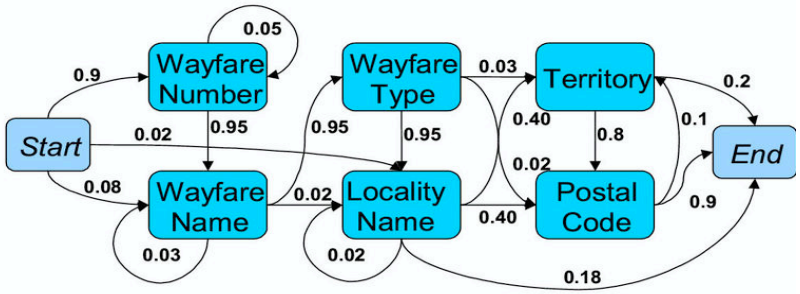
Semi-automatic techniques for:

- **Pattern enforcement**
  - Syntactic patterns (date formatting)
  - Semantic patterns (name/address)
- **Value update** to satisfy a set of rules, constraints, FDs, CFDs, Denial Constraints (DCs), Matching Dependencies (MDs) with minimal number of changes. [Ilyas, Chu, 2015]
- **Value imputation** with statistical methods to replace outliers or missing values
- **Data fusion**

Febrl: Data standardization with HMM

[Churches et al., 2002]
[Christen et al., 2002]

HMM for Address Standardization



	To state							
From state	***Start***	**Wayfare Number**	**Wayfare Name**	**Wayfare Type**	**Locality Name**	**Territory**	**Postal Code**	***End***
Start	0	0.9	0.08	0	0.02	0	0	0
Wayfare Number	0	0.05	0.95	0	0	0	0	0
Wayfare Name	0	0	0.03	0.95	0.02	0	0	0
Wayfare Type	0	0	0	0	0.95	0.03	0.02	0
Locality name								
Territory								
Postal Code								
End								

	State							
Observation Symbol	***Start***	**Wayfare Number**	**Wayfare Name**	**Wayfare Type**	**Locality Name**	**Territory**	**Postal Code**	***End***
NU	-	0.9	0.01	0.01	0.01	0.01	0.1	-
WN	-	0.01	0.5	0.01	0.1	0.01	0.01	-
WT	-	0.01	0.01	0.92	0.01	0.01	0.01	-
LN	-	0.01	0.1	0.01	0.8	0.01	0.01	-
TR	-	0.01	0.07	0.01	0.01	0.94	0.01	-
PC	-	0.04	0.01	0.01	0.01	0.01	0.85	-
UN	-	0.02	0.31	0.03	0.06	0.01	0.01	-

Selection of representative training data
"17 Epping St Smithfield New South Wales 2987"

Tokenization based on Look-up Tables
['17', 'epping', 'street', 'smithfield', 'nsw', '2987']

Tagging
['NU', 'LN', 'WT', 'LN', 'TR', 'PC']
number-locality name-wayfare type-locality name-territory-postal code

Frequency-based Maximum Likelihood Estimates
8^6 = 262,144 possible combinations of hidden states

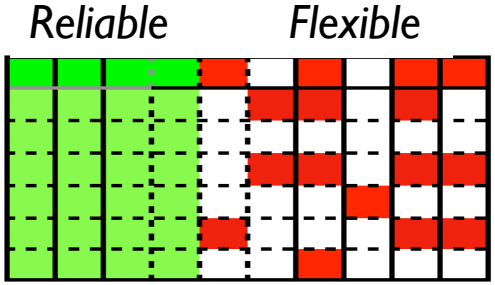
- *Start* -> Wayfare Name (NU) -> Locality Name (LN) -> Postal Code (WT) -> Territory (LN) -> Postal Code (TR) -> Territory (PC) ->*End*
  $0.08 \times 0.01 \times 0.02 \times 0.8 \times 0.4 \times 0.01 \times 0.1 \times 0.01 \times 0.8 \times 0.01 \times 0.1 \times 0.01 \times 0.2 = 8.19 \times 10^{-17}$
- *Start* -> Wayfare Number (NU) -> Wayfare Name (LN) -> Wayfare Type (WT) -> Locality (LN) -> Territory (TR) -> Postal Code (PC) ->*End*
  $0.9 \times 0.9 \times 0.95 \times 0.1 \times 0.95 \times 0.92 \times 0.95 \times 0.8 \times 0.4 \times 0.94 \times 0.8 \times 0.85 \times 0.9 = 1.18 \times 10^{-2}$

http://users.cecs.anu.edu.au/~Peter.Christen/Febrl/febrl-0.3/febrldoc-0.3/node24.html#chapter:hmm-standard

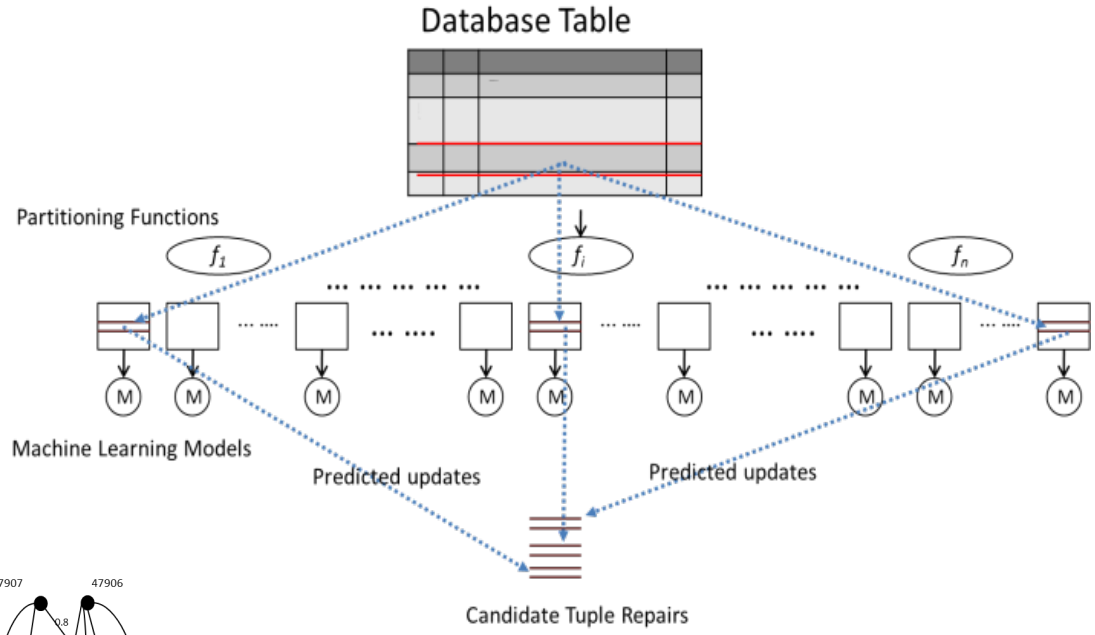
SCARE: SCalable Automatic Repair

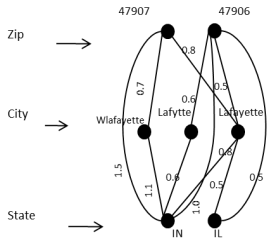
[Yakout, Berti-Equille, Elmagarmid, SIGMOD'13]

Goal: Find the repair that would maximize the sum of the probabilities of the values co-occurrence (i.e., association strength between predicted and reliable values) under a certain update cost budget.



1. Modeling Dependency and Predicting Updates
2. Data Partitioning
3. Tuple Repair Selection





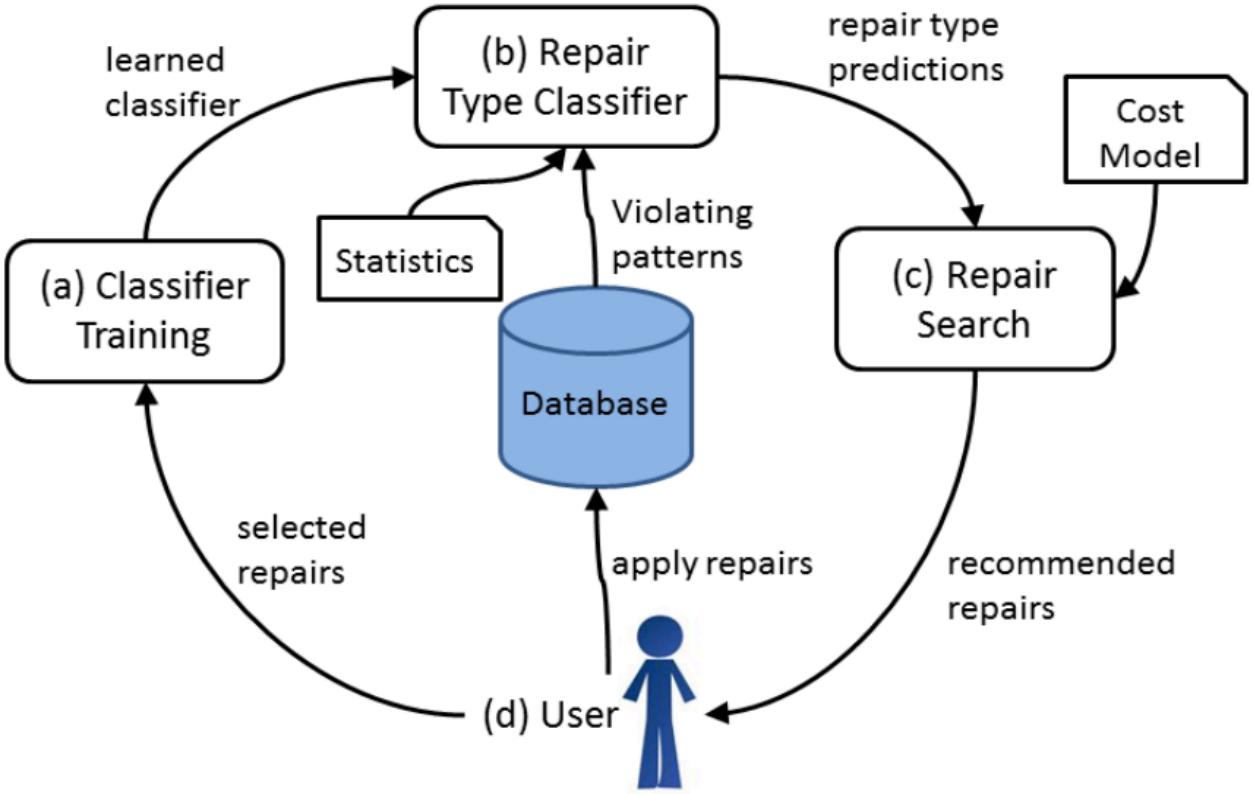
Value predictions for Flexible Attributes E1, E2, E3

Continuous Data Cleaning

[Volkovs et al., ICDE'14]

Goal: Using a logistic classifier to

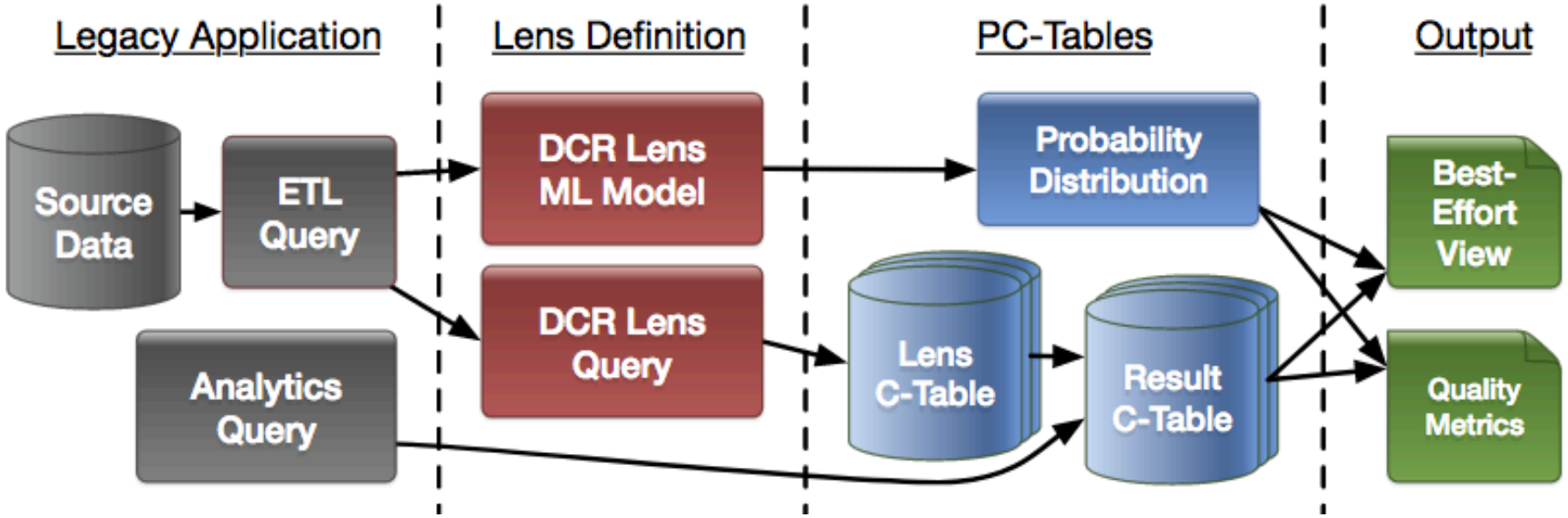
- learn from past user repair preferences to recommend next more accurate repairs;
- predict the type of repair needed to resolve an inconsistency.



On-demand ETL with Lenses

[Yang et al., VLDB'15]

Specification of Lens with classifiers from the massive online analysis (MOA) framework for Domain Constraint Repair (DCR).



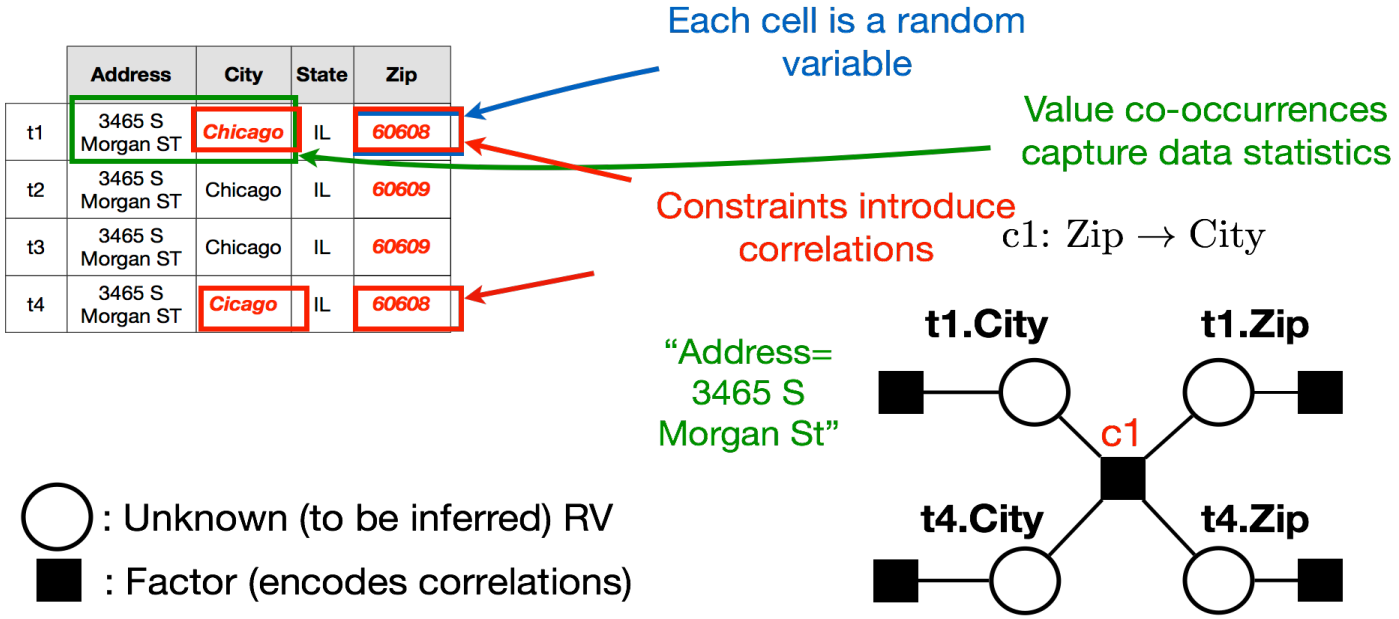

```
CREATE LENS SaneProduct AS SELECT * FROM Product
  USING DOMAIN_REPAIR( category string NOT NULL,
                       brand    string NOT NULL );
```

id	name	brand	category
P123	Apple 6s, White	$Var('X', \texttt{R1})$	phone
P124	Apple 5s, Black	$Var('X', \texttt{R2})$	phone
P125	Samsung Note2	Samsung	phone
P2345	Sony 60 inches	$Var('X', \texttt{R4})$	$Var('Y', \texttt{R4})$
P34234	Dell, Intel 4 core	Dell	laptop
P34235	HP, AMD 2 core	HP	laptop

HoloClean

[Rekatsinas et al., VLDB 2017]
https://github.com/HoloClean/HoloClean

HoloClean generates a factor graph capturing co-occurrences, correlations based on a set of constraints and external evidences. It uses SGD to learn parameters and infer the marginal distribution of unknown variables with Gibbs sampling.



	Address	City	State	Zip
t1	3465 S Morgan ST	Chicago	IL	60608
t2	3465 S Morgan ST	Chicago	IL	60609
t3	3465 S Morgan ST	Chicago	IL	60609
t4	3465 S Morgan ST	Cicago	IL	60608

Denial constraints:

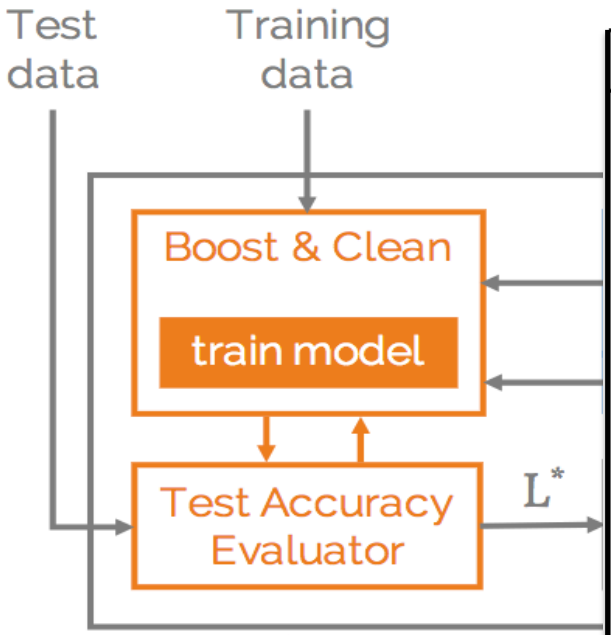
$$\forall t_1, t_2 \in D : \neg(t1[Zip] = t2[Zip] \wedge t1[City] \neq t2[City])$$

$$\forall t_1, t_2 \in D : \neg(t1[Zip] = t2[Zip] \wedge t1[State] \neq t2[State])$$

BoostClean

[Krishnan et al., 2017]

BoostClean selects an ensemble of methods (statistical and logic rules) for error detection and for repair combinations using statistical boosting.



Algorithm 2: Boost-and-Clean Algorithm

Data: (X, Y)

1 Initialize $W_i^{(1)} = \frac{1}{N}$

2 $\mathcal{L}$ generates a set of classifiers $\mathcal{C}\{C^{(0)}, C^{(1)}, ..., C^{(k)}\}$ where $C^{(0)}$ is the base classifier and $C^{(1)}, ..., C^{(k)}$ are derived from the cleaning operations.

3 **for** $t \in [1, T]$ **do**

4 C_t = Find $C_t \in \mathcal{C}$ that maximizes the weighted accuracy on the test set. ϵ_t = Calculate weighted classification error on the test set $\alpha_t = \ln(\frac{1-\epsilon_t}{\epsilon_t})$ $W_i^{(t+1)} \propto W_i^{(t)} e^{-\alpha_t y_i C_t(x_i)}$: down-weight correct predictions, up-weight incorrectly predictions.

5 **return** $C(x) = \text{sign}(\sum_t^T \alpha_t C_t(x))$

A Condensed View

Repair System	ML Approach	Goal
Febrl [Churches et al., 2002]	HMM and MLE	Standardizing loosely structured texts (e.g., name/ address) based on the probabilistic model learnt from training data
SCARE [Yakout, Berti-Equille, Elmagarmid, SIGMOD'13]	Multiple ML models used to capture data dependencies across multiple data partitions	Find the candidate repair that maximizes the likelihood repair benefit under a cost threshold of the update
Continuous Cleaning [Volkovs et al., ICDE'14]	Logistic classifiers	Learning from past user repair preferences to recommend next more accurate repairs
Lens [Yang et al. ,VLDB'15]	Various ML models encoded in Domain Constraints	Declarative on-Demand ETL with prioritized curation tasks based on probabilistic query processing and PC-Tables
HoloClean [Rekatsinas et al., VLDB 2017]	Probabilistic inference on factor graphs with SGD and Gibbs sampling	Mixing statistical and logical rules, DCs, MDs, etc. to infer candidate repairs in a scalable way with domain pruning and constraint relaxation
BoostClean [Krishnan et al., 2017]	AdaBoost	Mixing statistical and logical rules, domain constraints for detection and repair combinations to maximize the predictive accuracy over test data

Shortcomings of ML-based cleaning

Problem

- No knowledge of ground truth (the "minimal" change may not be the correct one)
- When data is missing (what data should be added?)

Solution:

- Use the crowd (of experts) to assist
- But… since data is large, focus of "hot" spots

QOCO [Bergman, Milo, Novgorodov, SIGMOD'15]
Uses the crowd to identify wrong query answers, and corrects the cause

DANCE [Assadi, Milo, Novgorodov ICDE'17, WebDB'18]
When identifying integrity constraints violation, uses the crowd to correct the cause

Optimizing crowd usage

Goal: minimize the number of questions to the crowd

General heuristic:

Identify (and ask first about) data items whose update may potentially eliminate the maximal number of violations.

Implementation of the heuristic in QOCO:

- Tracking the provenance of wrong query answers
- Asking about tuples that participate to maximal number of assignments

Implementation of the heuristic in DANCE:

- Tracking (recursively) the provenance of constraints violation
- Building a dependency graph for the tuples
- Running "page-rank" on the graph to identify potentially influential tuples

Outline

Introduction

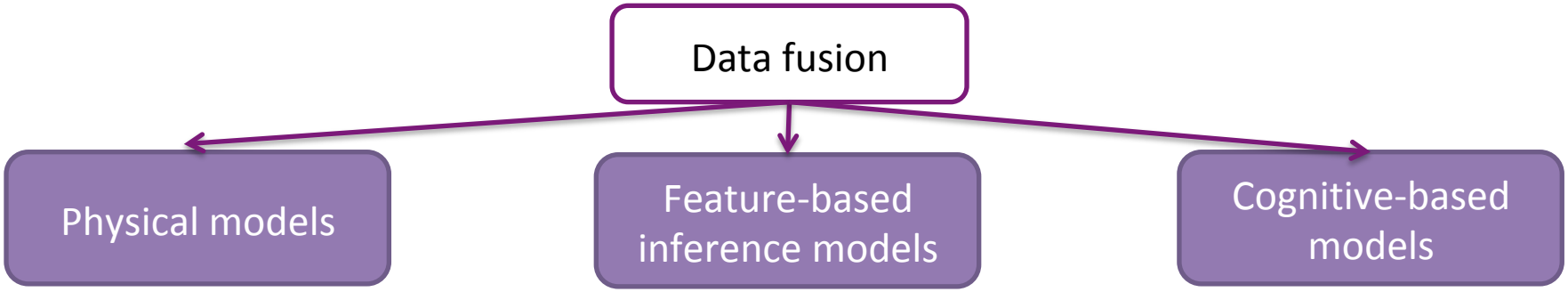
- Motivations
- SWOT Analysis

Part 1- ML-Powered Data Curation

- Record Linkage, Deduplication, Entity Resolution
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

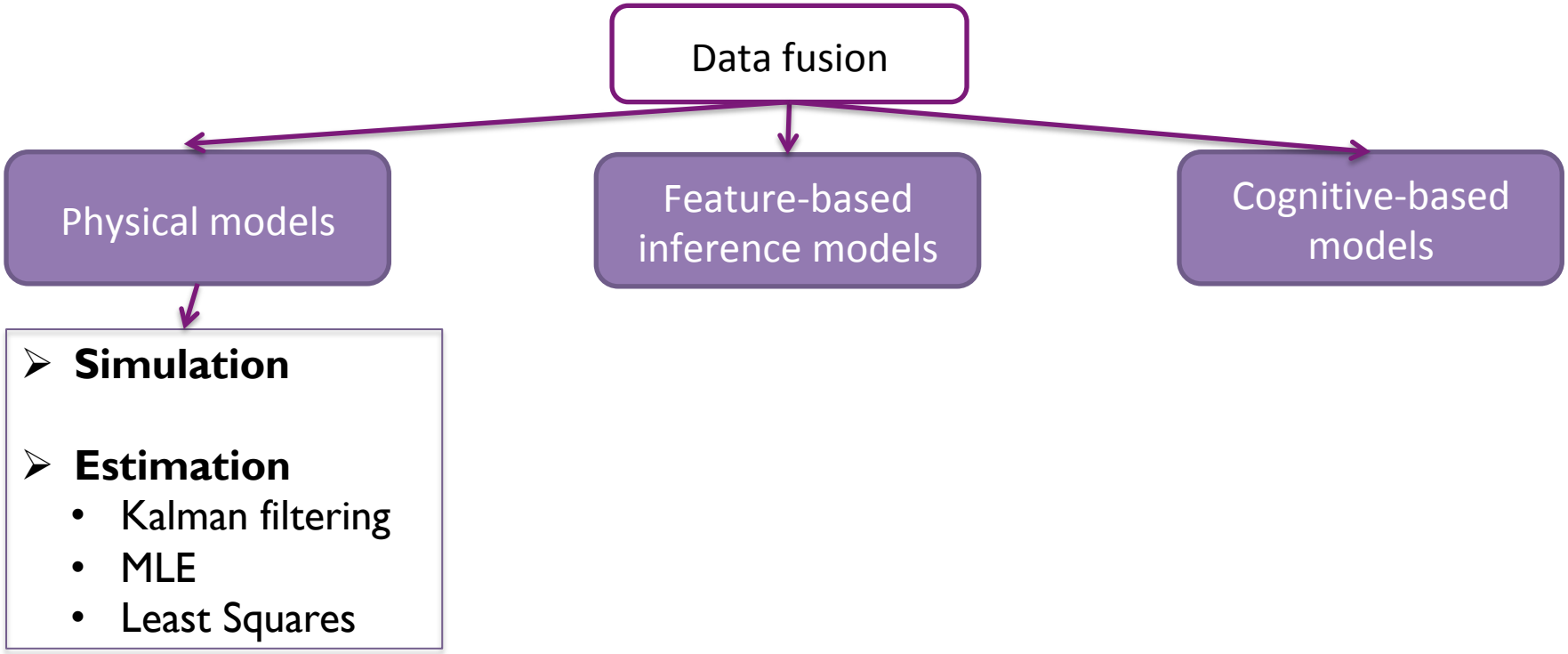
Taxonomy of Data Fusion Techniques

(Not limited to what data fusion means for DB community) [Hall, 1992]



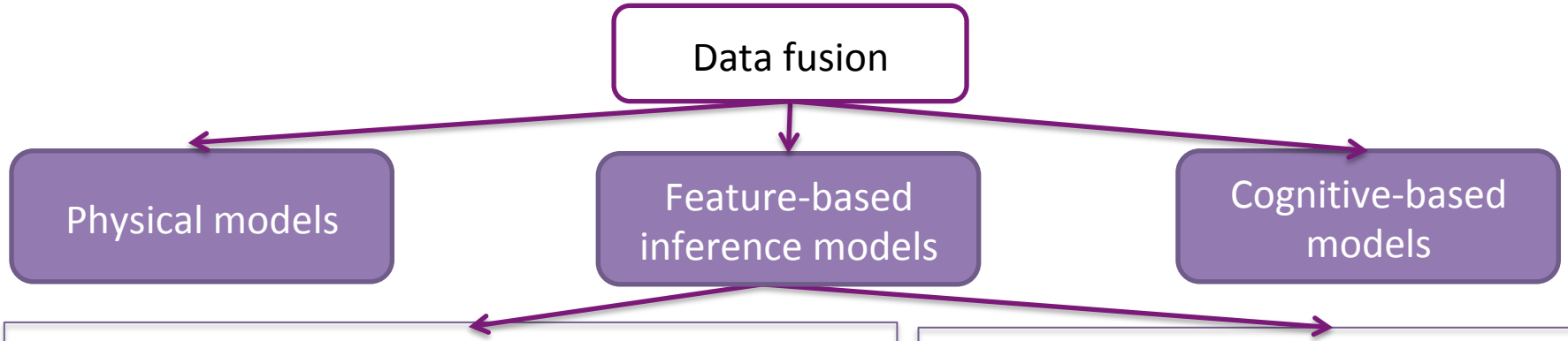
Taxonomy of Data Fusion Techniques

[Hall, 1992]



Taxonomy of Data Fusion Techniques

(Not limited to what data fusion means for DB community) [Hall, 1992]



➢ **Parametric**

- *Statistical-based algorithms*
  - Classical inference
  - Bayesian
  - Dempster-Schafer

SLiMFast

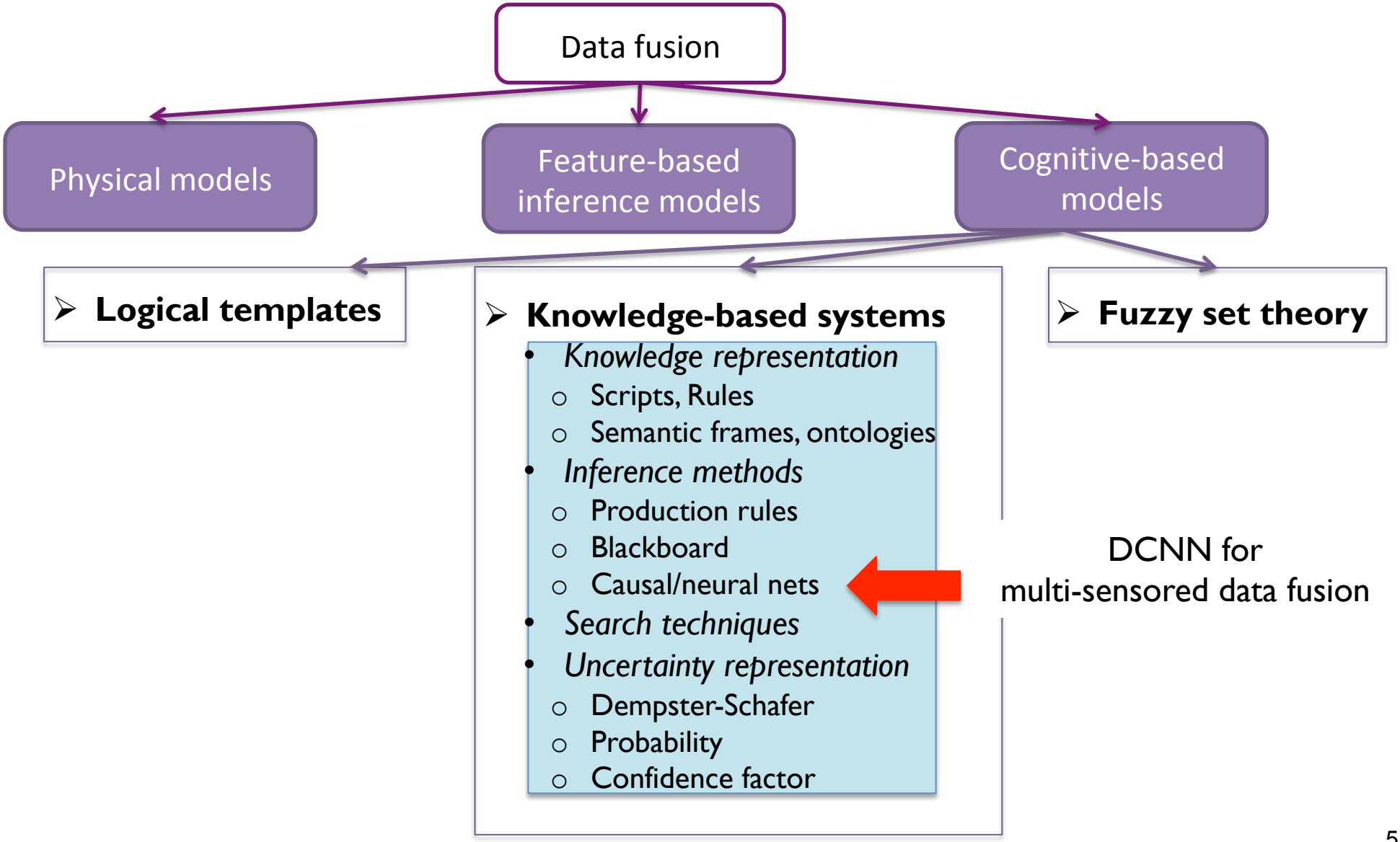
- *Cluster algorithms*
  - Hierarchical Agglomerative (k= Ward's method)
  - Hierarchical Divisive
  - Iterative Partitioning (k=Hill climbing)
  - Density search
  - Factor analytic
  - Clumping
  - Graph theoretic

➢ **Non-Parametric**

- Adaptive neural nets
  - Binary input nets (Hopfield)
  - Continuous valued input (perception)
- Entropic techniques
- Pattern matching / templating
- Figure of merit
- Measure of correlation
- Thresholding logic
- Heuristic methods : voting, scoring, ranking, consensus methods for conflict resolution [Dong, Naumann, Tutorial VLDB'09]

Taxonomy of Data Fusion Techniques

(Not limited to what data fusion means for DB community) [Hall, 1992]



SLiMFast: Probabilistic Models for Data Fusion

[Joglekar et al., SIGMOD'17]

User-specified Input

Source Observations

Source	Object ID	Value
A1	GIGYF2, Parkinson	False
A1	GBA, Parkinson	True
A2	GIGYF2, Parkinson	True
A3	GIGYF2, Parkinson	False
A3	GBA, Parkinson	True

Domain Features

Source	Feature
A1	PubYear=2009
A1	Citations=34
A2	PubYear=2008
A2	Citations=128
A3	Study=GWAS

Ground Truth

Object ID	Value
GBA, Parkinson	True

Output

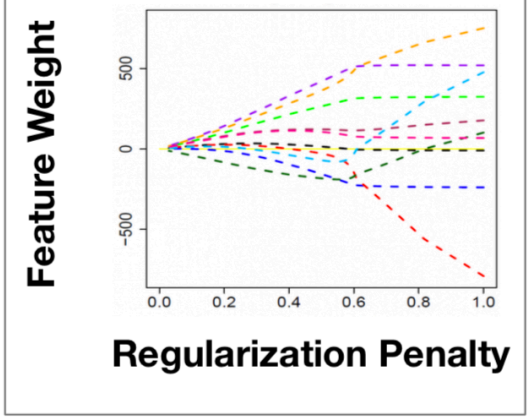
Truth Discovery

Object ID	True Value
GIGYF2, Parkinson	False
GBA, Parkinson	True

Source Accuracy

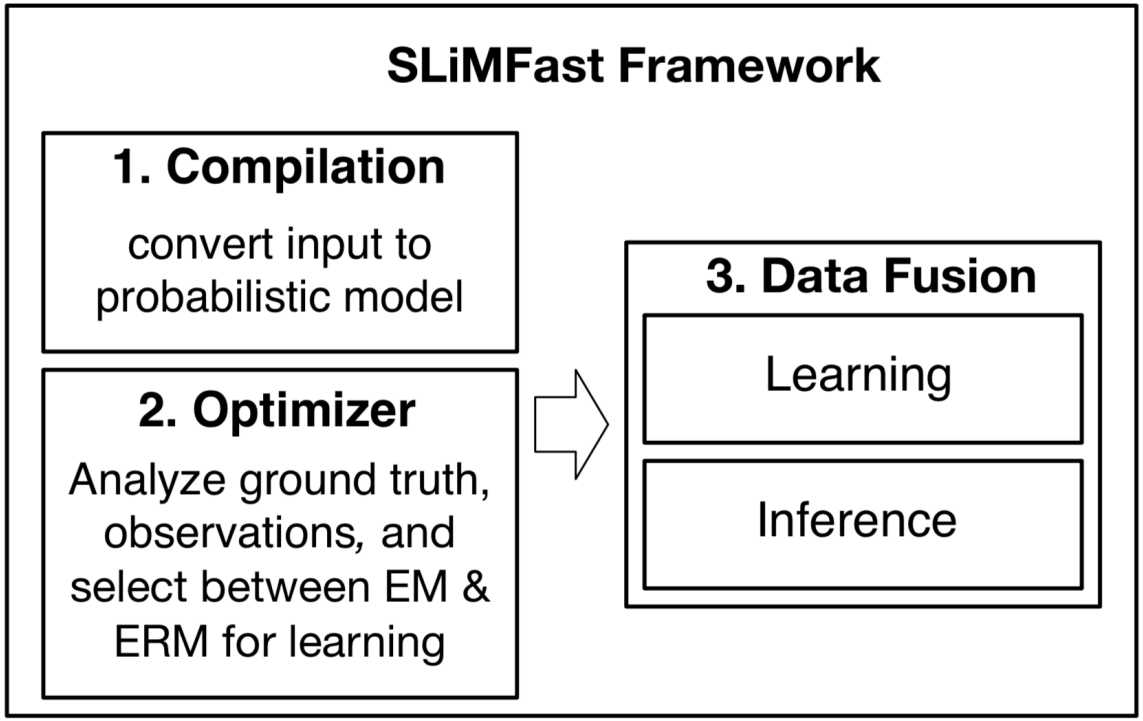
Source	Feature
A1	0.94
A2	0.71
A3	0.85

Source Accuracy Analysis



SLiMFast: Probabilistic Models for Data Fusion

[Joglekar et al., SIGMOD'17]



Reliability scores (model parameters)

$$P(T_o = d|\Omega) = \frac{1}{Z} \exp \sum_{(o,s)\in\Omega} \sigma_s \mathbb{1}_{v_{o,s}=d}$$

Normalizing constant (valid distribution)

$$\sigma_S = \log\left(\frac{\text{Accuracy of Source S}}{\text{1-Accuracy of Source S}}\right)$$

Accuracy = Probability that a source is correct

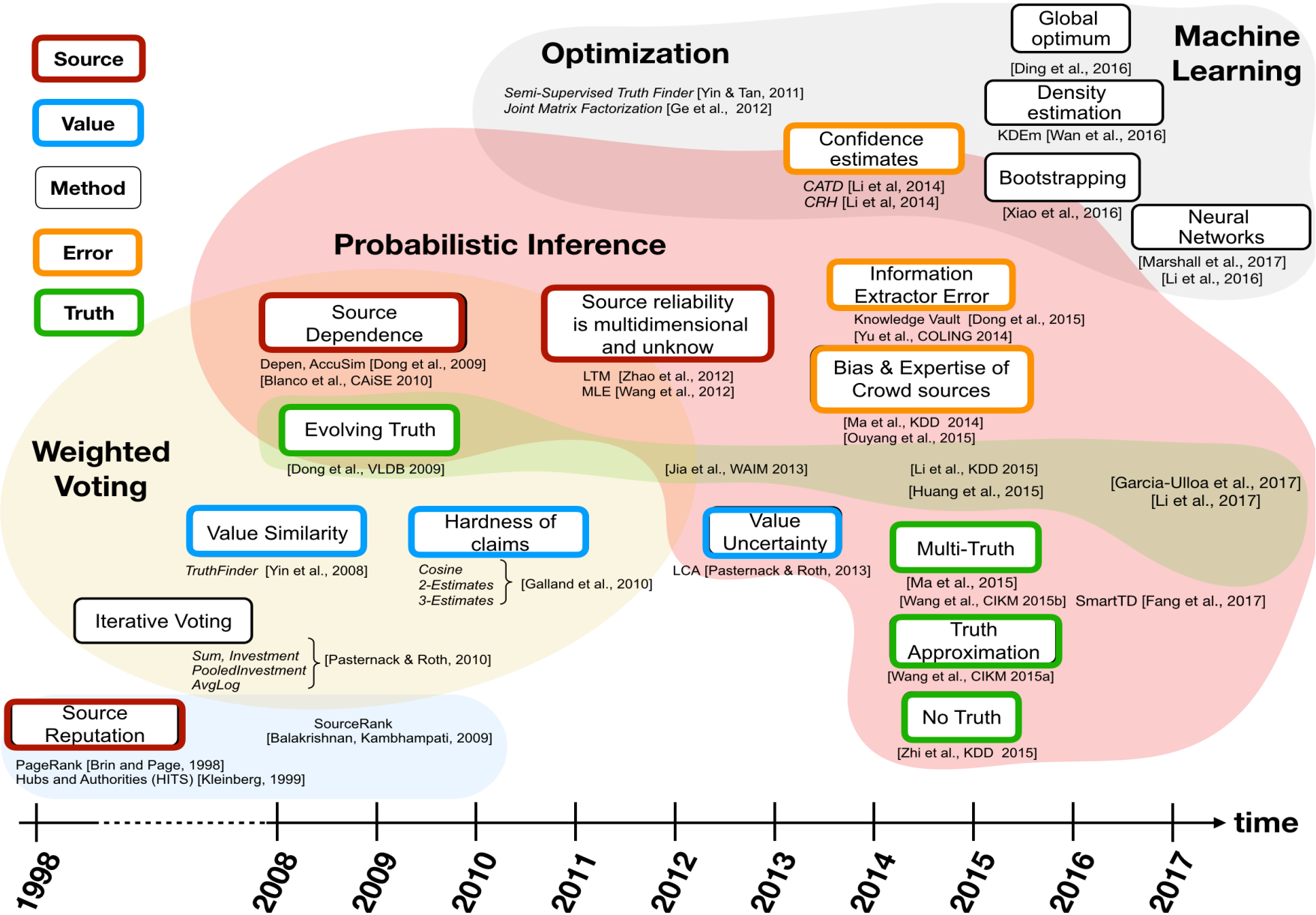
$$A_s = 1/(1+\exp(-w_s - \sum_{k\in K} w_k f_{s,k}))$$

To solve data fusion, SLiMFast :

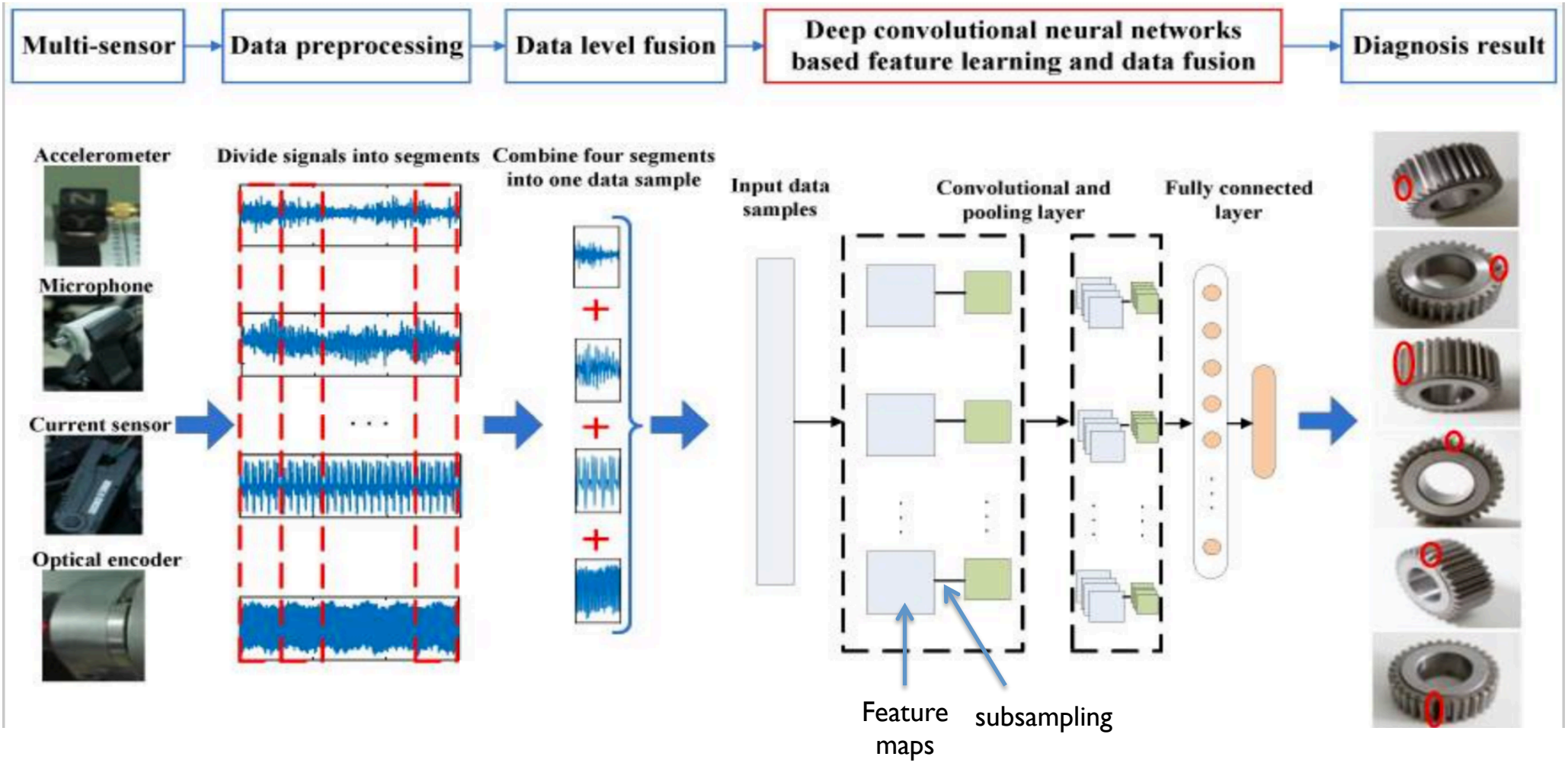
- learns the parameters w of the logistic regression model by optimizing the likelihood $l(w) = log\ P(T|\Omega; w)$ where T corresponds to the set of all variables To,
- infer the maximum a posteriori (MAP) assignments to variables To using ERM (ground truth) or EM (source observation overlap, avg accuracy of sources)

Data fusion and truth finding evolution

[Berti-Equille, Encyclopedia 2018]



Multi-Sensor Data Fusion for Fault Diagnosis using DCNN [Luyang et al., Sensors'17]



Outline

Introduction

- Motivations
- SWOT Analysis

Part 1- ML-Powered Data Curation

- Record Linkage, Deduplication, Entity Resolution
- Error Repair and Pattern Enforcement
- Data and Knowledge Fusion
- Concluding Remarks and Open Issues

Concluding Remarks – Part 1

- ML provides a principled framework and efficient tools for optimizing many DM tasks
- ML crucially needs principled data curation
- However, some tasks require **Humans in the loop**
- There are many opportunities for:
  - Cool ML applications to data management
  - Revisiting DB technology **with** and **for** ML
  - Managing and orchestrating human/machine resources

Open Issues – Part I

- **Usability**:
  - To consider Humans as resources
  - To be understood, interpreted, and trusted by Humans
  - To ease/self-adapt the design, tuning, and use
- **Efficiency:**
  - Runtime
  - Incremental
- **Accuracy:**
  - Reduce impact of dirty data
  - Augmenting the training set
  - Ensembling

Usability (1): Humans as Resources

Challenge 1: **Adjusting "Human-in-the-Loop"**

- Seamless integration of humans as resources for ML-powered DM
- "Taskify" and minimize the amount of interactions with the users while, at the same time, maximize the potential "ML benefit" for selecting/cleaning/labeling training data and other data management tasks

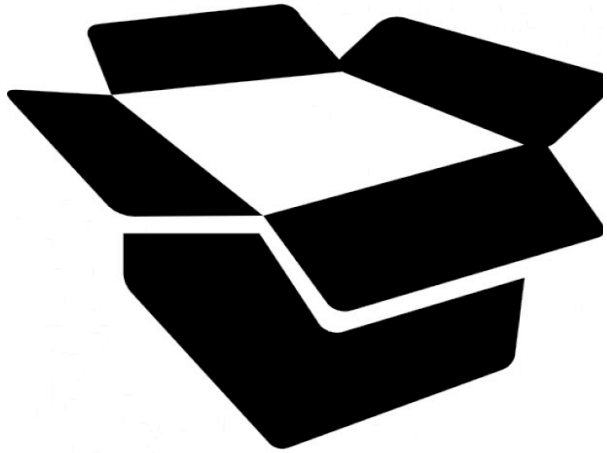
- **Current efforts: Crowdsourcing and active learning**
  - Data cleaning with oracle crowds [Bergman et al., SIGMOD'15]
  - Entity resolution: CrowdER [Wang et al., VLDB'12], Corleone [Gokhale, et al., SIGMOD'14]
  - Data fusion and truth inference [Zheng et al., VLDB'17]
- **Direction**:
  - Adaptive and quality-driven orchestration of Humans and Tools for ML-powered DM

Usability (2): Building trust

Challenge 2: Open the "Black-Box" and customize it

- Improve the interpretability of ML-based decisions
- Build the trust: ML-based decisions should be interpretable, explainable, reproducible to be trusted
- Adapt ML-based DM to on-demand, incremental, progressive tasks

- **Current efforts:**
  - Trusted Machine Learning [Ghosh et al., AAAI'17]
  - Model-Agnostic Explanations [Ribeiro et al., KDD'16]
  - On-demand ETL [Yang et al., VLDB'15]
  - ActiveClean [Krishnan et al., VLDB'16]
  - Continuous cleaning for considering incremental changes to the data and to the constraints [Volkovs et al., ICDE'14]

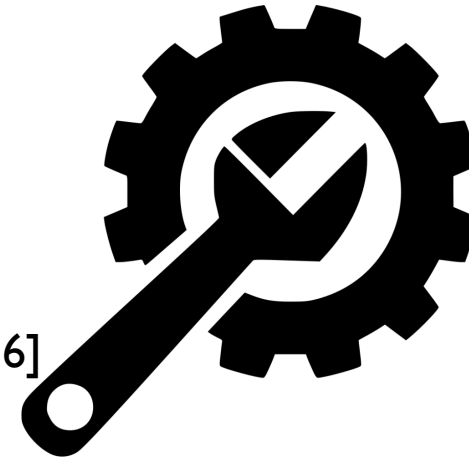


- **Directions**:
  - Causality and explanations in ML-based DM and their effective representation
  - Reversibility and repeatability
  - Data privacy/security: What if adversarial learning is applied ?

Usability (3) : Easy to build, tune, and test

Challenge 3: **Engineering ML-based DM applications**

- Model building and feature selection
- Model interoperability and model selection



- **Current efforts:**
  - Systematizing/optimizing model selection
    [Kumar, Boehm, Yang, SIGMOD'17 Tutorial],
    MSMS [Kumar et al., SIGMODRec'15], Zombie [Anderson et al., 2016]
  - Declarative ML tasks
  - Interactive model building: Ava [John et al., CIDR'17], Vizdom [Crotty et al., VLDB'15]
  - Meta-learning, bandit techniques
  - PMML, ONNX, PFA for model interoperability
- **Directions**:
  - Analysis of dependability of models
  - Model debugging, versioning, and management (e.g., for large models)
  - Managing ML model provenance and elicitation
  - Transfer pre-trained models from task-/domain-agnostic to *-specific DM

Efficiency

- **Challenge 4: Incremental ML application to DM**
  - When we have more training data or refresh/delete some data (obsolete), shall we retrain ML model from scratch? Can we do incremental training/learning? For what cost/trade-off?
- **Challenge 5: Runtime ML-based DM**
  - Could we orchestrate and optimize data annotation and preprocessing tasks? Design cost models, candidate plans?
  - To what extent could we use transfer learning to reduce training data collection/preprocessing cost ?

Accuracy (1)

- **Challenge 6: Reduce the impact of dirty data**

  Glitch types and their distributions can be very different in the datasets used for training, testing, and validation and they affect accuracy of ML models in different ways:

  - How could we capture the good, the bad and the ugly combinations?
  - Should we robustify the ML algorithms or/and the data curation? Would both be inevitably better/ necessary?

  – **Find optimal data cleaning strategies for a given ML-based DM application**

  - Can we predict the ±delta in ML accuracy that a given data curation strategy brings to the model?

Accuracy (2)

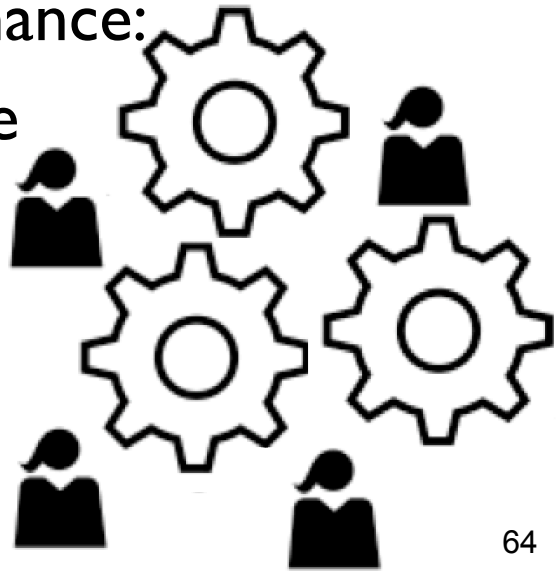
- **Challenge 7: Synthetic training data generation**

  Copy/Transform existing labeled data to augment the training set
  [Ratner et al., NIPS'17]

- **Challenge 8: Model/Feature recommendation and ensembling**

  Many ML models can be parameterized, applied and combined in different ways leading to various quality performance:

  - Could we define a predictive scoring of the models and their ensembles ?
  - Would ensembling be (inevitably) better?



Thanks!

References - Part 1 (1)

[Anderson et al., 2016] http://www.vldb.org/pvldb/vol7/p1657-anderson.pdf
[Arasu et al., SIGMOD'10] https://dl.acm.org/citation.cfm?id=1807252
[Assadi, Milo, Novgorodov, WebDB'18] http://slavanov.com/research/webdb18.pdf
[Bahmani et al., SUM'15] https://arxiv.org/pdf/1602.02334.pdf
[Battacharya, Getoor, TKDD'07] https://dl.acm.org/citation.cfm?id=1217304
[Bellare et al., KDD'12] http://ilpubs.stanford.edu:8090/1036/1/main.pdf
[Bergman et al., SIGMOD 2015] http://www.vldb.org/pvldb/vol8/p1900-bergman.pdf
[Berti-Equille, Encyclopedia 2018] Encyclopedia of Big Data Technologies, Springer (To Appear), 2018
[Biggio et al., ICML'12] https://icml.cc/Conferences/2012/papers/880.pdf
[Bilenko et al., ICDM'06] http://ieeexplore.ieee.org/document/4053037/
[Bilenko, Mooney, KDD'03] https://dl.acm.org/citation.cfm?id=956759
[Chaudhuri et al., ICDE'05] http://ieeexplore.ieee.org/document/1410199/
[Chaudhuri et al., VLDB'07] http://www.vldb.org/conf/2007/papers/research/p327-chaudhuri.pdf
[Chen et al., SIGMOD'09] https://dl.acm.org/citation.cfm?id=1559869
[Christen et al., 2002] http://users.cecs.anu.edu.au/~christen/publications/adm2002-cleaning.pdf
[Christen, 2012] http://ieeexplore.ieee.org/document/5887335/
[Churches et al., 2002] http://www.biomedcentral.com/1472-6947/2/9/
[Crotty et al., VLDB'15] http://www.vldb.org/pvldb/vol8/p2024-crotty.pdf
[Dean, NIPS 2017] http://learningsys.org/nips17/assets/slides/dean-nips17.pdf
[Doan et al., HILDA@SIGMOD'17] http://pages.cs.wisc.edu/~anhai/papers1/hil-in-em-hilda17.pdf
[Dzakovic, XLDB'18] https://conf.slac.stanford.edu/xldb2018/event-information/lightning-talks
[Ebraheem et al., Arxiv 2017] https://arxiv.org/pdf/1710.00597.pdf
[Fellegi, Sunter, 1969] A theory for record linkage. J. Am. Stat. Assoc. 1969;64(328):1183–210.
[Fisher et al., KDD'15] https://dl.acm.org/citation.cfm?id=2783396
[Getoor, Machanavajjhala, Tutorial VLDB' 12] http://legacydirs.umiacs.umd.edu/~getoor/Tutorials/ER_VLDB2012.pdf
[Gokhale et al., SIGMOD'14] https://dl.acm.org/citation.cfm?id=2588576
[Gosh et al., AAAI'17] https://aaai.org/ocs/index.php/WS/AAAIW17/paper/download/15206/14765

References - Part 1 (2)

[Gupta, Sarawagi, VLDB'09] https://dl.acm.org/citation.cfm?id=1687627.1687661

[Hall, 1992] Mathematical Techniques in Multisensor Data Fusion, ArtechHouse, 1992

[Hassanzadeh et al., PVLDB'09] http://www.vldb.org/pvldb/2/vldb09-1025.pdf

[Hu et al, 2017] http://users.cecs.anu.edu.au/~u5170295/papers/pakdd-hu-2017.pdf

[Ilyas, Chu, 2015] https://cs.uwaterloo.ca/~ilyas/papers/IlyasFnTDB2015.pdf

[John et al., CIDR'17] http://pages.cs.wisc.edu/~jignesh/publ/Ava.pdf

[Joglekar, et al., SIGMOD'17] https://dl.acm.org/citation.cfm?id=3035951

[Kooli et al., ACIIDS'18] https://link.springer.com/chapter/10.1007%2F978-3-319-75420-8_1

[Köpcke et al., VLDB'10] http://www.vldb.org/pvldb/vldb2010/papers/E04.pdf

[Koudas, Srivastava, Sarawagi, Tutorial SIGMOD'06] http://www.cs.toronto.edu/~koudas/docs/aj.pdf

[Kraska et al. 2017] https://arxiv.org/abs/1712.01208

[Krishnan et al., VLDB'16] http://www.vldb.org/pvldb/vol9/p948-krishnan.pdf

[Krishnan et al., 2017] https://arxiv.org/pdf/1711.01299.pdf

[Kumar et al., SIGMODRec'15] https://adalabucsd.github.io/papers/2015_MSMS_SIGMODRecord.pdf

[Kumar, Boehm, Yang, SIGMOD'17 Tutorial] https://adalabucsd.github.io/papers/2017_Tutorial_SIGMOD.pdf

[Luyang et al., Sensors'17] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5335931/

[Marchand, Rubinstein, VLDB'17] http://www.vldb.org/pvldb/vol10/p1322-rubinstein.pdf

[Marcus, Arxiv, 2018] https://arxiv.org/ftp/arxiv/papers/1801/1801.00631.pdf

[Mintz et al., 2009] https://dl.acm.org/citation.cfm?id=1690287

[Natarajan et al., NIPS'13] https://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf

[Papadakis et al., TKDE 2013] http://ieeexplore.ieee.org/document/6255742/

[Papadakis, Palpanas, Tutorial ICDE'16] http://www.mi.parisdescartes.fr/~themisp/publications/tutorialIcde16.pdf

[Polyzotis et al., SIGMOD'17] https://dl.acm.org/citation.cfm?id=3035918.3054782

[Qian et al., CIKM'17] https://dl.acm.org/citation.cfm?id=3132949

References - Part 1 (3)

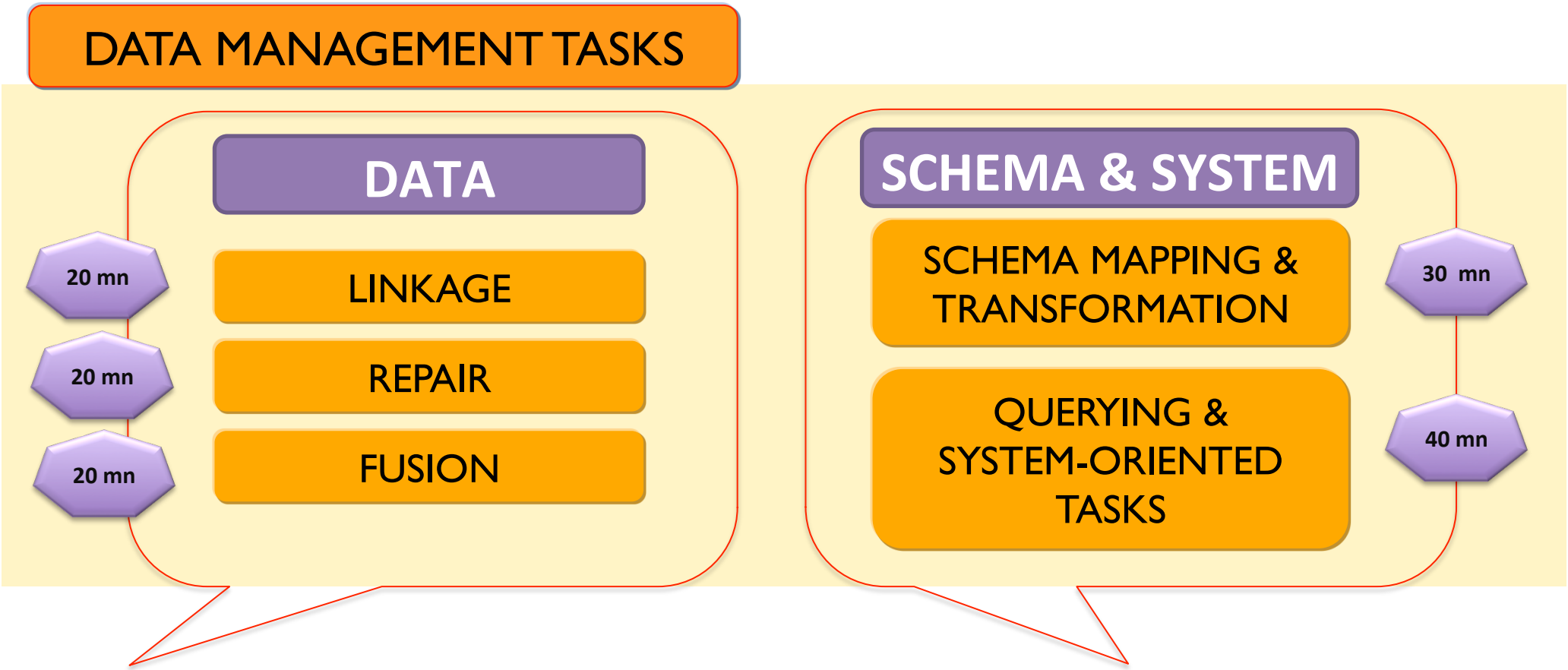
[Ratner et al., NIPS'17] https://arxiv.org/pdf/1709.01643.pdf
[Rekatsinas et al., VLDB'17] http://www.vldb.org/pvldb/vol10/p1190-rekatsinas.pdf
[Ribeiro et al., KDD'16] http://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf
[Sarawagi, Bhamidipaty, KDD'02] https://dl.acm.org/citation.cfm?id=775087
[Sharma et al. 2018] https://arxiv.org/abs/1801.05643
[Shin et al., 2015] http://www.vldb.org/pvldb/vol8/p1310-shin.pdf
[Singla, Domingos, PKDD'05] http://alchemy.cs.washington.edu/papers/pdfs/singla-domingos05b.pdf
[Tang, KDD'17 tutorial] https://sites.google.com/site/pkujiantang/home/kdd17-tutorial
[Tejada et al. KDD'02] https://dl.acm.org/citation.cfm?id=775099
[Vesdapunt et al., VLDB'14] http://www.vldb.org/pvldb/vol7/p1071-vesdapunt.pdf
[Volkovs et al., ICDE'14] http://www.cs.toronto.edu/~mvolkovs/icde14_data_cleaning.pdf
[Wang et al., SIGMOD'13'] https://dl.acm.org/citation.cfm?id=2465280
[Wang et al., VLDB'12] http://vldb.org/pvldb/vol5/p1483_jiannanwang_vldb2012.pdf
[Wu et al. SIGMOD'18] https://arxiv.org/pdf/1703.05028.pdf
[Xiao et al., Neurocomputing 2014] https://www.sciencedirect.com/science/article/pii/S0925231215001198
[Yakout, Berti-Equille, Elmagarmid SIGMOD'13] https://dl.acm.org/citation.cfm?id=2463706
[Yang et al., VLDB'15] http://www.vldb.org/pvldb/vol8/p1578-yang.pdf
[Zheng et al., VLDB'17] http://www.vldb.org/pvldb/vol10/p541-zheng.pdf

ML to Data Management: A Round Trip

PART II

Angela Bonifati

Our Focus: ML applications to DM



Tutorial Part I
(morning)

Tutorial Part II
(afternoon)

Outline

Part II- ML-Powered Data Integration

- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Outline

Part II- ML-Powered Data Integration

- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

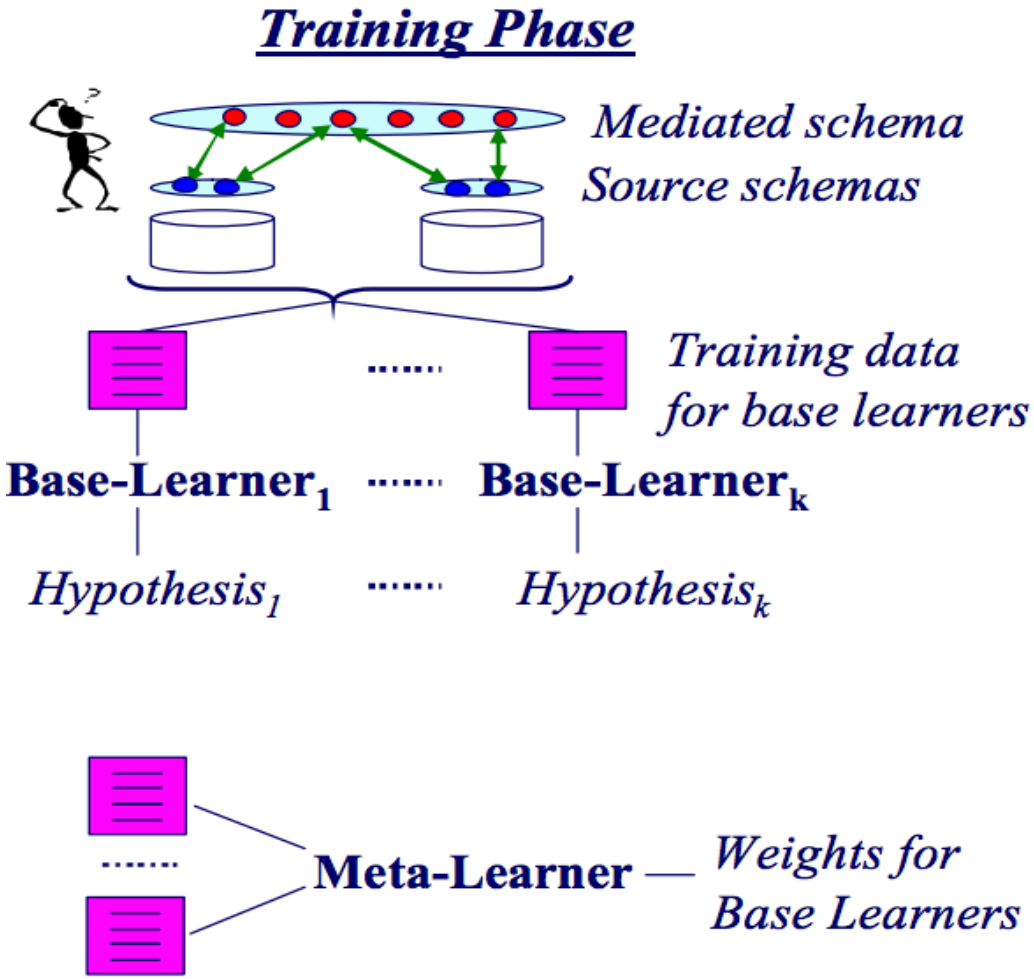
- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Schema Matching and ML

- Schema matching is the process of identifying semantic correspondences between schema elements (a common problem to DB, AI, KR)
- Such correspondences can be arbitrarily complex (1-1, 1-m, n-m) and have a confidence value [0..1]
- Representative ML-based schema matching approaches include:
  - LSD [Doan et al. Sigmod01]
  - GLUE [Doan et al. WWW02]
  - SemInt [Li & Clifton, DKE00]
  - Automatch/Autoplex [Berlin et al. Caise02]

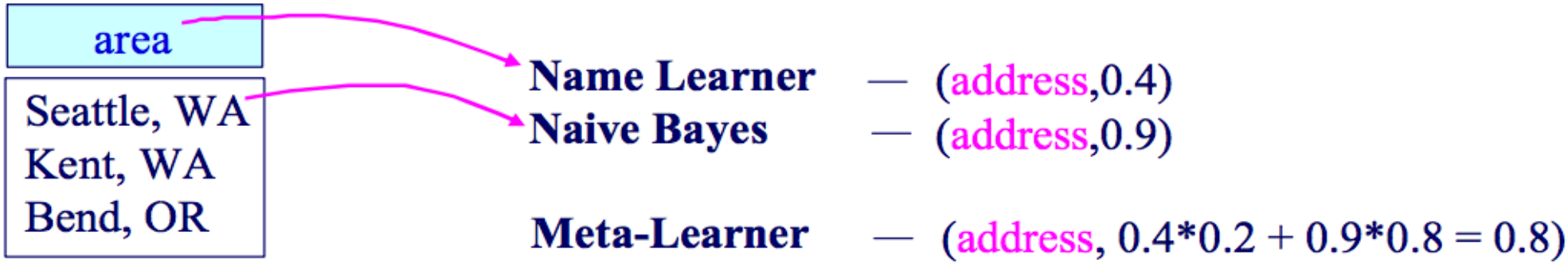
The LSD Approach

- Multi-strategy learning with different base learners (one for schema elements, one for instances)
- Combines them in a Meta-Learner
- Leverages 'stacking' to learn weights of the different learners in the Meta-Learner
- Training involves a few data sources



Stacking as a multi-learning technique

- Training
  - uses training data to learn weights
  - one weight for each (base-learner,mediated-schema element) pair
  - E.g. weight (Name-Learner,address) = 0.2 (on schema-element name)
  - E.g. weight (Naive-Bayes,address) = 0.8 (on schema-element value)
- Matching: combine predictions of base learners



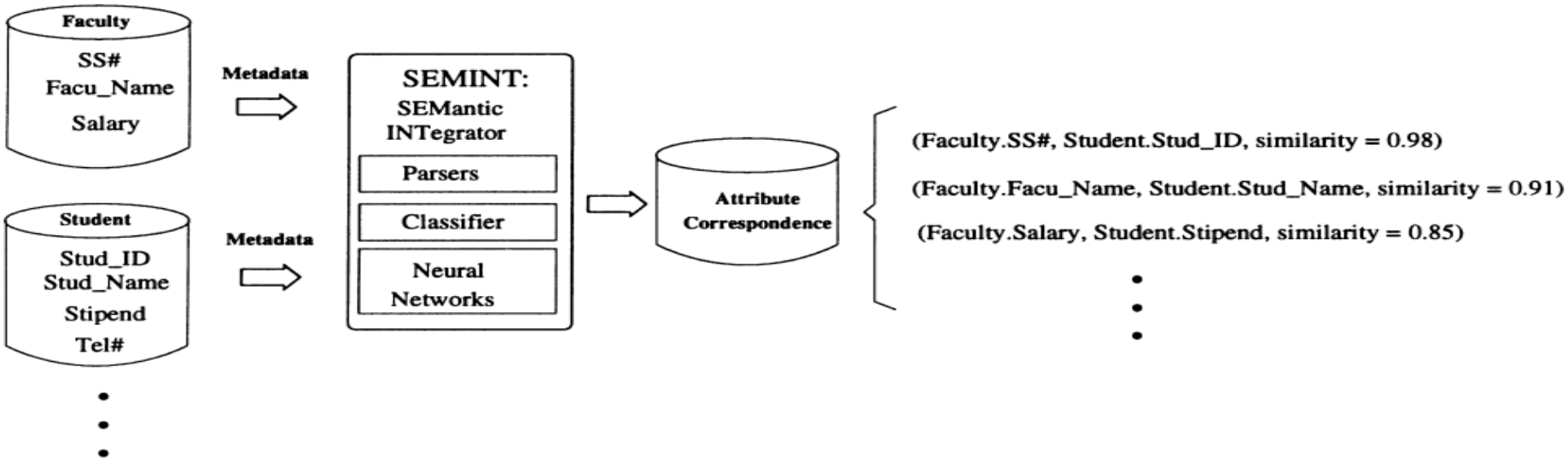
GLUE: Learning to find similar ontological concepts

- Glue applies ML technique to find, for each concept node in a taxonomy, the most similar concept in the other taxonomy
- It leverages the joint probability distribution:
  - P(A,B), P(A,not(B)),P(not(A),B), P(not(A),not(B))
- ML is used to infer whether P(A,B) can be approximated with P (A intersect B)
  - By defining a classifier for instances containing concept A (B) and using it to classify instances of B (A)
- It applies the multi-learning approach of LSD

SEMINT

- It leverages the DBMS specific parsers to extract metadata (schema elements, constraints etc.)
- Such metadata is given as input to neural networks in order to feed the learning process
- Matching is done during the training process

W.-S. Li, C. Clifton / Data & Knowledge Engineering 33 (2000) 49–84



AutoMatch

- It leverages probabilistic knowledge from schema examples "mapped" by domain experts into an attribute dictionary (based on Bayesian learning)
- Given a pair of "client" schemas that need to be matched, Automatch matches them "through" its dictionary and uses the Minimum Cost Maximum Flow network algorithm to find the optimal matching
- Automatch employs statistical feature selection techniques to learn an efficient representation of the examples (as few as 10% of the initial values are employed).

Motro et al. "Automatch Revisited". Seminal Contributions to Information Systems Engineering 2013:
Domingos et al. " Conditions for the optimality of the simple bayesian classifier" ICDM96

Schema Mapping and ML

- Schema mapping is the process of identifying schema transformations expressed in fragments of FO logics and to use them to compute the solution of the transformation
- The transformations are expressed as source-to-target dependencies (logical assertions with CQs on both sides and existential variables in the RHS)
- Recent ML-based schema mapping approaches include:
  - CMD [Kimmig et al., ICDE'17]
  - GAV Learn [ten Cate et al. , PODS'18]

CMD: Probabilistic Schema Mapping

- Probabilistic approaches to schema mapping rely on probabilistic modeling and statistical relational learning (SRL) [2].
- Specifically, Collective Mapping Discovery[1] encodes the mapping selection objective as a program in probabilistic soft logic (PSL)
- It uses as input metadata (under the form of a set of candidate s-t tgds) and potentially imperfect evidence (in the form of a data example) to select an optimal mapping

[1] Kimmig et al. "Collective, Probabilistic Approach to Schema Mapping", ICDE17
[2] L. Getoor and B. Taskar, Eds., *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

CMD Objective function

- The goal is to minimize a cost function containing the size (#atoms of M, the # of unexplained atoms in the target, and the # of erroneous tuples)
- Providing a discrete solution to the CMD optimization problem is NP-hard, thus an approximate solution with theoretical guarantees is proposed

$$\underset{\mathcal{M}\subseteq\mathcal{C}}{\operatorname{argmin}}\Big(\sum_{t\in J}[1-\mathsf{explains}_{\mathsf{full}}(\mathcal{M},$$
$$+\sum_{t\in K_{\mathcal{C}}-J}[\mathsf{error}_{\mathsf{full}}(\mathcal{M},t)]$$
$$+\mathsf{size}_{\mathsf{m}}(\mathcal{M})\Big)$$

GAV Learn
(Active Learning for GAV Mappings)

- The goal is to derive a syntactic specification of a GAV mapping from a given set of data examples and from a “black-box" implementation (i.e. the oracle, a special type of user).
- GAVLearn relies on the following fact:
  - GAV mappings are polynomial-time learnable in Angluin’s model of exact learning with membership/equivalence queries.
- GAVLearn is an active learning algorithm
  - it accomplishes its task by “actively doing experiments (tests) on the software"

A Condensed View

TOOL NAME	ML APPROACH	GOAL
LSD	Multi-strategy Learning	Schema Matching
Glue	Multi-strategy Learning	Ontology Matching
Automatch	Bayesian networks	Schema Matching
SemInt	Neural networks	Schema Matching
CMD	Statistical Relational Learning	Schema Mapping
GAV Learn	Active Learning	Schema Mapping

Outline

Part II- ML-Powered Data Integration

- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Constraint Discovery with ILP

- [Flach et al.,AIComm00] focus on the problem of using Inductive Logic Programming to FD/MVD discovery in relational databases
  - Bottom-up ILP algorithms: take the instances for hypothesis construction
  - Top-down ILP algorithms: adhere to a generate-and-test approach
- They rely on generality ordering on the space of all possible definitions:
  - a predicate definition is more general than another if the least Herbrand model of the first is a model of the second (i.e. the first entails the second)
- Three dependency induction algorithms: TD, Bidirectional, BU

Top-Down Algorithm and pros of ILP

- An agenda-based search algorithm
  - Input: a relation r
  - Output: a cover of DEP(r)
  - Initialise: set of the most general dependencies (from most general to most specific)

```
begin
    DEPS := ∅;
    Q := initialise(R);
    while Q ≠ ∅
    do D := next item from Q; Q := Q − D;
        if some witnesses t1, . . . , tn from r violate D
        then Q := Q ∪ spec(R,D,t1, . . . , tn)
        else DEPS := DEPS ∪ {D}
        fi
    od
    output DEPS
end.
```

- ILP leads to obtain:
  - interpretable results
  - in-DBMS implementation and scalable execution (QuickFoil [Zeng et Al., PVLDB14])

A ML Approach to FK Discovery

- Underlying assumption [Rostin et al, WebDB'09]:
  - choice of features is more influential on the achievable performance than the choice of classification method
  - extensive manual study to find meaningful features by using common sense and by carefully studying positive and negative examples.
  - Feature derivation for INDs (10 different features among which coverage, columnName, OutOfRange, ValueLengthDiff etc.)

Practical Study on FKs

- Given some real-world biological datasets (SCOP, MSD, UniProt), two movie datasets and the TPC-H benchmark
- Given four ML algorithms in the Weka ML tool (Naive Bayes, SVM, J48 and DT)
  - the study tackles the comparison of
    - Results of different feature selection methods (Ranked search, InfoGain, Randomized Search, X2-statistics)

F-measures of the classifiers

- J48 and DecisionTab obtain the best results in the majority of the cases
- For UniProt, SVM works better than the others

DS for learning / evaluation	*Naive Bayes*	*SVM*	*J48*	*DecisionTab*	*Avg*
D6 / D1	0.86	**0.92**	0.84	0.8	0.855
D7 / D2	0.80	0.86	0.86	**0.93**	0.817
D8 / D3	0.71	0.71	**1.0**	0.8	0.805
D9 / D4	**1.0**	**1.0**	**1.0**	**1.0**	1.0
DA / D5	0.86	0.90	**0.95**	**0.95**	0.915
Average	0.846	0.78	**0.930**	0.896	

Table 4. Results (F-Measure) of four different classifiers on five different datasets. Best results per row are in bold.

The Regex Learning Problem

- It consists of learning a regex expression (on arbitrary size of the alphabet and with no restrictions on the use of Kleene-star and disjunction)
  - Input: a set of positive and negative examples + an initial regular expression (from domain knowledge)
  - Output: the regex with highest F-measure

The ReLIE Algorithm

- ReLIE [Li et al., EMNLP08] is a greedy hill climbing search procedure that chooses, at every iteration, the regex with the highest F-measure.
- An iteration in ReLIE consists of:
  - Applying every transformation on the current regex Rnew to obtain a set of candidate regexes
  - From the candidates, choosing the regex R' whose F-measure over the training dataset is maximum
- To avoid overfitting, ReLIE terminates when either of the following conditions is true: (i) there is no improvement in F-measure over the training set; (ii) there is a drop in F-measure when applying R' on the validation set.
- ReLIE compared with MinorThird (an implementation of CRF) is proved to be superior in most of the cases except a few exceptions (larger training dataset)

A Condensed View

Authors	**ML/AI APPROACH**	**GOAL**
Flach et al. 99	Inductive Logic Programming	FD/IND Discovery
Rostin et al. 09	Naive Bayes, SVM, J48 and DT	FD Discovery
Li et al. 08	Hill-Climbing Algorithm	Regex Expressions Discovery

Outline

Part II- ML-Powered Data Integration

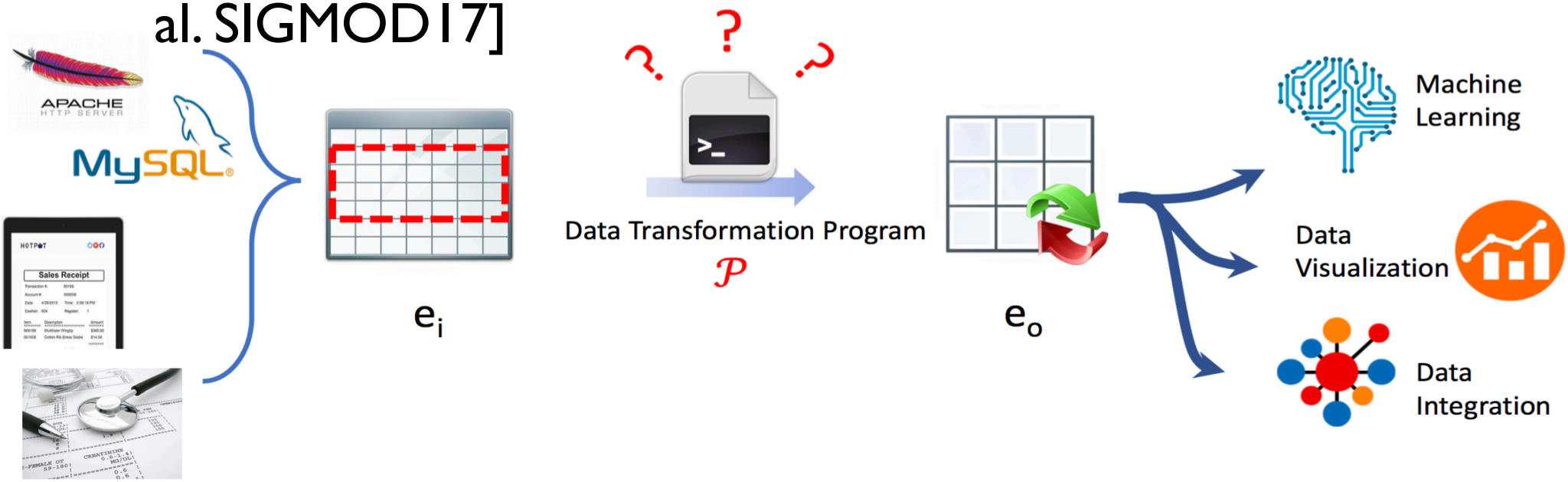
- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Foofah: Synthetising a Data Transformation

- Given as user input a pair E=(e_i, e_o) of sample raw data e_i and transformed view e_o of e_i
  - synthetize a program P that takes E as input
- Leverages program synthesis as a search problem [Jin et al. SIGMOD17]



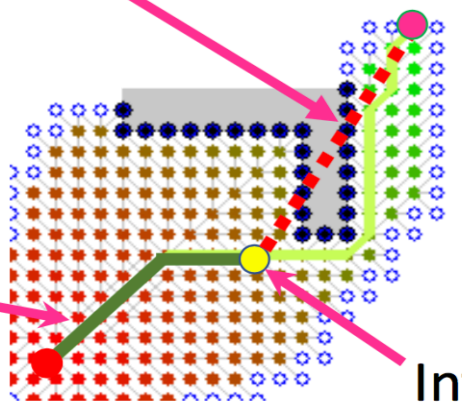
A* Search Algorithm

- A* search algorithm keeps exploring the most promising node - smallest f(n)
- g(n) nr. of Potter's Wheel operations [Hellerstein, 2001]
- h(n) estimate of the latter, or estimate of the nr. of columns or table-edit distance heuristic

- f(n) = g(n) + h(n)

Estimated Part: estimated distance to the goal state

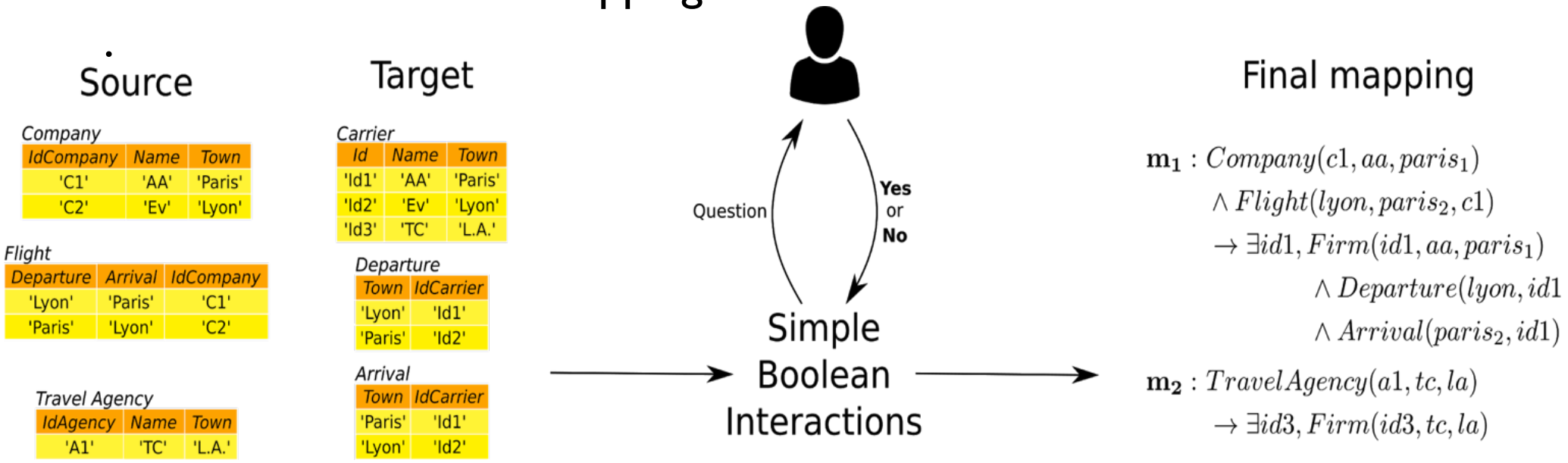
Observed Part: traveled distance for the current state



Intermediate state 'n'

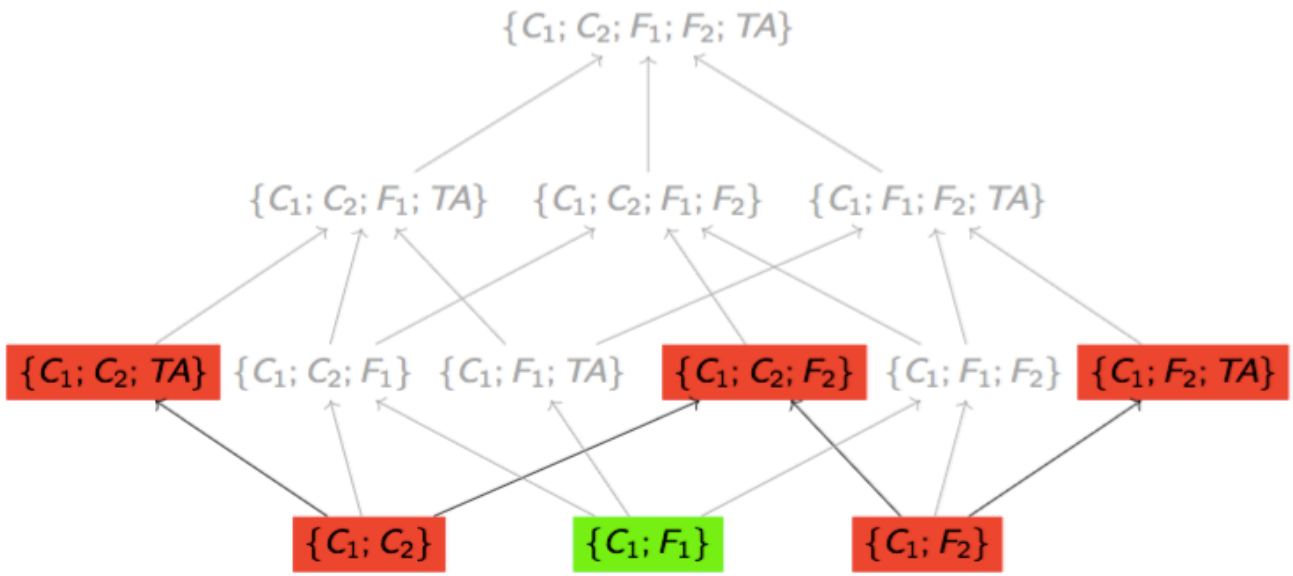
From raw tuples to complex mappings (with the user in the loop)

- Mapping design: from data curators to ordinary users [Bonifati et al. SIGMOD17]
- Allows a user to provide *arbitrary* exemplar tuples.
- (Minimally) Interacts with the user via simple boolean questions in order to discover the mapping that the user has in mind.



Interactive Lattice Exploration

- The user is interactively exploring a lattice of possibilities in which the different reductions of the LHS of the mappings are reported:



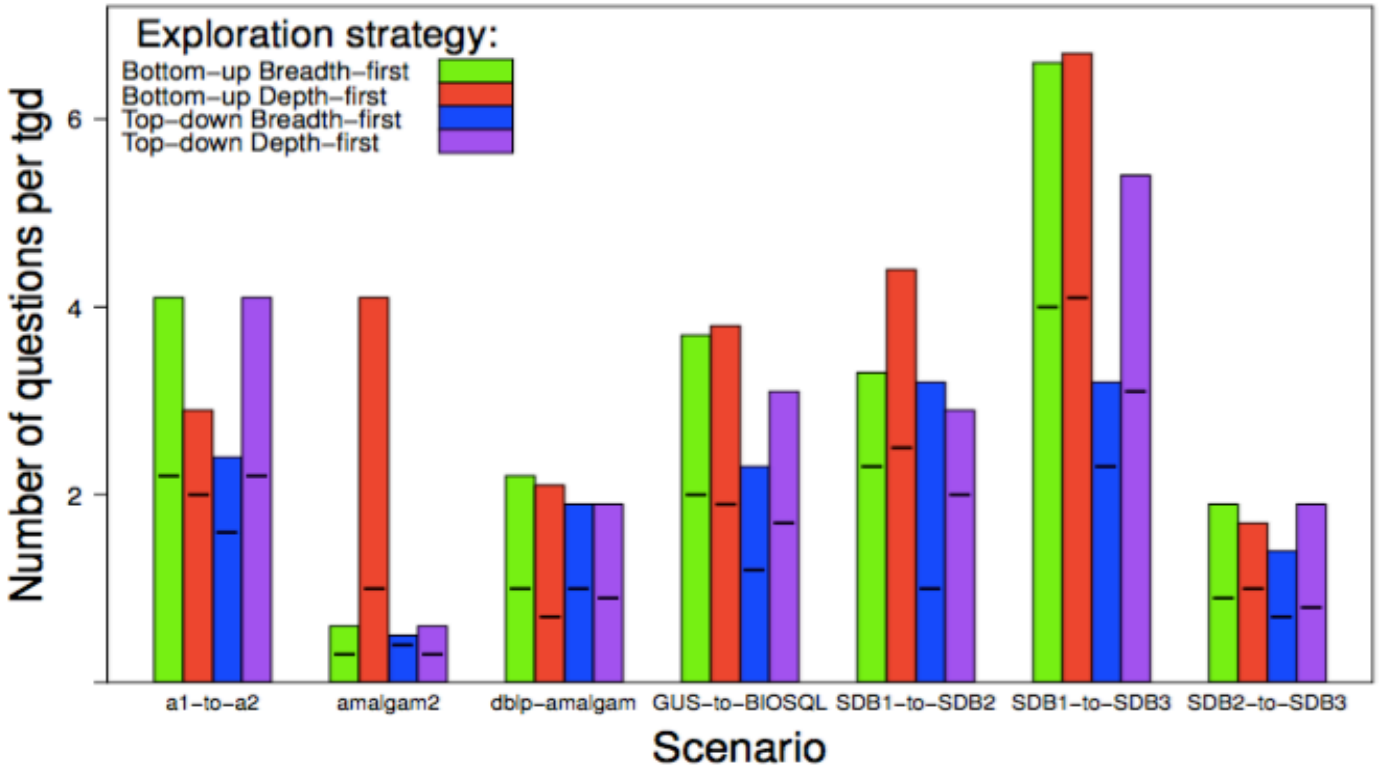
Generated mapping :

$\Sigma = \{Company(\text{c1}, \text{aa}, \text{paris}) \wedge Flight(\text{lyon}, \text{paris}, \text{c1})$

$\Rightarrow \exists \text{id1}, Carrier(\text{id1}, \text{aa}, \text{paris}) \wedge Departure(\text{lyon}, \text{id1}) \wedge Arrival(\text{paris}, \text{id1})\}$

Effectiveness of the Interactive Method

- All exploration strategies keep the number of questions (per tgd) low along atom refinement.



A Condensed View

Tool Name/ Authors	**ML/AI APPROACH**	**GOAL**
Foofah/Jin et al. 2017	A* Search	Raw table transformation discovery
Bonifati et al. 2017	Lattice-based Exploration	Schema mapping discovery

Outline

Part II- ML-Powered Data Integration

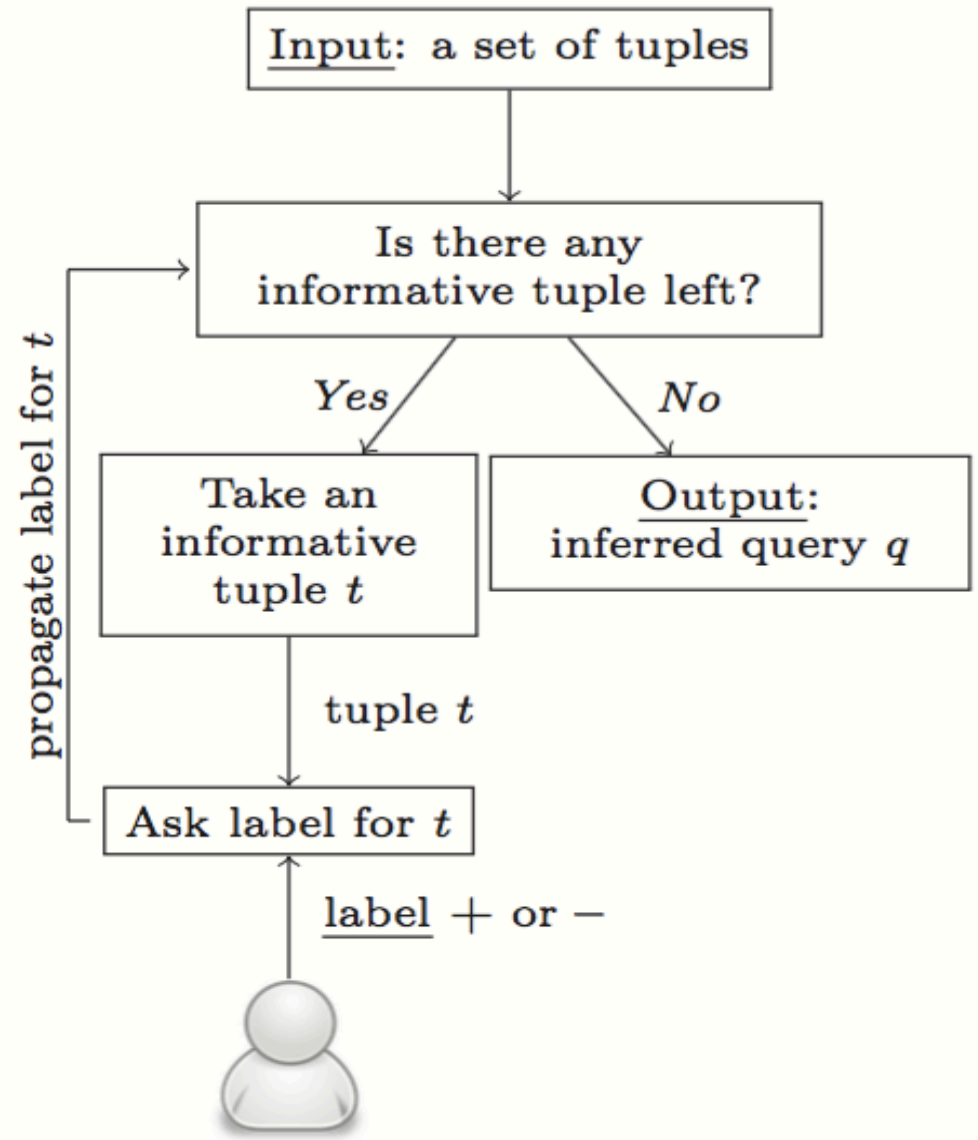
- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

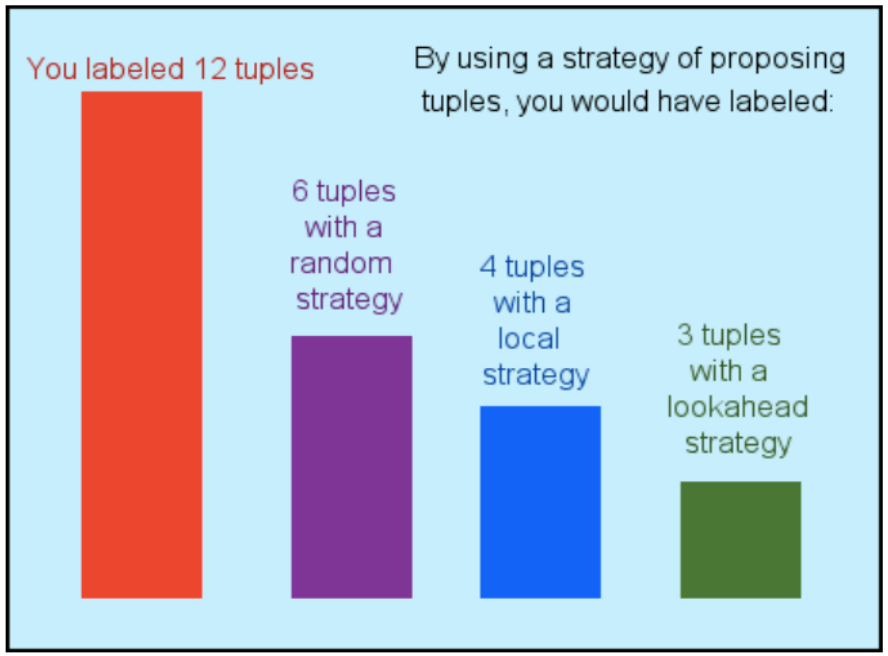
Relational Query Inference

- Problem of interest: query inference via simple tuple labeling (positive or negative)
- Setting: large amount of denormalized data coming from disparate data sources
- Informative tuples that participate to the inference are retained, non-informative ones are pruned



Join Inference Machine (JIM)

- Tuples are labeled as positive or negative by the user [Bonifati et al., ACM TODS16]
- Some strategies are better than others, and the system outputs a comparison among strategies
- The benefit of using a strategy can be presented to the user

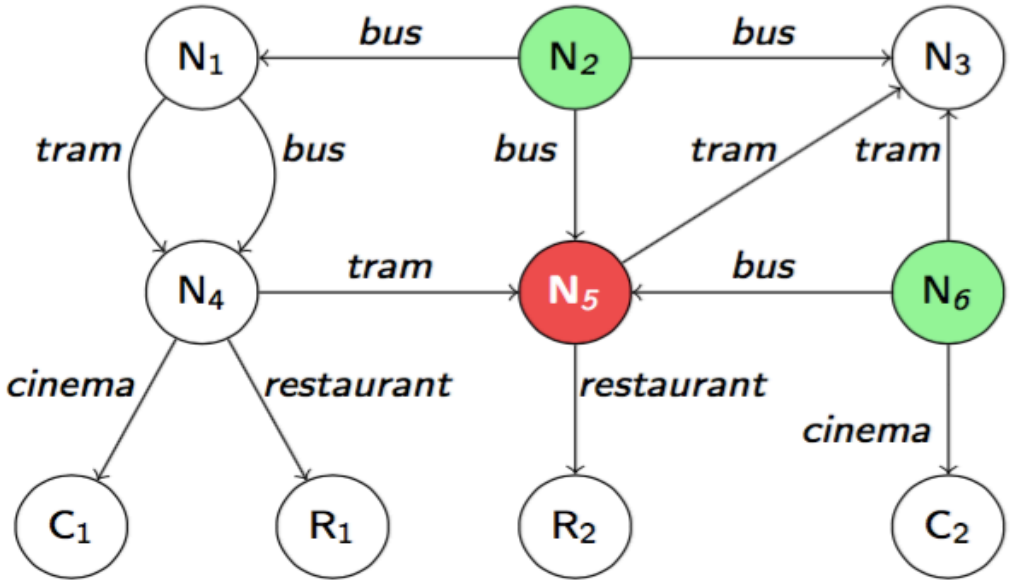


From	*To*	*Airline*	*City*	*Discount*	
Paris	Lille	AF	NYC	AA	(1)
Paris	Lille	AF	Paris	None	(2)
Paris	Lille	AF	Lille	AF	(3)
Lille	NYC	AA	NYC	AA	(4)
Lille	NYC	AA	Paris	None	(5)
Lille	NYC	AA	Lille	AF	(6)
NYC	Paris	AA	NYC	AA	(7)
NYC	Paris	AA	Paris	None	(8)
NYC	Paris	AA	Lille	AF	(9)
Paris	NYC	AF	NYC	AA	(10)
Paris	NYC	AF	Paris	None	(11)
Paris	NYC	AF	Lille	AF	(12)

Learning Path Queries

- Input: Positive and Negative Examples
- Output: The path query that the user has 'in mind'

- Compute consistent queries wrt. the set of input examples
  - (tram+bus)* cinema
  - bus
  - ….
- One can learn in PTIME the query that the user has in mind [Bonifati et al.,EDBT15] by using grammar induction on Regular Path Queries - RPQs



Learning Algorithm[1] for Path Queries

- For each positive node, select its smallest consistent path (SCP). Since the nr. of consistent paths can be infinite, bound by k.
- Generalize SCPs by state merge in the automaton corresponding to the RPQ
- Assuming that k is fixed, the algorithm is polynomial:
  - It returns a consistent query or it abstains from answering.
- Main proved result: For every path query q, there exists a graph and a polynomial set of examples (characteristic sample) that guarantees that the algorithm learns q in polynomial time.

[1] E. M. Gold. Complexity of automaton identification from given data. Information and Control, 1978.

A Condensed View

Tool Name/ Authors	**ML/AI APPROACH**	**GOAL**
JIM/Bonifati et al. 2016	Lattice-based Exploration	Join Query Inference
Bonifati et al. 2015	Grammar Induction Techniques	Path Query Inference

Outline

Part II- ML-Powered Data Integration

- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Recent line of work on DB systems/ML

Disclaimer (borrowed from C. Jermaine's Keynote@EDBT18)

- The ML community has mainly focused on defining models and on application-oriented ML tasks and not on the principles of designing an ML system

- The Database community can provide insights in that direction (given the experience in query optimization, tuning, distributed query evaluation etc.)

Recent line of work on DB systems/ML

We will (non-exhaustively) focus on the following DB contributions:

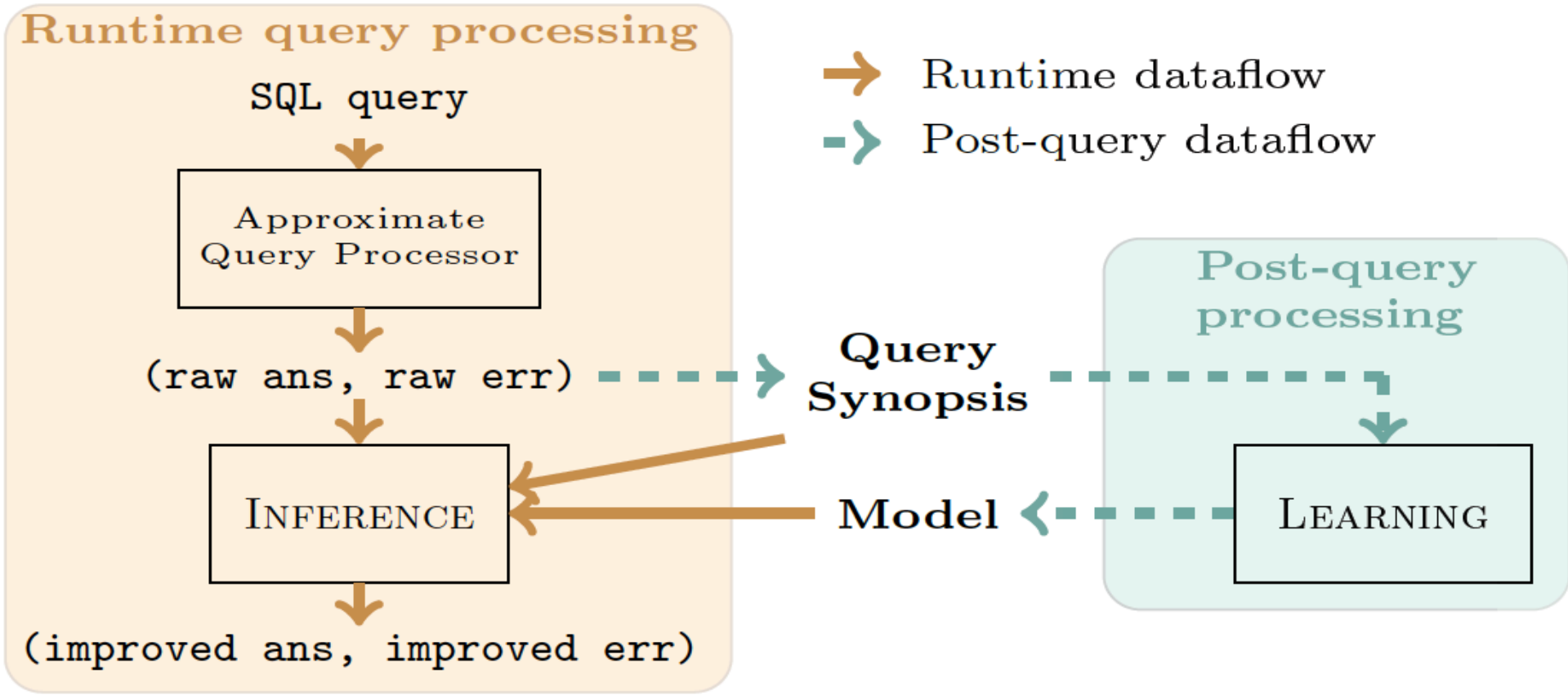
- ML techniques to improve Approximate Query Processing (AQP)
  - relevant for data science/massive data analysis
- ML techniques for DB tuning
  - Interesting problem in the DM stack
- DB techniques to improve feature extraction/labeling training data
  - Relevant for ML

Learning From Past Queries (AQP)

- Intelli[1] is an AQP system that lets improve a raw answer of a classic AQP by using a query synopsis and a model
- When a new query arrives, it goes in the query synopsis as a triple (q, ans, ε)
- The learning module allows to improve the previous triple by leveraging the history in the query synopsis, thus leading to an improved triple (q_i, ans_i, ε_i)
- Where ε_i is shown to be not larger than ε (Theorem proved in the paper)

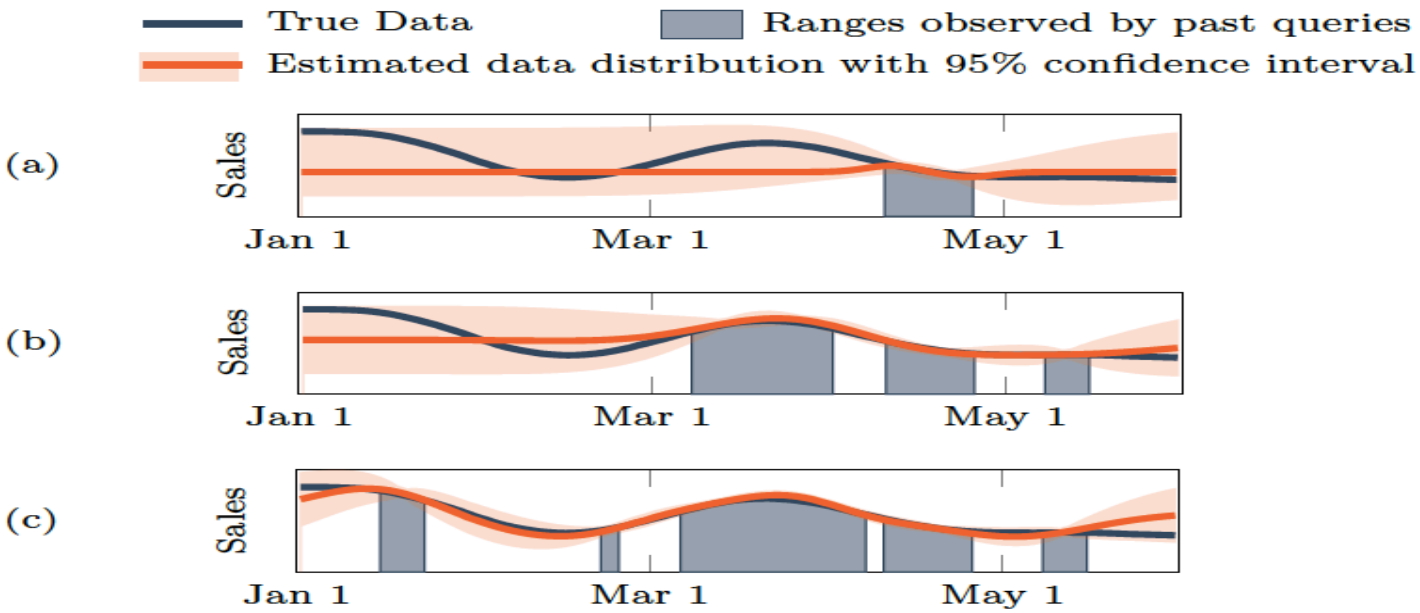
[1] Park et al. "Database Learning: Toward a Database that Becomes Smarter Every Time", SIGMOD17

Intelli: Architecture



Intelli: Underlying Principles

- Queries may still benefit one another even if they access different columns of the data.
- Query answers mutually depend on the underlying distribution of the data
- The more queries are processed, the closer is the estimated data distribution to the true data (a-1 query; b- 2 queries; c- 5 queries etc.)



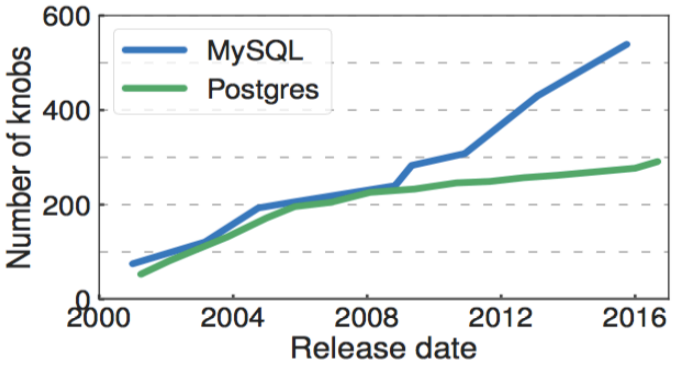
Intelli: Limitations

- Bound by the underlying AQP engine's error estimate
- Can evaluate only AVG, COUNT, SUM (no MIN/MAX, no arbitrary joins)
- The rapidity of the inference depends on the smoothness of the aggregated values' pdf (probability distribution function).
- However, even for non-smooth pdfs, Intelli never worsens the original raw answers (Theorem 1).
- Empirically tested on different data and query distributions

OtterTune: Learning How to Tune a DBMS

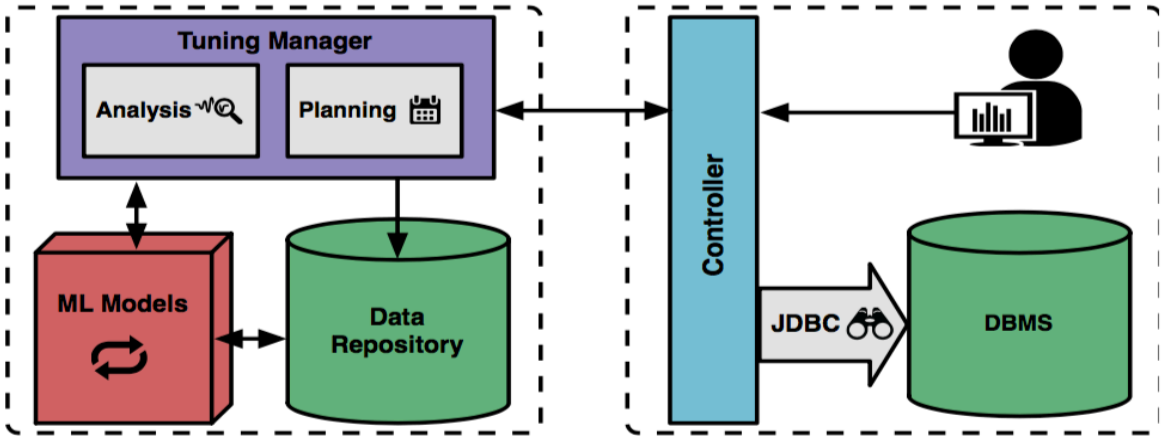
- Manually tuning a DBMS is expensive and time-consuming
  - Several knobs need to be adjusted and they are not standardized, not universal and not independent; moreover, their default configuration is notoriously bad
- OtterTune[2] proposes to leverage supervised and unsupervised learning to automatically tune a DBMS
- It empirically proves that the obtained configurations are as good/better than the ones generated by DBAs

[2] Van Aken et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning" Sigmod 2017



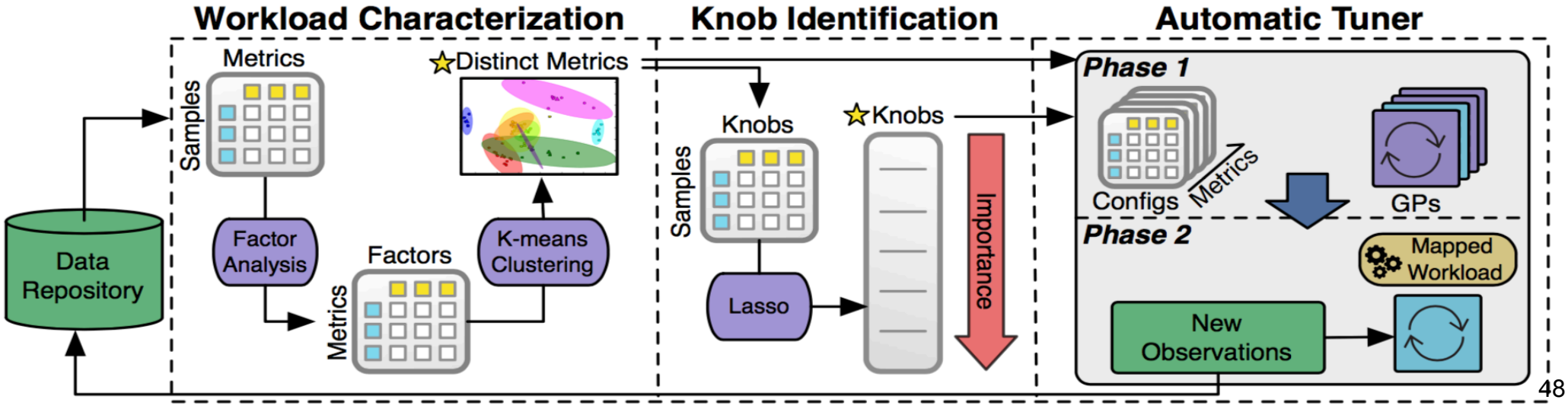
OtterTune Architecture

- The DBA chooses the metric (latency, throughput etc.) he wants to work on and the **controller** connects to the DBMS and gets the knob configuration
- Then, it enters an observation period in which one metric is observed and the DBA can optionally choose to run a set of queries or a workload trace; the result is given to the **tuner manager**
- OtterTune then matches the target workload to a past workload of the same kind
- It then recommends a knob configuration that is optimized to tune a given metric
- It also provides the controller with an estimate of how close the obtained knob configuration is to the best configuration seen so far



OtterTune Automatic Tuning

- Workload Characterization: model discovery starts by collecting DBMS statistics and identifying the smallest set of metrics (with no redundancy)
- Knob Identification: uses a popular feature selection technique called Lasso to expose the most influential knobs (on the system performances)
- Automatic Tuner: (1) Mapping the current workload to a previous one with similar characteristics; (2) recommend configurations by using Gaussian Process (GPs) regression



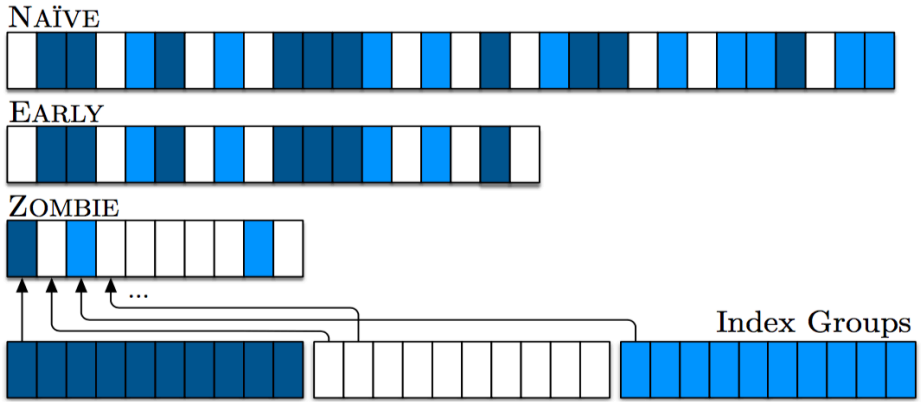
Zombie: Input Selection for Fast Feature Engineering

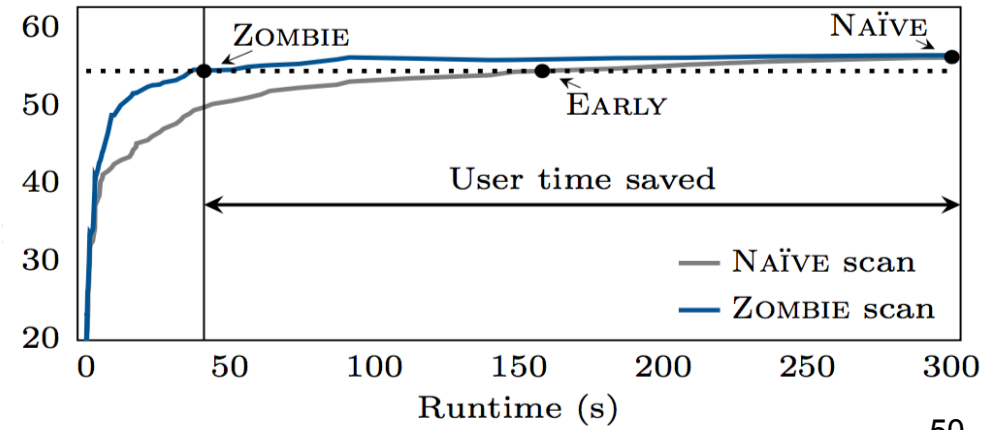
- Feature Engineering and Extraction are the most time-consuming operations in ML
- How can we leverage results in query optimization and database indexing techniques in order to reduce the amount of raw data for feature extraction and minimize the size of the training set used to train a model?
- In Zombie[3], index groups are created out of raw data with k-means clustering; then, it learns (with multi-armed bandit strategy) which groups are more likely to contain the most interesting features.

[3] Anderson et al. "Runtime Support for Human-in-the-Loop Feature Engineering Systems" IEEE Data Engineering Bulletin 2016

Zombie versus Bulk Scan

- Idea: you can stop earlier if you are satisfied with the output of a quality function q thus saving user time
- The dots indicate the 'plateauing of the learning curve', where the processing can be stopped at any time

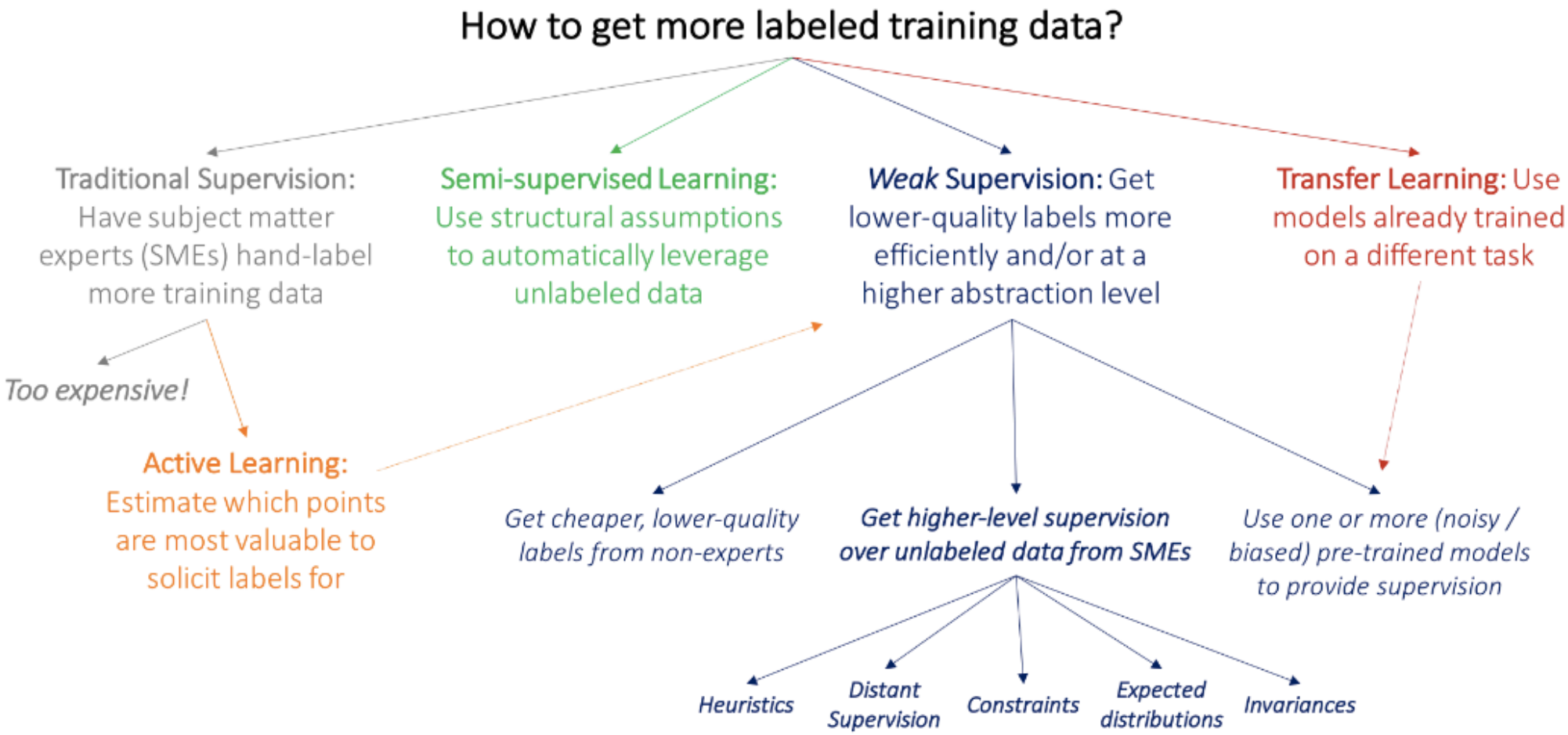




Snorkel: speeding up ML training

- Massive labeling datasets is oftentimes a bottleneck and not always feasible for *any* real-world dataset
- In Snorkel [Ratner et al., PVLDB17], labeling functions are specified via the data programming paradigm: *accuracy* of one function over the other is automatically established and the selected functions are then used to train an end model
- Even *low-accurate* labeling functions defined by users may turn to be apt to obtain *high-quality* models with weak supervision

Classification of the needs of ML areas in terms of labeled training data



https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html

F: Linear Regression over Factorized Databases

- F[3]: A unified framework to express and solve optimization problems for in-database analytics
- Let Q be a feature extraction join query and D a database that defines the training dataset Q(D) for an optimization problem.
- Training dataset computed as join of database tables

$$\begin{pmatrix} y^{(1)} & x_1^{(1)} & \dots & x_n^{(1)} \\ y^{(2)} & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ y^{(m)} & x_1^{(m)} & \dots & x_n^{(m)} \end{pmatrix}$$

$y^{(i)}$ are labels, $x_1^{(i)}, \dots, x_n^{(i)}$ are features, all mapped to reals.

[3] Schleich et al. "Learning Linear Regression Models over Factorized Joins" ACM Sigmod 2016

F: Linear Regression over Factorized Databases

- The goal is to learn the parameters Θ of the following linear function (that approximates the label y of unseen tuples $(x_1,\ldots..,x_n)$)

$$h_\Theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n.$$

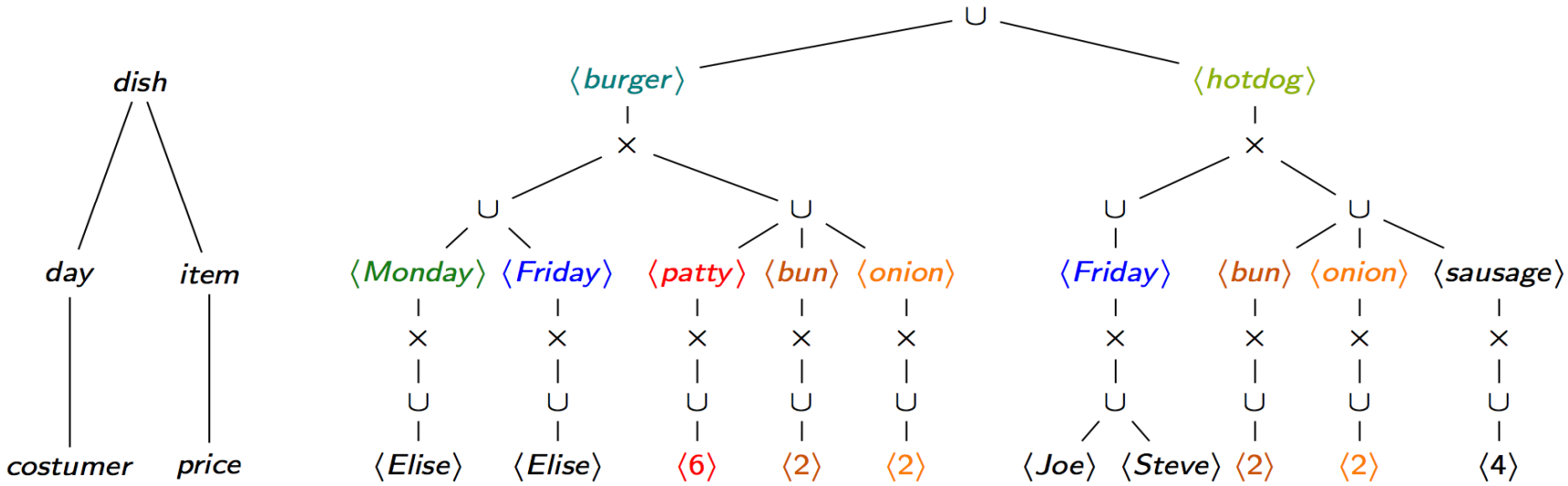
- The least squares regression model with a cost function is considered

$$J(\Theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\Theta(x^{(i)}) - y^{(i)})^2$$

- The Batch Gradient Descent (BGD) Algorithm is applied to learn the Θ

F: Linear Regression over Factorized Databases

- The rough idea is to decouple the computation of Θ from the computation of co-factors, the latter being dependent on input data and executed on the factorized (compressed) version of the database



Join tree

Factorized representation of the join result

A Condensed View*

TOOL NAME	ML APPROACH	GOAL
Intelli	Statistical Inference	Approximate Query Processing(AQP)
Ottertune	GP Regression	DB Tuning
Zombie	Multi-armed bandit strategy	Improve Feature Extraction
Snorkel	a new programming model for weakly-supervised ML	Accelerate ML training
F	Linear Regression	In-database analytics

* not including open-source libraries

Outline

Part II- ML-Powered Data Integration

- ML in Schema-based Transformations
- ML in Schema Constraint Discovery
- ML in Schema Transformation Specification

Part II- ML-Powered Querying and System-oriented Data Management Tasks

- Query Learning
- ML in System-oriented DM Tasks
- Concluding Remarks and Open Issues

Concluding Remarks – Part II

- ML provides a principled framework and efficient tools for inferring database queries and complex transformation abstractions, and for optimizing core system-oriented DM tasks (tuning, join and query evaluation/optimization)
- There are many opportunities for:
  - Studying the **interplay** and the fine-grained combination of **DM/ML** tasks
  - Using DBMS technology to **generalize ML tasks** (the latter being data-dependent as opposed to the former)
  - Thoroughly understanding the **system requirements** of ML tools and their modeling/optimization tasks
  - Orientating our attention to ML techniques that lead to **interpretable/explainable** results

Open Issues (1) – Part II

- Data Transformation and Constraint Discovery:
  - Long-lasting wave of adoption of ML techniques over the last two decades; do they evolve with evolution of ML?
  - Understanding the 'ML community' needs for data/schema transformation and constraint inference
  - Transformation and constraints are 'knowledge' about the data and they declarative; do ML tasks need declarativeness?

- Transformation/Query Specification:
  - Users have a principal role, as in labeling tasks for ML; user supervision in ML can be a useful resource for us
  - Looking at the cases in which no gold standard transformation is given

Open Issues (II) – Part II

- System-oriented DM Tasks:
  - Many tasks benefit from one particular ML techniques; others have not been yet under scrutiny: which ML techniques best suit (or not) a given DM task?
  - Are computational costs, performances important for ML tasks in DM?
  - Are the ML tasks embeddable in a DBMS?
- Other DM tasks (not considered in this tutorial):
  - Distributed/Parallel computation in DM/ML tasks
  - Towards "online ML" in the spirit of "online querying"

ML to Data Management: A Round Trip

Thanks and Questions.

(a pdf of the tutorial will be soon available on our homepages and ICDE18 website)

References* - Part II

[Zeng et Al., PVLDB14] http://www.vldb.org/pvldb/vol8/p197-patel.pdf
[Doan et al. SIGMOD01] https://dl.acm.org/citation.cfm?doid=375663.375731
[Doan et al., WWW02] https://dl.acm.org/citation.cfm?doid=511446.511532
[Li & Clifton, DKE00] https://www.sciencedirect.com/science/article/pii/S0169023X99000440?via%3Dihub
[ten Cate et al., PODS18] (to appear)
[Flach & Savnik, AICommun.99] https://content.iospress.com/articles/ai-communications/aic182
[Rostin et al., WebDB08] http://webdb09.cse.buffalo.edu/papers/Paper30/rostin_et_al_final.pdf
[Jin et al., SIGMOD17] https://dl.acm.org/citation.cfm?doid=3035918.3064034
[Li et al., EMNLP08] http://www.aclweb.org/anthology/D08-1003
[Bonifati et al., SIGMOD17] https://dl.acm.org/citation.cfm?doid=3035918.3064028
[Ratner et al. PVLDB17] http://www.vldb.org/pvldb/vol11/p269-ratner.pdf

* Whenever citations do not appear directly on the corresponding slides

Extra Slides

Not used in the tutorial

OtterTune: Possible Improvements

- Input from the DBA still needed to guide the process
- No means to automatically detect (learn?) the hardware profile
- Not all the costs are taken into account (for instance restarting the DBMS and then identifying knobs that can become bottlenecks in that case)
- An initial assumption is that the DBA has followed the guidelines for a well-specified physical design (indexes, materialized views are already in place…)
- Check the behavior with different regression models

The MADLib Library
http://madlib.apache.org/

- Provides (open source) methods for supervised/ unsupervised learning, descriptive statistics and support models
- The methods are designed for in-and out-of-core execution, and for parallel DBMS as well (uses SQL + Python)
- Designed by GreenPlum/UC Berkeley/Wisconsin/Florida and published in PVLDB17; now part of Apache software suite

Category	Method
Supervised Learning	Linear Regression Logistic Regression Naive Bayes Classification Decision Trees (C4.5) Support Vector Machines
Unsupervised Learning	k-Means Clustering SVD Matrix Factorization Latent Dirichlet Allocation Association Rules
Decriptive Statistics	Count-Min Sketch Flajolet-Martin Sketch Data Profiling Quantiles
Support Modules	Sparse Vectors Array Operations Conjugate Gradient Optimization

Google's TensorFlow
https://github.com/tensorflow/tensorflow

- A distributed ML System
  - providing an API for forward model (represented as a function f(x, θ), where x is problem-specific input and θ is external knowledge)
    - f can be any model (Linear Regression, Neural Networks etc.)
  - an automatic differentiation engine
    - Programmer specifies model and loss in a declarative manner; no need to understand math
  - a compute engine
    - Intrinsic parallel execution and use of the 'compute graph' to be replicated on several compute servers