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
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.

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

Autonomous vehicles

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

eHealth
Automate screening tool for medical imagery diagnostics, bio-augmentation, etc.
Hot Topic for DB community

[VLDB’17 Keynote]

Deep Learning (m)eats Databases
(shortened)

Jens Dittrich

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

Dong Agashe
agashedd.he@gmail.com

Magdalenas Daskalakis
daskal@mcs.st-andrews.ac.uk

Michael Gertz
gertz@cs.berkeley.edu

Theo Kladas
kladas@cs.uoregon.edu

Cheeke Tan
tan@cs.berkeley.edu

4

SIGMOD’17 Workshop

[ SIGMOD’17 Tutorial]

Database Meets Deep Learning: Challenges and Opportunities

Wei Wang, Meihui Zhang, Gaoxi Chen

VLDB’17 Keynote Workshop

workshop@SIGMOD

SIGMOD’15 Panel

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

Arnab Kuncar
arkuncar@uchicago.edu

Matthew Dankers
dankersm@cs.ucsd.edu

Jun Yang
jun.yang@cornell.edu

SIGMOD Record 2016

(SIGMOD’15 Panel)

ACM SIGMOD Blog

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 playing a great role in modern applications. Accompanying the huge wealth of information, the web also introduces new challenges to data processing systems and methods. This year’s workshop 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, QA, 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 Luke Duggan (Amazon Research), Alios Polyzoas (Google), Jens Dittrich (Gwanda University), Avin Balakrishnan (University of California, San Diego), and Peter Fiallo (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 summarise our interviews along the following four questions.

DEEM

2nd Workshop on Data Management for End-to-End Machine Learning

(SIGMOD Blog, Feb. 2018)
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

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

DATA MANAGEMENT TASKS

DATA

- LINKAGE
- REPAIR
- FUSION

SCHEMA & SYSTEM

- SCHEMA MAPPING & TRANSFORMATION
- QUERYING & SYSTEM-ORIENTED TASKS

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)

STRENGTHS

WEAKNESSES

OPPORTUNITIES

THREATS
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.

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

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/](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]
   

   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]
SWOT Analysis (9)

THREATS

4. Adversarial Learning

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

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Will Forth</td>
<td>354-564-339</td>
<td>Ada Bd</td>
</tr>
<tr>
<td>Jacky Khan</td>
<td>435-232-129</td>
<td>Marple Street</td>
</tr>
<tr>
<td>Dom Hack</td>
<td>235-575-689</td>
<td>Main Street</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Database S**

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack Khan</td>
<td>435-223-129</td>
<td>Marple Street</td>
</tr>
<tr>
<td>Hans Ford</td>
<td>354-564-339</td>
<td>Clover Bd</td>
</tr>
<tr>
<td>Tom Hack</td>
<td>235-557-689</td>
<td>Main Street</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Cleaning**

**Standardization**

**Attribute selection**

**Blocking**

**Record pair comparison**

**Decision Model**

**Linkage decision:** 

\[
RL(pair) = \frac{P(\text{vector} \mid \text{pair} \in \text{Match})}{P(\text{vector} \mid \text{pair} \in \text{Non Match})}
\]

**RL(pair)**

- Non Match
- Potential Match
- Match
ML for Entity Resolution (ER)

[Getoor, Machanavajjhala, Tutorial VLDB’12]
[Christen, 2012]
ML-based ER approaches (1)

- **Blocking**
  - **Unsupervised**
    - **Attribute-clustering blocking** [Papadakis et al., TKDE 2013]
  - **Block-clustering** [Fisher et al., KDD’15]
  - **Supervised**
    - **Learning blockers** [Bilenko et al., ICDM’06]
  - **Active**
    - **Crowdsourcing** [Wang et al., VLDB’12][Gokhale et al., SIGMOD’14]

[Note: The diagram includes a reference to a Tutorial ICDE’16, but it is not connected to any content in the text provided.]
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

[Christen, 2008]

Decision trees: interpretable, efficient to apply
Perceptrons: efficient incremental training

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

Training examples

<table>
<thead>
<tr>
<th>f₁</th>
<th>f₂</th>
<th>...</th>
<th>fₙ</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>...</td>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>...</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>...</td>
<td>0.4</td>
<td>1</td>
</tr>
</tbody>
</table>

Unlabeled list

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

<table>
<thead>
<tr>
<th>f₁</th>
<th>f₂</th>
<th>...</th>
<th>fₙ</th>
<th>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>...</td>
<td>0.3</td>
<td>?</td>
</tr>
<tr>
<td>1.0</td>
<td>0.4</td>
<td>...</td>
<td>0.2</td>
<td>?</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2</td>
<td>...</td>
<td>0.5</td>
<td>?</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1</td>
<td>...</td>
<td>0.6</td>
<td>?</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>...</td>
<td>0.4</td>
<td>?</td>
</tr>
<tr>
<td>0.0</td>
<td>0.1</td>
<td>...</td>
<td>0.1</td>
<td>?</td>
</tr>
</tbody>
</table>

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

[Christen, 2008]

[Christen, 2008]

[Christen, 2008]

[Christen, 2008]

[Christen, 2008]

[Christen, 2008]

[Christen, 2008]
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 $\varphi$ for $R1$ and $R2$:

$$\land_{j \in [1, k]} (R1[X1[j]] \approx_j R2[X2[j]]) \rightarrow R1[Z_1] = R2[Z_2],$$
Two assumptions:

• A pre-trained word embeddings for all words in the dataset already exists;

• The pre-trained word embeddings that were trained in a task-agnostic manner are sufficient for the ER task.

Deep learning for ER (1)

Record pair → Relevant word extraction → Word embedding → DNN → Binary classification

FastText

GloVe

Word2Vec

DNN

MLP, LSTM, CNN

Match

UnMatch

Record 1

Renseignements Téléphoniques PagesJaunes

204
rd-pt Pont de Sèvres
92100
BOULOGNE BILLANCOURT

Record 2

PAGESJAUNES

204
DU PONT DE SEVRES
92100

Word embedding

Convolutional hidden layers using several filters + activation layer (ReLU)

Pooling layer (max-pooling, avg-pooling)

Fully connected layer

[Kooli et al., ACIIDS’18]

https://www.pagesjaunes.fr/
Deep learning for ER (2)

Record pair → Relevant word extraction → Word embedding → DNN → Binary classification

GloVE → LSTM-RNN

DeepER [Ebraheem et al., Arxiv 2017]

tuple t → A1 ... Ap ... Am

Embedding lookup layer → Composition (avg, LSTM) layer → Similarity layer → Dense layer → Classification layer

tuple t′ → A1 ... Ap ... Am
Recent Results

• Evaluation of ER with adaptive importance sampling
  [Marchand, Rubinstein, VLDB’17]

• Outside the DB sphere:

  OASIS: a tool for efficient evaluation of classifiers

  Overview

  OASIS is a tool for evaluating binary classifiers when ground truth class labels are not immediately available, but can be obtained at some cost (e.g., by asking human annotators). The tool takes an unlabelled test set as input and intelligently selects items to label so as to provide a precise estimate of the classifier’s performance, whilst minimising the amount of labelling required. The underlying strategy for selecting the items to label is based on a technique called adaptive importance sampling, which is optimised for the classifier performance measure of interest. Currently, OASIS supports estimation of the weighted F-measures, which includes the F1-score, precision and recall.

Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
vince@hit.utdallas.edu
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

HMM for Address Standardization

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
\[ 8^6 = 262,144 \text{ possible combinations of hidden states} \]

- \( \text{Start} \rightarrow \text{Wayfare Name (NU)} \rightarrow \text{Locality Name (LN)} \rightarrow \text{Postal Code (WT)} \rightarrow \text{Territory (LN)} \rightarrow \text{Postal Code (TR)} \rightarrow \text{Territory (PC)} \rightarrow \text{End} \)
  \[ 0.08 \times 0.01 \times 0.02 \times 0.8 \times 0.4 \times 0.01 \times 0.1 \times 0.01 \times 0.8 \times 0.01 \times 0.1 \times 0.01 \times 0.2 = 8.19 \times 10^{-17} \]

- \( \text{Start} \rightarrow \text{Wayfare Number (NU)} \rightarrow \text{Wayfare Name (LN)} \rightarrow \text{Wayfare Type (WT)} \rightarrow \text{Locality Name (LN)} \rightarrow \text{Territory (TR)} \rightarrow \text{Postal Code (PC)} \rightarrow \text{End} \)
  \[ 0.9 \times 0.9 \times 0.95 \times 0.1 \times 0.95 \times 0.92 \times 0.95 \times 0.8 \times 0.4 \times 0.94 \times 0.8 \times 0.85 \times 0.9 = 1.18 \times 10^{-2} \]

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

[Churches et al., 2002]
[Christen et al., 2002]
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 $E_1, E_2, E_3$
Continuous Data Cleaning

**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.

[Volkovs et al., ICDE'14]
On-demand ETL with Lenses

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 );
```

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>brand</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>P123</td>
<td>Apple 6s, White</td>
<td>Var(’X’, R1)</td>
<td>phone</td>
</tr>
<tr>
<td>P124</td>
<td>Apple 5s, Black</td>
<td>Var(’X’, R2)</td>
<td>phone</td>
</tr>
<tr>
<td>P125</td>
<td>Samsung Note2</td>
<td>Samsung</td>
<td>phone</td>
</tr>
<tr>
<td>P2345</td>
<td>Sony 60 inches</td>
<td>Var(’X’, R4)</td>
<td>Var(’Y’, R4)</td>
</tr>
<tr>
<td>P34234</td>
<td>Dell, Intel 4 core</td>
<td>Dell</td>
<td>laptop</td>
</tr>
<tr>
<td>P34235</td>
<td>HP, AMD 2 core</td>
<td>HP</td>
<td>laptop</td>
</tr>
</tbody>
</table>
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.

[Rekatsinas et al., VLDB 2017]

https://github.com/HoloClean/HoloClean

Denial constraints:

\[
\forall t_1, t_2 \in D : \neg (t_1[Zip] = t_2[Zip] \land t_1[City] \neq t_2[City])
\]

\[
\forall t_1, t_2 \in D : \neg (t_1[Zip] = t_2[Zip] \land t_1[State] \neq t_2[State])
\]
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

1. Initialize $W_i^{(1)} = \frac{1}{N}$
2. $\mathcal{L}$ generates a set of classifiers $C\{C^{(0)}, C^{(1)}, \ldots, C^{(k)}\}$ where $C^{(0)}$ is the base classifier and $C^{(1)}, \ldots, C^{(k)}$ are derived from the cleaning operations.
3. for $t \in [1, T]$ do
   4. $C_t = \text{Find } C_t \in \mathcal{C} \text{ that maximizes the weighted accuracy on the test set. } 
   \epsilon_t = \text{Calculate weighted classification error on the test set } \alpha_t = \ln \left( \frac{1}{\epsilon_t} \right) \n   W_i^{(t+1)} \propto W_i^{(t)} e^{-\alpha_t y_i C_t(x_i)}: \text{down-weight correct predictions, up-weight incorrectly predictions.}$
5. return $C(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t C_t(x))$
# A Condensed View

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

- **Physical models**
  - Simulation
  - Estimation
    - Kalman filtering
    - MLE
    - Least Squares

- **Feature-based inference models**

- **Cognitive-based models**
Taxonomy of Data Fusion Techniques
(Not limited to what data fusion means for DB community) [Hall, 1992]

- **Physical models**
- **Feature-based inference models**
- **Cognitive-based models**

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

- **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]

<table>
<thead>
<tr>
<th>Physical models</th>
<th>Feature-based inference models</th>
<th>Cognitive-based models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical templates</td>
<td>Knowledge-based systems</td>
<td>Fuzzy set theory</td>
</tr>
</tbody>
</table>

### Knowledge-based systems

- **Knowledge representation**
  - Scripts, Rules
  - Semantic frames, ontologies
- **Inference methods**
  - Production rules
  - Blackboard
  - Causal/neural nets
- **Search techniques**
- **Uncertainty representation**
  - Dempster-Schafer
  - Probability
  - Confidence factor

DCNN for multi-sensored data fusion
SLiMFast: Probabilistic Models for Data Fusion

[Joglekar et al., SIGMOD’17]
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 $T_o$,
- infer the maximum a posteriori (MAP) assignments to variables $T_o$ 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 (I)

• **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 I (1)

[Anderson et al., 2016]  
[Arasu et al., SIGMOD'10]  
[Assadi, Milo, Novgorodov, WebDB'18]  
[Bahmani et al., SUM'15]  
[Battatcharya, Getoor, TKDD'07]  
[Bellare et al., KDD'12]  
[Bergman et al., SIGMOD 2015]  
[Berti-Equille, Encyclopedia 2018]  
[Biggio et al., ICML'12]  
[Bilenko et al., ICDM'06]  
[Bilenko, Mooney, KDD'03]  
[Chaudhuri et al., ICDE'05]  
[Chaudhuri et al., VLDB’07]  
[Chen et al., SIGMOD’09]  
[Christen et al., 2002]  
[Christen, 2012]  
[Churches et al., 2002]  
[Crotty et al., VLDB’15]  
[Dean, NIPS 2017]  
[Doan et al., HILDA@SIGMOD’17]  
[Dzakovic, XLDB'18]  
[Ebraheem et al., Arxiv 2017]  
[Fellegi, Sunter, 1969]  
[Fisher et al., KDD’15]  
[Getoor, Machanavajjhala, Tutorial VLDB’12]  
[Gokhale et al., SIGMOD’14]  
[Gosh et al., AAAI’17]

https://dl.acm.org/citation.cfm?id=1807252  
https://dl.acm.org/citation.cfm?id=1217304  
http://www.vldb.org/pvldb/vol8/p1900-bergman.pdf  
Encyclopedia of Big Data Technologies, Springer (To Appear), 2018  
https://icml.cc/Conferences/2012/papers/880.pdf  
https://dl.acm.org/citation.cfm?id=956759  
https://dl.acm.org/citation.cfm?id=1559869  
http://www.biomedcentral.com/1472-6947/2/9/  
https://conf.slac.stanford.edu/xldb2018/event-information/lightning-talks  
https://dl.acm.org/citation.cfm?id=2783396  
[Getoor, Machanavajjhala, Tutorial VLDB’12]  
Gokhale et al., SIGMOD’14  
https://dl.acm.org/citation.cfm?id=2588576  
References - Part I (2)

[Gupta, Sarawagi, VLDB’09] [Hall, 1992]
[Hassanzadeh et al., PVLDB’09] [Hu et al., 2017]
[Ilyas, Chu, 2015] [John et al., CIDR’17]
[Joglekar, et al., SIGMOD’17] [Kooli et al., ACIIDS’18]
[Köpcke et al., VLDB’10] [Koudas, Srivastava, Sarawagi, Tutorial
[Kraska et al. 2017] [Krishnan et al., VLDB’16]
[Krishnan et al., 2017] [Kumar et al., SIGMODRec’15]
[Kumar, Boehm, Yang, SIGMOD’17 Tutorial]
[Luyang et al., Sensors’17] [Marchand, Rubinstein, VLDB’17]
[Marchand, Rubinstein, VLDB’17] [Marcus, Arxiv, 2018]
[Mintz et al., 2009] [Natarajan et al., NIPS’13]
[Papadakis et al., TKDE 2013] [Polyzotis et al., SIGMOD’17]
[Papadakis, Palpanas, Tutorial ICDE’16] [Qian et al., CIKM’17]

https://dl.acm.org/citation.cfm?id=1687627.1687661
Mathematical Techniques in Multisensor Data Fusion, ArtechHouse, 1992
http://www.vldb.org/pvldb/2/vldb09-1025.pdf
http://pages.cs.wisc.edu/~jignesh/publ/Ava.pdf
https://dl.acm.org/citation.cfm?id=3035951
https://link.springer.com/chapter/10.1007%2F978-3-319-75420-8_1
SIGMOD’06]
http://www.cs.toronto.edu/~koudas/docs/aj.pdf
https://arxiv.org/abs/1712.01208
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5335931/
https://dl.acm.org/citation.cfm?id=1690287
https://dl.acm.org/citation.cfm?id=3035918.3054782
https://dl.acm.org/citation.cfm?id=3132949
References - Part I (3)

[Sarawagi, Bhamidipaty, KDD’02] https://dl.acm.org/citation.cfm?id=775087
[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
[Wang et al., SIGMOD’13’] https://dl.acm.org/citation.cfm?id=2465280
[Yakout, Berti-Equille, Elmagarmid SIGMOD’13] https://dl.acm.org/citation.cfm?id=2463706
ML to Data Management:
A Round Trip

PART II
Angela Bonifati
Our Focus: ML applications to DM

DATA MANAGEMENT TASKS

DATA
- LINKAGE
- REPAIR
- FUSION

SCHEMA & SYSTEM
- SCHEMA MAPPING & TRANSFORMATION
- QUERYING & SYSTEM-ORIENTED TASKS

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

  Seattle, WA
  Kent, WA
  Bend, OR

  Name Learner — (address, 0.4)
  Naive Bayes — (address, 0.9)
  Meta-Learner — (address, 0.4*0.2 + 0.9*0.8 = 0.8)
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:
  - $P(A,B)$, $P(A,\neg B)$, $P(\neg A,B)$, $P(\neg A,\neg B)$
- ML is used to infer whether $P(A,B)$ can be approximated with $P(A \text{ intersect } B)$
  - 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

\[ W.-S. \text{ Li, C. Clifton / Data \\ \\ & Knowledge Engineering 33 (2000) 49–84}\]
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).

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
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.

\[
\arg\min_{\mathcal{M} \subseteq C} \left( \sum_{t \in J} [1 - \text{explains}_{\text{full}}(\mathcal{M}, t)] + \sum_{t \in K_C - J} [\text{error}_{\text{full}}(\mathcal{M}, t)] + \text{size}_m(\mathcal{M}) \right)
\]
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

<table>
<thead>
<tr>
<th>TOOL NAME</th>
<th>ML APPROACH</th>
<th>GOAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSD</td>
<td>Multi-strategy Learning</td>
<td>Schema Matching</td>
</tr>
<tr>
<td>Glue</td>
<td>Multi-strategy Learning</td>
<td>Ontology Matching</td>
</tr>
<tr>
<td>Automatch</td>
<td>Bayesian networks</td>
<td>Schema Matching</td>
</tr>
<tr>
<td>SemInt</td>
<td>Neural networks</td>
<td>Schema Matching</td>
</tr>
<tr>
<td>CMD</td>
<td>Statistical Relational Learning</td>
<td>Schema Mapping</td>
</tr>
<tr>
<td>GAV Learn</td>
<td>Active Learning</td>
<td>Schema Mapping</td>
</tr>
</tbody>
</table>
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)

• 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.

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

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 $R_{new}$ 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

<table>
<thead>
<tr>
<th>Authors</th>
<th>ML/AI APPROACH</th>
<th>GOAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flach et al. 99</td>
<td>Inductive Logic Programming</td>
<td>FD/IND Discovery</td>
</tr>
<tr>
<td>Rostin et al. 09</td>
<td>Naive Bayes, SVM, J48 and DT</td>
<td>FD Discovery</td>
</tr>
<tr>
<td>Li et al. 08</td>
<td>Hill-Climbing Algorithm</td>
<td>Regex Expressions Discovery</td>
</tr>
</tbody>
</table>
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. SIGMOD'17]
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

\[ f(n) = g(n) + h(n) \]
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.

![Diagram showing mapping design process](image)

**Source**

<table>
<thead>
<tr>
<th>Company</th>
<th>&lt;/br&gt;IdCompany</th>
<th>Name</th>
<th>Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘C1’</td>
<td>‘AA’</td>
<td>‘Paris’</td>
<td></td>
</tr>
<tr>
<td>‘C2’</td>
<td>‘Ev’</td>
<td>‘Lyon’</td>
<td></td>
</tr>
</tbody>
</table>

**Flight**

<table>
<thead>
<tr>
<th>Departure</th>
<th>Arrival</th>
<th>IdCompany</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Lyon’</td>
<td>‘Paris’</td>
<td>‘C1’</td>
</tr>
<tr>
<td>‘Paris’</td>
<td>‘Lyon’</td>
<td>‘C2’</td>
</tr>
</tbody>
</table>

**Travel Agency**

<table>
<thead>
<tr>
<th>IdAgency</th>
<th>Name</th>
<th>Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘A1’</td>
<td>‘TC’</td>
<td>‘L.A.’</td>
</tr>
</tbody>
</table>

**Carrier**

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘id1’</td>
<td>‘AA’</td>
<td>‘Paris’</td>
</tr>
<tr>
<td>‘id2’</td>
<td>‘Ev’</td>
<td>‘Lyon’</td>
</tr>
<tr>
<td>‘id3’</td>
<td>‘TC’</td>
<td>‘L.A.’</td>
</tr>
</tbody>
</table>

**Departure**

<table>
<thead>
<tr>
<th>Town</th>
<th>IdCarrier</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Lyon’</td>
<td>‘id1’</td>
</tr>
<tr>
<td>‘Paris’</td>
<td>‘id1’</td>
</tr>
</tbody>
</table>

**Arrival**

<table>
<thead>
<tr>
<th>Town</th>
<th>IdCarrier</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘Lyon’</td>
<td>‘id2’</td>
</tr>
<tr>
<td>‘L.A.’</td>
<td>‘id2’</td>
</tr>
</tbody>
</table>

**Final mapping**

\[
m_1 : \text{Company}(c1, aa, paris_1) \land \text{Flight}(lyon, paris_2, c1) \\
\quad \rightarrow \exists id1, \text{Firm}(id1, aa, paris_1) \land \text{Departure}(lyon, id1) \land \text{Arrival}(paris_2, id1)
\]

\[
m_2 : \text{TravelAgency}(a1, tc, la) \\
\quad \rightarrow \exists id3, \text{Firm}(id3, tc, la)
\]
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:**

\[ \Sigma = \{ \text{Company}(c1, aa, paris) \land \text{Flight}(lyon, paris, c1) \} \]

\[ \Rightarrow \exists \text{id1}, \text{Carrier}(\text{id1}, aa, paris) \land \text{Departure}(lyon, \text{id1}) \land \text{Arrival}(paris, \text{id1}) \]
Effectiveness of the Interactive Method

• All exploration strategies keep the number of questions (per tgd) low along atom refinement.
## A Condensed View

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

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Airline</th>
<th>City</th>
<th>Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris</td>
<td>Lille</td>
<td>AF</td>
<td>NYC</td>
<td>AA</td>
</tr>
<tr>
<td>Paris</td>
<td>Lille</td>
<td>AF</td>
<td>Paris</td>
<td>None</td>
</tr>
<tr>
<td>Paris</td>
<td>Lille</td>
<td>AF</td>
<td>Lille</td>
<td>AF</td>
</tr>
<tr>
<td>Lille</td>
<td>NYC</td>
<td>AA</td>
<td>NYC</td>
<td>AA</td>
</tr>
<tr>
<td>Lille</td>
<td>NYC</td>
<td>AA</td>
<td>Paris</td>
<td>None</td>
</tr>
<tr>
<td>Lille</td>
<td>NYC</td>
<td>AA</td>
<td>Lille</td>
<td>AF</td>
</tr>
<tr>
<td>NYC</td>
<td>Paris</td>
<td>AA</td>
<td>NYC</td>
<td>AA</td>
</tr>
<tr>
<td>NYC</td>
<td>Paris</td>
<td>AA</td>
<td>Paris</td>
<td>None</td>
</tr>
<tr>
<td>Paris</td>
<td>NYC</td>
<td>AF</td>
<td>NYC</td>
<td>AA</td>
</tr>
<tr>
<td>Paris</td>
<td>NYC</td>
<td>AF</td>
<td>Paris</td>
<td>None</td>
</tr>
<tr>
<td>Paris</td>
<td>NYC</td>
<td>AF</td>
<td>Lille</td>
<td>AF</td>
</tr>
</tbody>
</table>

You labeled 12 tuples
- 6 tuples with a random strategy
- 4 tuples with a local strategy
- 3 tuples with a lookahead strategy

By using a strategy of proposing tuples, you would have labeled:
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\textsuperscript{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.

\textsuperscript{1} E. M. Gold. Complexity of automaton identification from given data. Information and Control, 1978.
## A Condensed View

<table>
<thead>
<tr>
<th>Tool Name/Authors</th>
<th>ML/AI APPROACH</th>
<th>GOAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIM/Bonifati et al. 2016</td>
<td>Lattice-based Exploration</td>
<td>Join Query Inference</td>
</tr>
<tr>
<td>Bonifati et al. 2015</td>
<td>Grammar Induction Techniques</td>
<td>Path Query Inference</td>
</tr>
</tbody>
</table>
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 \((q, \text{ans, } \varepsilon)\)

• The learning module allows to improve the previous triple by leveraging the history in the query synopsis, thus leading to an improved triple \((q_i, \text{ans}_i, \varepsilon_i)\)

• Where \(\varepsilon_i\) is shown to be not larger than \(\varepsilon\) (Theorem proved in the paper)

\(^1\) Park et al. “Database Learning: Toward a Database that Becomes Smarter Every Time”, SIGMOD 17
**Intelli: Architecture**

**Runtime query processing**
- SQL query
- Approximate Query Processor
- (raw ans, raw err)
- Inference
- (improved ans, improved err)

**Post-query processing**
- Query Synopsis
- Model
- Learning

**Dataflow**
- Runtime dataflow
- Post-query dataflow
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.

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

How to get more labeled training data?

- **Traditional Supervision:** Have subject matter experts (SMEs) hand-label more training data
  - *Too expensive!*

- **Semi-supervised Learning:** Use structural assumptions to automatically leverage unlabeled data

- **Weak Supervision:** Get lower-quality labels more efficiently and/or at a higher abstraction level

- **Transfer Learning:** Use models already trained on a different task

- **Active Learning:** Estimate which points are most valuable to solicit labels for

  - **Get cheaper, lower-quality labels from non-experts**
  - **Get higher-level supervision over unlabeled data from SMEs**
  - Use one or more (noisy / biased) pre-trained models to provide supervision

Heuristics, Distant Supervision, Constraints, Expected distributions, Invariances

https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html
F: Linear Regression over Factorized Databases

- **F³**: 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

\[
\begin{pmatrix}
  y^{(1)} & x_1^{(1)} & \ldots & x_n^{(1)} \\
  y^{(2)} & x_1^{(2)} & \ldots & x_n^{(2)} \\
  \vdots & \vdots & \ddots & \vdots \\
  y^{(m)} & x_1^{(m)} & \ldots & x_n^{(m)}
\end{pmatrix}
\]

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

F: Linear Regression over Factorized Databases

• The goal is to learn the parameters $\Theta$ 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 $\Theta$
The rough idea is to decouple the computation of $\Theta$ from the computation of co-factors, the latter being dependent on input data and executed on the factorized (compressed) version of the database.
### A Condensed View*  

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

* 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 (I) – 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

[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
[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

* 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

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised Learning</td>
<td>Linear Regression</td>
</tr>
<tr>
<td></td>
<td>Logistic Regression</td>
</tr>
<tr>
<td></td>
<td>Naive Bayes Classification</td>
</tr>
<tr>
<td></td>
<td>Decision Trees (C4.5)</td>
</tr>
<tr>
<td></td>
<td>Support Vector Machines</td>
</tr>
<tr>
<td>Unsupervised Learning</td>
<td>k-Means Clustering</td>
</tr>
<tr>
<td></td>
<td>SVD Matrix Factorization</td>
</tr>
<tr>
<td></td>
<td>Latent Dirichlet Allocation</td>
</tr>
<tr>
<td></td>
<td>Association Rules</td>
</tr>
<tr>
<td>Deccriptive Statistics</td>
<td>Count-Min Sketch</td>
</tr>
<tr>
<td></td>
<td>Flajolet-Martin Sketch</td>
</tr>
<tr>
<td></td>
<td>Data Profiling</td>
</tr>
<tr>
<td></td>
<td>Quantiles</td>
</tr>
<tr>
<td>Support Modules</td>
<td>Sparse Vectors</td>
</tr>
<tr>
<td></td>
<td>Array Operations</td>
</tr>
<tr>
<td></td>
<td>Conjugate Gradient Optimization</td>
</tr>
</tbody>
</table>
Google’s TensorFlow
https://github.com/tensorflow/tensorflow

• A distributed ML System
  – providing an API for forward model (represented as a function $f(x, \theta)$, where $x$ is problem-specific input and $\theta$ 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