CS 8803-MDS

Human-in-the-loop Data Analytics

Lecture 13

10/05/22

Logistics

Progress Report (1%)

due Fridays 5PM at 10/21, 10/28, 11/4, 11/11, 11/18

option to submit 4/5 and have one report double count

Next week

how to make progress in research

Today's class

[Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks]

Author: Bojun, Siddhi

Reviewer: Shubham, Shen En

Archaeologist: Aniruddha

Practioner: Jingfan

Researcher: Ting

PORTLAND ACM SIGMOD + PODS 2020

ACM SIGMOD/PODS International Conference on Management of Data
June 14-19th, 2020
Portland, Oregon USA

SIGMOD PODS

ACM SIGMOD/PODS International Conference on Management of Data June 14 - June 19, 2020 Portland, OR, USA

Welcome

- Homepage
- News

Organization

- Experience Report
- Organization
- SIGMOD PC
- PODS PC

2020 ACM SIGMOD/PODS @ Portland, OR, USA

SIGMOD/PODS 2020 Experience Report now available

Coronavirus Updates

Updates on SIGMOD/PODS 2020 Registration (12 May 2020)

Calls For Submissions

- Important Dates
- Calls for Submissions

PODS Program

- Program Overview
- Detailed Program
- Research Papers
- Keynote Talk

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Bojun Yang & Siddhi Pandare

Motivation

- Data-preparation steps like Pivot and Join need skilled users
- Automating data preparation steps can improve efficiency of the user (technical and non-technical experts)
- Data preparation recommendation systems automate commonly used operators

Overview

Pandas library + jupyter notebooks is commonly used for data preparation

```
# First, add the platform and device to the user usage.
result = pd.merge(result,
                  devices,
                  left_on='device',
                  right_on='Model',
                  how='left')

result.head()
```

Fig. Merge (Join) in Pandas

Overview

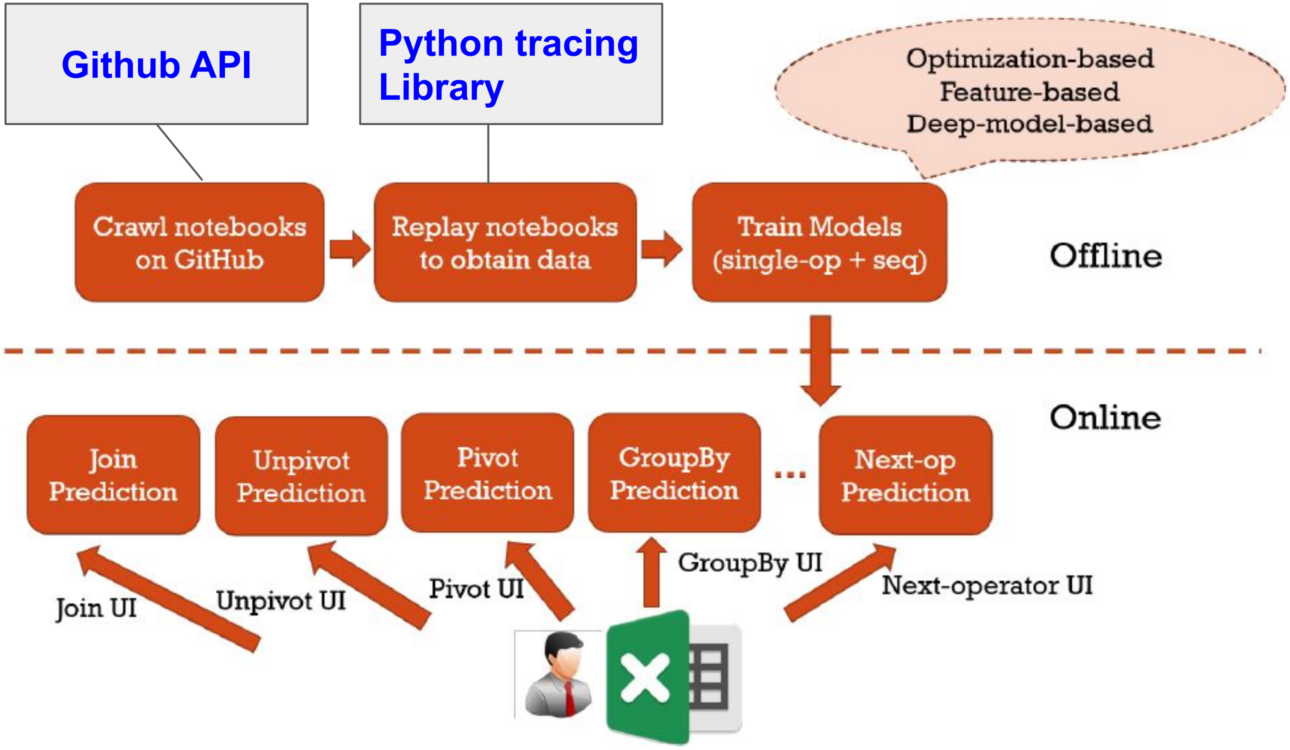


Fig. System Architecture

Join/ Merge

Problem: Given Tables T and T' find columns (S, S') that are likely to join.

author	bestsellers_date	title	description	publisher	rank	rank_on_list
Dean R Koontz	2008-05-24	ODD HOURS	Odd Thomas, who can communicate..	Bantam	0	1
Stephenie Meyer	2008-05-24	THE HOST	Aliens have taken control...	Little, Brown	1	2
Emily Giffin	2008-05-24	LOVE THE ONE	A woman's happy marriage is...	St. Martin's	2	3
Patricia Cornwell	2008-05-24	THE FRONT	Massachusetts state investigator...	Putnam	0	4

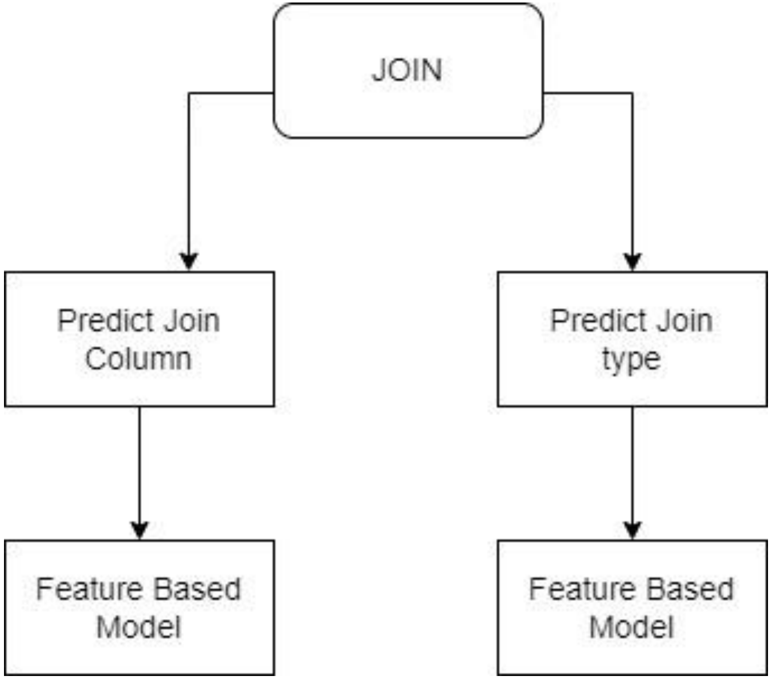
$\bowtie$

title_on_list	weeks_on_list
10TH ANNIVERSARY	8
11/22/63	9
11TH HOUR	8
1225 CHRISTMAS TREE	4
12TH OF NEVER	6

Fig. Example of join

Proposed Solution

- Two steps:
  - Predict Join column
  - Predict Join type



Join: Features

- Distinct-value-ratio
  - Ratio of distinct tuples in S and S’ over total number of rows. At least one of them should be have this ratio close to 1 (key column)
- Value overlap
  - Pairs of high value overlap are likely to be join columns.
- Value range overlap
  - Calculate the min/max range of S and S’ then calculate the overlap of the ranges.
- Col-value-types
  - Two string columns with high overlap are likely to be join columns than two integer columns with high overlap.
- Leftness, sortedness, single-column-candidate, Table statistics

GroupBy/Aggregate - Problem/Example

given table T and columns $\{C_i\} \epsilon T$

Candidate GroupBy Cols | Candidate Agg Cols

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

GroupBy: [Company, Year]
Aggregate: [Revenue]

Company	Year	Revenue
AEROJET ROCKETD	2006	6218.09
AEROJET ROCKETD	2007	6342.45
AEROJET ROCKETD	2008	7088.62
...	...	...
YORK WATER CO	2007	1940.42
YORK WATER CO	2008	2168.71

GroupBy/Aggregate - Features 1

- Distinct-Value-Count: # of distinct values in C
    - GroupBy columns usually have a small cardinality
- Column-Data-Type: string, int, float, etc of data type
    - GroupBy columns more likely to use string data type
- Left-ness: how to the left of the table C is
    - GroupBy columns more likely to be near the left of the table
    - Agg columns more likely to be near the right of the table

Candidate GroupBy Cols | Candidate Agg Cols

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

GroupBy/Aggregate - Features 2

- Emptiness: nulls in C
  - GroupBy columns tend have low emptiness
- Value-Range: min-max range of C if it is numeric
  - GroupBy columns tend to have small ranges
- Peak-Frequency: frequency of most common value in C
- Column-Names: lookup in training data to see how often it is used by each op

Candidate GroupBy Cols

Candidate Agg Cols

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

GroupBy/Aggregate - Problem/Example

given table T and columns $\{C_i\} \epsilon T$

Candidate GroupBy Cols					Candidate Agg Cols	
Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

GroupBy: [Company, Year]
Aggregate: [Revenue]

Company	Year	Revenue
AEROJET ROCKETD	2006	6218.09
AEROJET ROCKETD	2007	6342.45
AEROJET ROCKETD	2008	7088.62
...	...	...
YORK WATER CO	2007	1940.42
YORK WATER CO	2008	2168.71

Pivot - What does it do

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

Index: [Sector, Ticker, Company]
Column: [Year]
Agg: Sum
Agg Column: Revenue

Sector	**Ticker**	**Company**	**2006**	**2007**	**2008**
Aerospace	AJRD	AEROJET ROCKETD	6218.09	6342.45	7088.62
	ATRO	ASTRONICS CORP	1050.97	1071.99	1198.11
Business Services	HHS	HARTE-HANKS INC	2473.75	2523.22	2820.07
	NCMI	NATL CINEMEDIA	856.92	874.06	976.89
Consumer Staples	YTEN	TIELD10 BIOSCI	533.13	543.79	607.77
	...	...	...	...	...
Utilities	YORW	YORK WATER CO	1902.37	1940.42	2168.70

Pivot - Prediction Overview

1. Predict index/header vs. aggregation columns
   a. Predicting index/header columns = predicting GroupBy columns
   b. Predicting agg columns = predicting agg columns
2. Predict to split index vs header (after user selects dimension columns)
   a. Hard for users and typically requires many trial and errors
   b. Predict affinity scores for pairwise columns
   c. Formulate the problem as an optimization problem using affinity scores and solve

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

Pivot - Index/Header vs. Aggregation Columns

- Directly apply GroupBy/Aggregation prediction
- We choose Sector Ticker, Company, Year for index/header columns
  - All GroupBy columns are reasonable choices for pivot index/header columns
- We choose Revenue as the aggregation column
  - All Aggregation columns are reasonable choices for pivot aggregation columns

Candidate GroupBy Cols					Candidate Agg Cols	
Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

Pivot - Predict Split Index vs. Header

Ticker	Company	Year	Aerospace	Business Services	...	Utilities
AJRD	AEROJET ROCKETD	2006	6218.09	NULL	...	NULL
AJRD	AEROJET ROCKETD	2007	6342.45	NULL	...	NULL
AJRD	AEROJET ROCKETD	2008	7088.62	NULL	...	NULL
ATRO	ASTRONICS CORP	2006	1050.97	NULL	...	NULL
...	...	...	...	...	...	...
HHS	HARTE-HANKS INC	2006	NULL	2473.75	...	NULL
...	...	...	...	...	...	...
YORW	YORK WATER CO	2008	NULL	NULL	...	2168.7

- Likelihood of 2 columns being on the same side of pivot (both index or both header)
  - Regression model to learn the affinity score between any 2 pair of columns

Pivot - Affinity Score Feature 1

Sector	Ticker	Company	Year	Quarter	Market Cap	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	Q1	1442.67	472.07
Aerospace	AJRD	AEROJET ROCKETD	2006	Q2	1514.80	489.22
...	...	...	...	...	...	...
Aerospace	BA	BOEING CO	2006	Q1	343.41	210.66
...	...	...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	Q4	600.19	271.73

20 sectors, 1000 companies, 3 years

- Emptiness-Reduction-Ratio

$$\frac{|\{u|u \in T(C_i)\}||\{v|v \in T(C_j)\}|}{|\{(u,v)|(u,v) \in T(C_i, C_j)\}|}$$

  - How much emptiness we can save by arranging C_i and C_j on the same side.
  - T(C) is unique values in column C.
  - Sector and Company: 20 * 1000 / 1000 = 20
  - Sector and Year: 3 * 20 / 60 = 1

Pivot - Affinity Score Feature 2

- Column-Position-Difference
  - Relative difference of position between C_i and C_j in T
  - Columns that are close to each other in T are more likely to be related and on same side of pivot

Regression Model Training w/ Real Pivot Tables

- Pairs of columns on same side (+1)
- Pairs of columns on different side (-1)
- Predict pairwise column affinity

Pivot - AMPT Optimization Problem

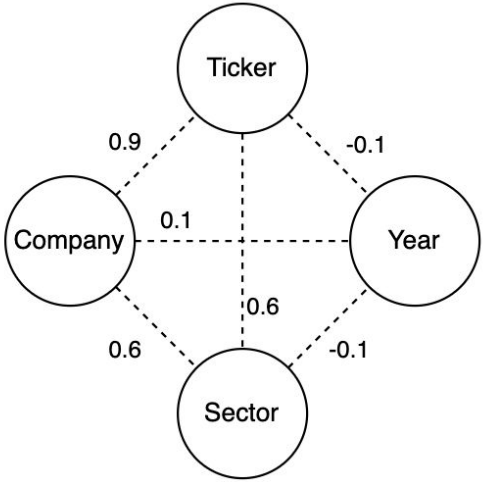
- Model each column as a vertex in the graph
- Use regression model to produce affinity scores on all edges
- Affinity-Maximizing Pivot-Table:

$$(\text{AMPT}) \quad \max \sum_{c_i, c_j \in C} a(c_i, c_j) + \sum_{c_i, c_j \in \overline{C}} a(c_i, c_j) - \sum_{c_i \in C, c_j \in \overline{C}} a(c_i, c_j) \qquad (1)$$

$$\text{s.t. } C \cup \overline{C} = C \qquad (2)$$ Fully covers C

$$C \cap \overline{C} = \emptyset \qquad (3)$$ Disjoint

$$C \neq \emptyset, \overline{C} \neq \emptyset \qquad (4)$$ Non-empty

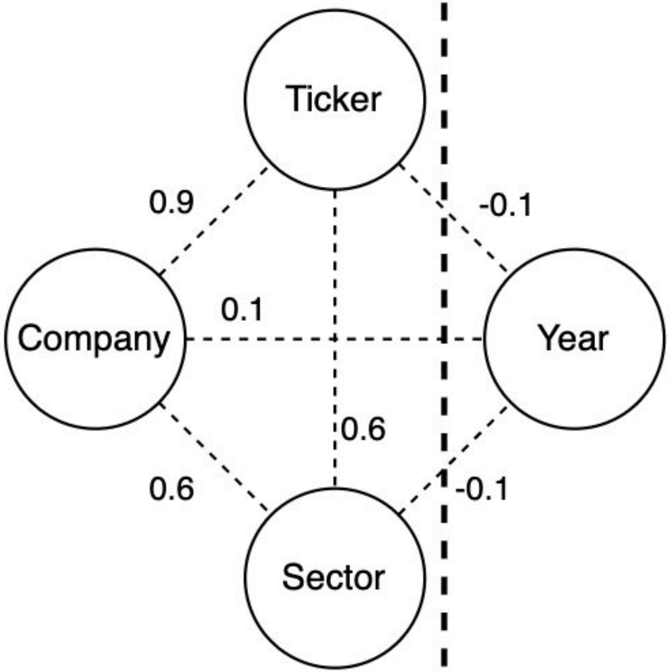


- AMPT reduces to [two-way graph cut](), solvable in polytime with [Stoer-Wagner Algorithm]()

Pivot - AMPT Example

- Intra pairwise C: 0.9 + 0.6 + 0.6 = 2.1
- Intra pairwise C': 0
- Inter pairwise: -0.1 - 0.1 + 0.1 = -0.1
- 2.1 + 0 - (-0.1) = 2.2

- Affinity scoring model + AMPT forumation allows us to find most likely pivot



Unpivot/ Melt

Problem: Predict set of columns to collapse in Unpivot

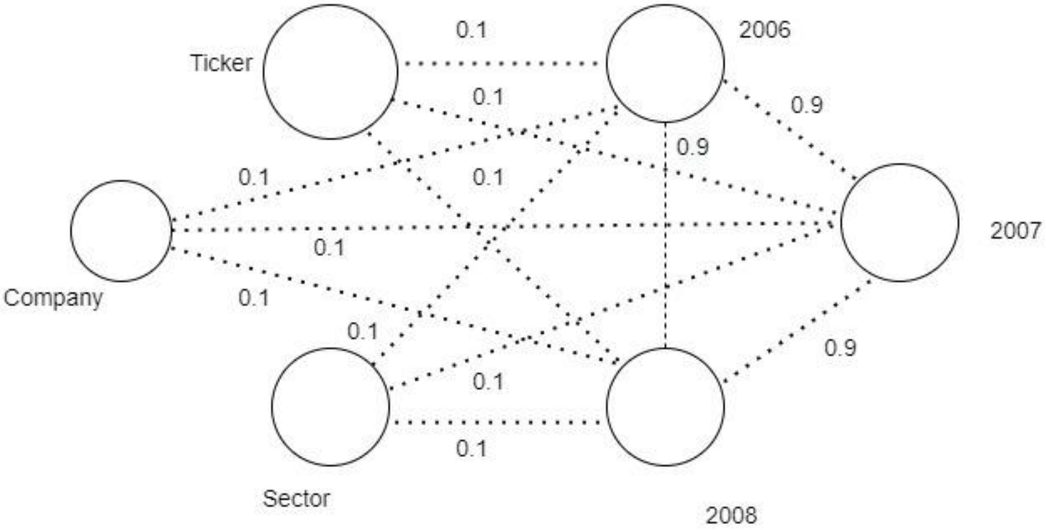
Sector	Ticker	Company	2006	2007	2008
Aerospace	AJRD	AEROJET ROCKETD	6218.09	6342.45	7088.62
	ATRO	ASTRONICS CORP	1050.97	1071.99	1198.11
Business Services	HHS	HARTE-HANKS INC	2473.75	2523.22	2820.07
	NCMI	NATL CINEMEDIA	856.92	874.06	976.89
Consumer Staples	YTEN	TIELD10 BIOSCI	533.13	543.79	607.77
	...	...	...	...	...
Utilities	YORW	YORK WATER CO	1902.37	1940.42	2168.70

Sector	Ticker	Company	Year	Revenue
Aerospace	AJRD	AEROJET ROCKETD	2006	6218.09
Aerospace	AJRD	AEROJET ROCKETD	2007	6342.45
Aerospace	AJRD	AEROJET ROCKETD	2008	7088.62
Aerospace	ATRO	ASTRONICS CORP	2006	1050.97
Aerospace	ATRO	ASTRONICS CORP	2007	1071.99
...	...	...	...	...
Utilities	YORW	YORK WATER CO	2008	2168.70

Unpivot on columns 2006, 2007, 2008

Compatibility score

- Compatibility score measures the likelihood of the columns being on the same side of the unpivot
- Like affinity score we train a regression model to find the compatibility-scores



Optimization: Compatibility-maximizing- Unpivot Table (CMUT)

Compatibility-maximizing- Unpivot Table (CMUT) :

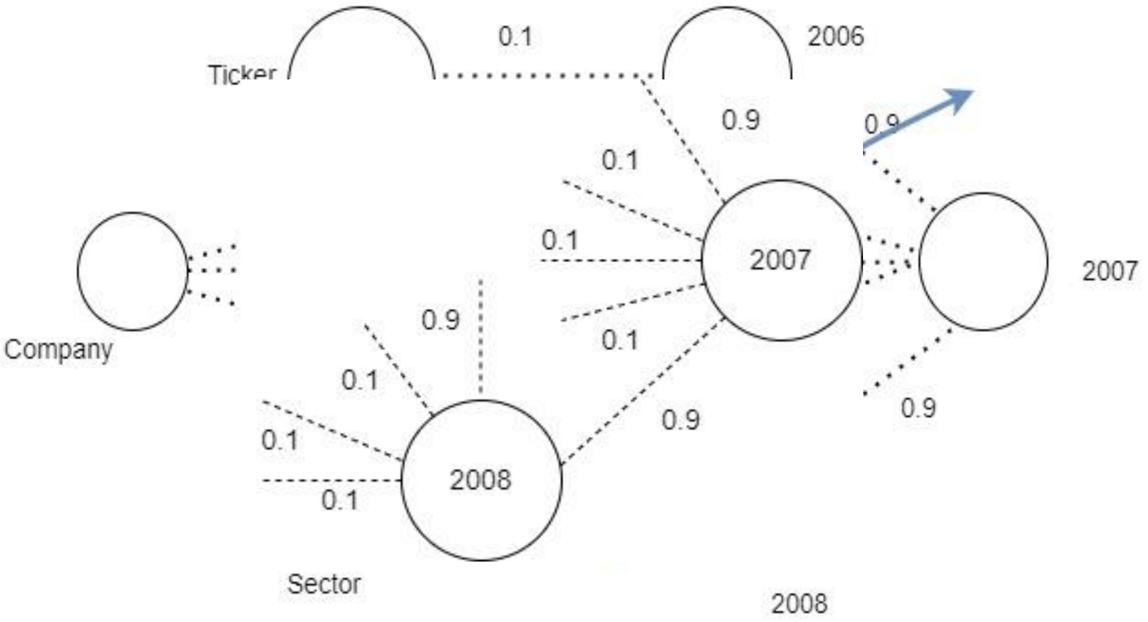
$$\text{(CMUT)} \ \max \ \underset{c_i, c_j \in C}{\text{avg}} a(c_i, c_j) - \underset{c_i \in C, c_j \in C \setminus C}{\text{avg}} a(c_i, c_j)$$

$$\text{s.t. } C \subset C$$

$$|C| \geq 2$$

Solution: Greedy Algorithm

Example : Unpivot



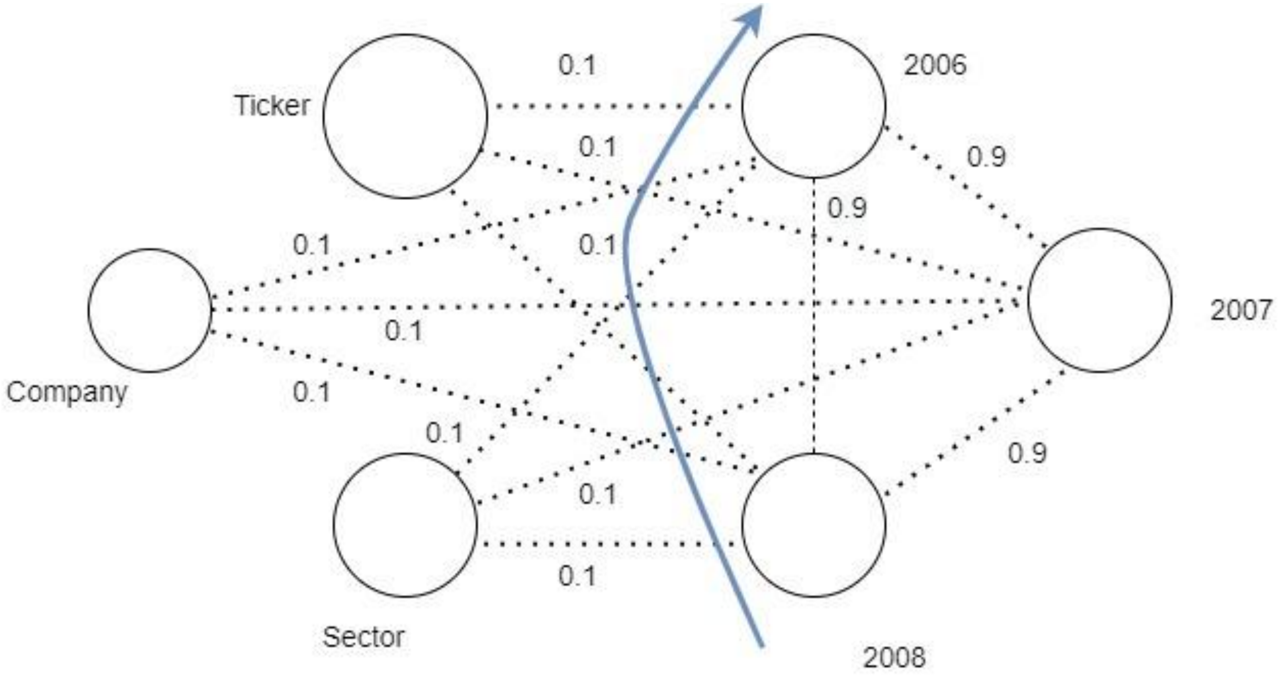
- Highest compatibility score: 2007, 2008
- Average intra-group compatibility

  = 0.9
- Average compatibility between selected and unselected columns

  = (0.1 * 6 + 0.9 * 2)/ 8

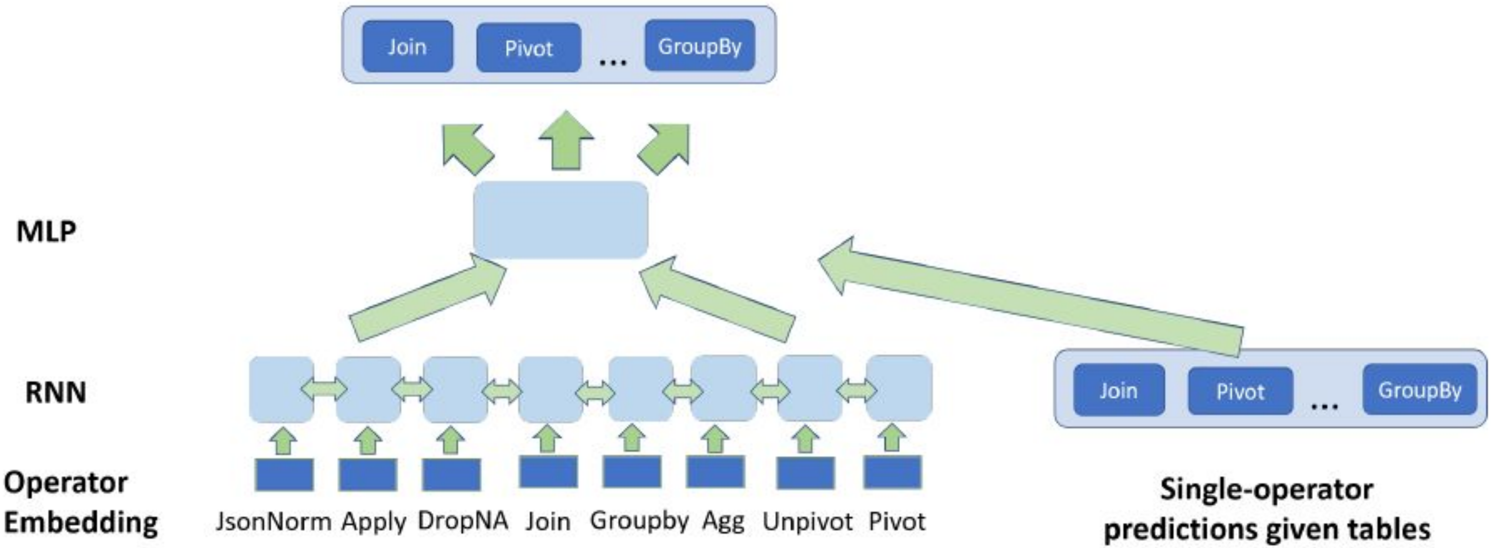
  = 0.3
- Objective function = 0.6

Example : Unpivot



- Compatibility score: 2006 with 2007 & 2008
- Average intra-group compatibility = 0.9
- Average compatibility between selected and unselected columns = 0.1
- Objective function = 0.8

Predict Next operator



- At timestamp t_i, predict next likely op at time i+1 given previously invoked ops and input table at time i

Evaluation: Dataset

- The data set was created by replaying and instrumenting a large number of Jupyter Notebooks.
- Filter identical or uninformative invocations.

operator	join	pivot	unpivot	groupby	normalize JSON
#nb crawled	209.9K	68.9K	16.8K	364.3K	8.3K
#nb sampled	80K	68.9K	16.8K	80K	8.3K
#nb replayed	12.6K	16.1K	5.7K	9.6K	3.2K
#operator replayed	58.3K	79K	7.2K	70.9K	4.3K
#operator post-filtering	11.2K	7.7K	2.9K	8.9K	1.9K

Evaluation metrics

- Precision@K = proportion of relevant predictions in K in the top-Ks

- Normalized Discounted Cumulative Gain (NDCG@K)

$$\mathrm{NDCG}_K = \frac{\mathrm{DCG}_K}{\mathrm{IDCG}_K}$$

where,

$$\mathrm{DCG}_K = \sum_{i=1}^{K} \frac{\mathrm{rel}_i}{\log_2(i+1)}$$

Evaluation - Join

method (all data)	prec@1	prec@2	ndcg@1	ndcg@2
Auto-Suggest	**0.89**	**0.92**	**0.89**	**0.93**
ML-FK	0.84	0.87	0.84	0.87
PowerPivot	0.31	0.44	0.31	0.48
Multi	0.33	0.4	0.33	0.41
Holistic	0.57	0.63	0.57	0.65
max-overlap	0.53	0.61	0.53	0.63
method (sampled data)	prec@1	prec@2	ndcg@1	ndcg@2
Auto-Suggest	**0.92**	-	**0.92**	-
Vendor-A	0.76	-	0.76	-
Vendor-C	0.42	-	0.42	-
Vendor-B	0.33	-	0.33	-

- Top methods from literature, bottom from commercial systems
- ML-FK, PowerPivot, Multi, Holistic designed for foreign key joins

Evaluation - Join Feature Group Importance

feature	left-ness	val-range-overlap	distinct-val-ratio	val-overlap
importance	**0.35**	**0.35**	**0.11**	**0.05**
feature	single-col-candidate	col-val-types	table-stats	sorted-ness
importance	**0.04**	**0.01**	**0.01**	**0.01**

- Left-ness and val-range-overlap more important features for ad-hoc joins by data scientists in the wild compared to val-overlap
    - Suggests accidental val-overlap may be common in practice

Evaluation - Join Type

method	prec@1
AUTO-SUGGEST	**0.88**
Vendor-A	0.78

- Vendors default to use inner-join → 78% of cases are indeed inner-joins

Evaluation - GroupBy

method	prec@1	prec@2	ndcg@1	ndcg@2	full-accuracy
AUTO-SUGGEST	**0.95**	**0.97**	**0.95**	**0.98**	**93%**
SQL-history	0.58	0.61	0.58	0.63	53%
Coarse-grained-types	0.47	0.52	0.47	0.54	46%
Fine-grained-types	0.31	0.4	0.31	0.42	38%
Min-Cardinality	0.68	0.83	0.68	0.86	68%
Vendor-B	0.56	0.71	0.56	0.75	45%
Vendor-C	0.71	0.82	0.71	0.85	67%

Evaluation - GroupBy Feature Importance

feature	col-type	col-name-freq	distinct-val	val-range
importance	**0.78**	**0.11**	**0.06**	**0.02**
feature	left-ness	peak-freq	empti-ness	
importance	**0.01**	**0.01**	**0.01**	

- Col-type most important - nothing new here
- Col-name-freq 2nd most important → prior knowledge on what columns are likely GroupBy
    - After seeing enough examples, knowing that columns named “year” are Groupby and not Agg

Evaluation - Pivot - Index vs. Header

method	full-accuracy	Rand-Index (RI)
AUTO-SUGGEST	**77%**	**0.87**
Affinity	42%	0.56
Type-Rules	19%	0.55
Min-Emptiness	46%	0.70
Balanced-Cut	14%	0.55

$$RI = \frac{\text{\#-correct-edges}}{\text{\#-total-edges}}$$

- No existing features for pivot, so compare with some related methods
- RI: how close the predicted split is to the ground-truth
  - An edge is correct if assignments of the two vertices incident to e are the same in the prediction and ground-truth (in same cluster or not)
  - Gives partial credit to predictions close enough to ground-truth

Evaluation - Unpivot

method	full accuracy	column precision	column recall	column F1
AUTO-SUGGEST	**67%**	**0.93**	**0.96**	**0.94**
Pattern-similarity	21%	0.64	0.46	0.54
Col-name-similarity	27%	0.71	0.53	0.61
Data-type	44%	0.87	0.92	0.89
Contiguous-type	46%	0.80	0.83	0.81

- Compare Auto-suggest with related methods
- 90% of the columns have an overlap with the ground-truth
    - Full accuracy is 67% because of the partially correct marked as incorrect

Evaluation - Predict Next Operator

method	prec@1	prec@2	recall@1	recall@2
AUTO-SUGGEST	**0.72**	**0.79**	**0.72**	**0.85**
RNN	0.56	0.68	0.56	0.77
N-gram model	0.40	0.53	0.40	0.66
Single-Operators	0.32	0.41	0.32	0.50
Random	0.23	0.35	0.24	0.42

- Auto-Suggest = RNN + Single-Op

Conclusion

- Data driven approach to learn how data scientists manipulate data
- Capture best-practices from notebooks to recommend data preparation steps for non technical users in self service data prep software

Thank you

Auto-suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Summary

- Leveraged collective wisdom of data scientists for “self-service” data preparation
- Crawled huge number of data science notebooks from Github
- Recommends next steps to help speed up data preprocessing coding
    - Single Operator Prediction
        - Join column prediction
        - Group By/Aggregation
        - Pivot
        - Unpivot
    - Next Operator Prediction : (*i + 1)th* step in the pipeline

Strengths

- Comprehensive analysis of how and which functions are used (eg: sum vs average)
- Detailed description of how prediction is done for all operators
- Extracted detailed information of function calls
- Notebook repair
    - Installed possible packages based on the errors
    - Found missing files
- Kept track of sequence of operations using a data-flow graph
- First attempt at harvesting invocations of diverse table-manipulation operations
- It’s a generic approach that can be potentially deployed on enterprise systems

Weaknesses / Open questions

- No information about compute used & time taken for crawling and running the offline system
- Would updates to already crawled notebooks be used?
- How are different python versions handled?
- Default parameters aren't recorded but they can change even in minor version upgrades
- Multiple files with same name and same distance from the root
- How do they verify correctness of the files?
- Some notebooks may be malicious and might corrupt the system

Weaknesses / Open questions

- Data frames may have two or more columns with same data (or subtle differences), how would this affect recommendations?
- Mainly focused on pandas and python
- User feedback/usage could be incorporated to supplement offline learning

ACCEPT

Review:
Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Shen En Chen

Summary of Contribution

The authors of the paper proposed Auto-Suggest, a contextualized smart data preparation framework that learns from Jupyter notebook workflow and recommends data prep operations to the user. Auto-Suggests provides improved recommendation quality on operations supported by prior research and commercial systems and extends its support to common but rarely supported operators such as “pivot” and “unpivot”. Compared to other work, Auto-Suggests is capable of recommending both the columns on which an operation should be applied and the next suitable operation given the current table. The authors developed a suite of heuristic-based features for the regression model on each prediction task, attaining much better performance than all baselines in most cases and discovering interesting counter-intuitive insights on the importance of different features . Algorithmically, the authors solves the column selection problem for the “pivot” operator in polynomial time with the Stoer-Wagner algorithm and that for the “unpivot” with a greedy algorithm.

Strong Points

1. Auto-Suggest avoids the potential costs of data collection and labeling by leveraging Jupyter notebooks publicly available on GitHub.
2. Auto-Suggest provides wide variety of operation predictions. It supports both single- and next-operator prediction. For the latter, it even offers 7 different operators as recommendation candidates.
3. Auto-Suggest outperforms all of the existing work and commercial products compared.
4. The authors framed the predictions for “pivot” and “unpivot” as Affinity Score Maximization and the Compatibility Maximization and solved them algorithmically in polynomial time.
5. Aside from recommendation quality, the experiments shed light on the differences between conventional wisdom and ad-hoc data preprocessing through investigating feature importances.

Opportunities for Improvement

1. The authors used the workflow in the notebooks crawled as a proxy of the ground truth. While this saves costs and covers several different use cases, more should be investigated in the representativeness of the collected data: is the data distribution of these notebook workflow similar to that of the workflow of commercial products like Tableau and Power BI?

2. As powerful as the paper demonstrated Auto-Suggest to be, the framework is not publicly available.

3. On join column prediction, Auto-Suggest performs only slightly better than ML-FK. It might be able to achieve better performance it incorporates the carefully engineered features of ML-FK.

4. For next-operator prediction, the authors did not compare Auto-Suggest against comperical systems such as the predictive-transformation in Trifacta and smart-suggestion in Salesforce Analytics Data Prep.

Overall Evaluation

Weak Accept

AUTO-SUGGEST: LEARNING-TO RECOMMEND DATA PREPARATION STEPS USING DATA SCIENCE NOTEBOOKS

ARCHEOLOGIST PRESENTATION 

ANIRUDDHA MYSORE

THEMES IN THE PAPER

Data-preparation operation recommendation

Data mining open-source code, specifically Jupyter notebooks

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Prior works

Search... Expand

Origin paper
Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks
Cong Yan, Yeye He 2020

Auto-Pipeline: Synthesize Data Pipelines By-Target Using Reinforcement Learning and Search
Junwen Yang, Yeye He, S. Chaudhuri 2021

Auto-transform
Zhongjun (Mark) Jin, Yeye He, S. Chauduri 2020

Auto-Transform: Learning-to-Transform by Patterns
Yeye He, Zhongjun (Mark) Jin, S. Chaudhuri 2020

Transform-Data-by-Example (TDE): An Extensible Search Engine for Data Transformations
Yeye He, Xu Chu, K. Ganjam, Yudian Zheng, V.... 2018

Uni-Detect: A Unified Approach to Automated Error Detection in Tables
Pei Wang, Yeye He 2019

Spine: Scaling up Programming-by-Negative-Example for String Filtering and Transformation
Chaoji Zuo, Sepehr Assadi, Dong Deng 2022

Unifacta: Profiling-driven String Pattern Standardization
Zhongjun (Mark) Jin, Michael J. Cafarella, H. Jagadish,... 2018

BlinkFill: Semi-supervised Programming By Example

Prior works

Download

These are papers that were most commonly cited by the papers in the graph.

This usually means that they are **important seminal works** for this field and it could be a good idea to get familiar with them.

Selecting a prior work will highlight all graph papers referencing it, and selecting a graph paper will highlight all referenced prior work.

Title
Wrangler: interactive visual specification of data transformation scripts
Detecting Data Errors: Where are we and what needs to be done?
KATARA: A Data Cleaning System Powered by Knowledge Bases and Crowdsourcing
Automating string processing in spreadsheets using input-output examples
Potter's Wheel: An Interactive Data Cleaning System
Spreadsheet data manipulation using examples
FlashExtract: a framework for data extraction by examples
Holistic data cleaning: Putting violations into context
Spreadsheet table transformations from examples

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Prior works

Search... Expand

Origin paper
Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks
Cong Yan, Yeye He 2020

Auto-Pipeline: Synthesize Data Pipelines By-Target Using Reinforcement Learning and Search
Junwen Yang, Yeye He, S. Chaudhuri 2021

Auto-transform
Zhongjun (Mark) Jin, Yeye He, S. Chauduri 2020

Auto-Transform: Learning-to-Transform by Patterns
Yeye He, Zhongjun (Mark) Jin, S. Chaudhuri 2020

Transform-Data-by-Example (TDE): An Extensible Search Engine for Data Transformations
Yeye He, Xu Chu, K. Ganjam, Yudian Zheng, V.... 2018

Uni-Detect: A Unified Approach to Automated Error Detection in Tables
Pei Wang, Yeye He 2019

Spine: Scaling up Programming-by-Negative-Example for String Filtering and Transformation
Chaoji Zuo, Sepehr Assadi, Dong Deng 2022

Unifacta: Profiling-driven String Pattern Standardization
Zhongjun (Mark) Jin, Michael J. Cafarella, H. Jagadish,... 2018

BlinkFill: Semi-supervised Programming By Example

Derivative works

Download

These are papers that cited many of the papers in the graph.

This usually means that they are **either surveys of the field or recent relevant works** which were inspired by many papers in the graph.

Selecting a derived work will highlight all graph papers cited by it, and selecting a graph paper will highlight all derivative works citing it.

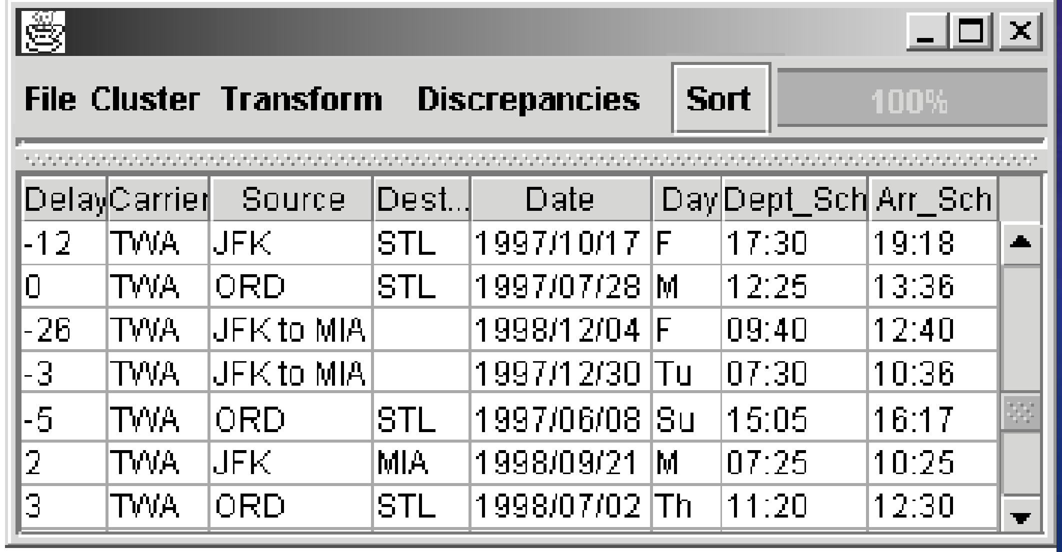
Title	Last author	Year	Citations	Graph references
Machine Learning and Data Cleaning: Which Serves the Other?	Theodoros Rekatsinas	2022	2	8
From Cleaning before ML to Cleaning for ML	Eugene Wu	2021	7	8
Automatic Error Correction Using the Wikipedia Page Revision History	Mohammad Mahdavi	2021	0	7
SPADE: A Semi-supervised Probabilistic Approach for Detecting Errors in Tables	J. Pujara	2021	0	7
TabReformer: Unsupervised Representation Learning for Erroneous...	Shaikh Quader	2021	0	6
Automating Data Quality Validation for Dynamic Data Ingestion	Sebastian Schelter	2021	6	6
Sudowoodo: Contrastive Self-supervised Learning for Multi-purpose Data Integrati...	Jin Wang	2022	0	5
Localizing Violations of Approximate Constraints for Data Error Detection	Mohan Zhang	2020	0	5
Data Errors: Symptoms, Causes and Origins	V. Markl	2022	0	5

RECOMMENDING DATA CLEANING OPERATIONS: THE TIMELINE

- 2001 – **Potter's wheel**
- 2011 - **Wrangler**
- 2020 – **Auto-Suggest**
- 2021- **Auto-Pipeline**
- 2022 - **Machine Learning and Data Cleaning: Which Serves the Other?**

[PRIOR WORK] 2001 – POTTER'S WHEEL

- Interactive data cleaning system – immediate feedback rather than batched transforms
- Infers structure of data
- Automatic discrepancy detection on applying transform



Example Values Split By User (| is user specified split position)	Inferred Structure	Comments
Taylor, Jane |, $52,072 Blair, John |, $73,238 Tony Smith |, $1,00,533	$(<\xi^*><\text{','}\ \mathit{Money}>)$	Parsing is doable despite no good delimiter. A *regular expression* domain can infer a structure of $[0-9,]* for last component.
MAA |to| SIN JFK |to| SFO LAX |–| ORD SEA |/| OAK	$(<\mathit{len\ 3\ identifier}><\xi^*><\mathit{len\ 3\ identifier}>)$	Parsing is possible despite multiple delimiters.
321 Blake #7 |, Berkeley |, CA 94720 719 MLK Road |, Fremont |, CA 95743	$(<\mathit{number}\ \xi^*><\text{','}\ \mathit{word}><\text{','}\ \mathit{(2\ letter\ word)\ (5\ letter\ integer)}>)$	Parsing is easy because of consistent delimiter.

Figure 10: Parse structures inferred from various split-by-examples

[PRIOR WORK] 2011 – WRANGLER

- Interface for transforming data + declarative transformation language
- Automatically suggests new operations
- Dataset – past user interactions on the same data

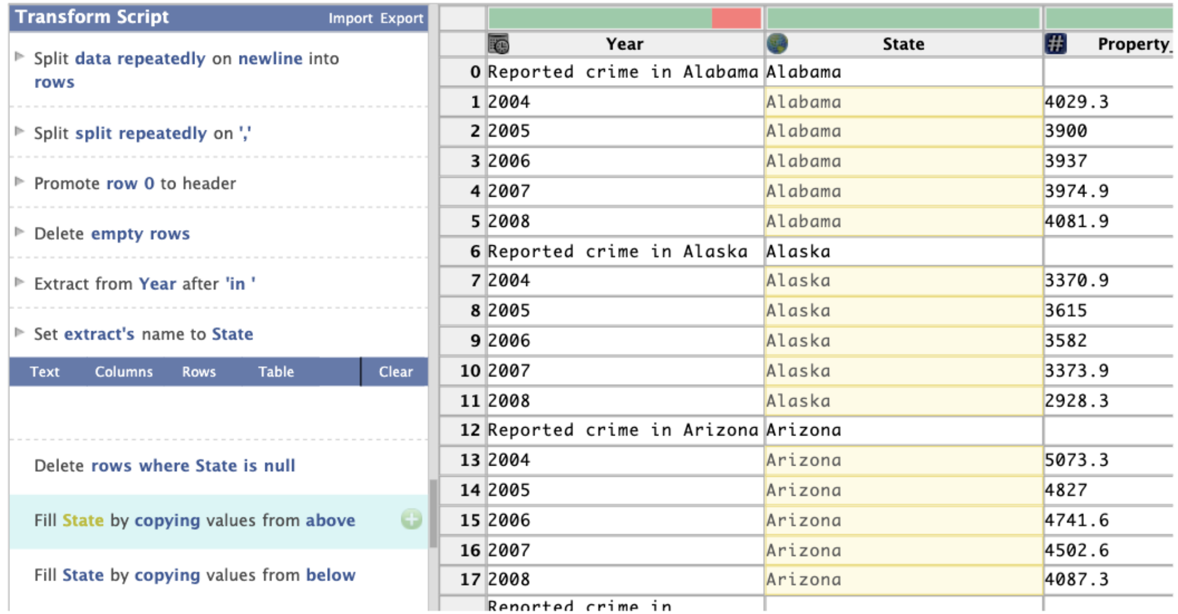
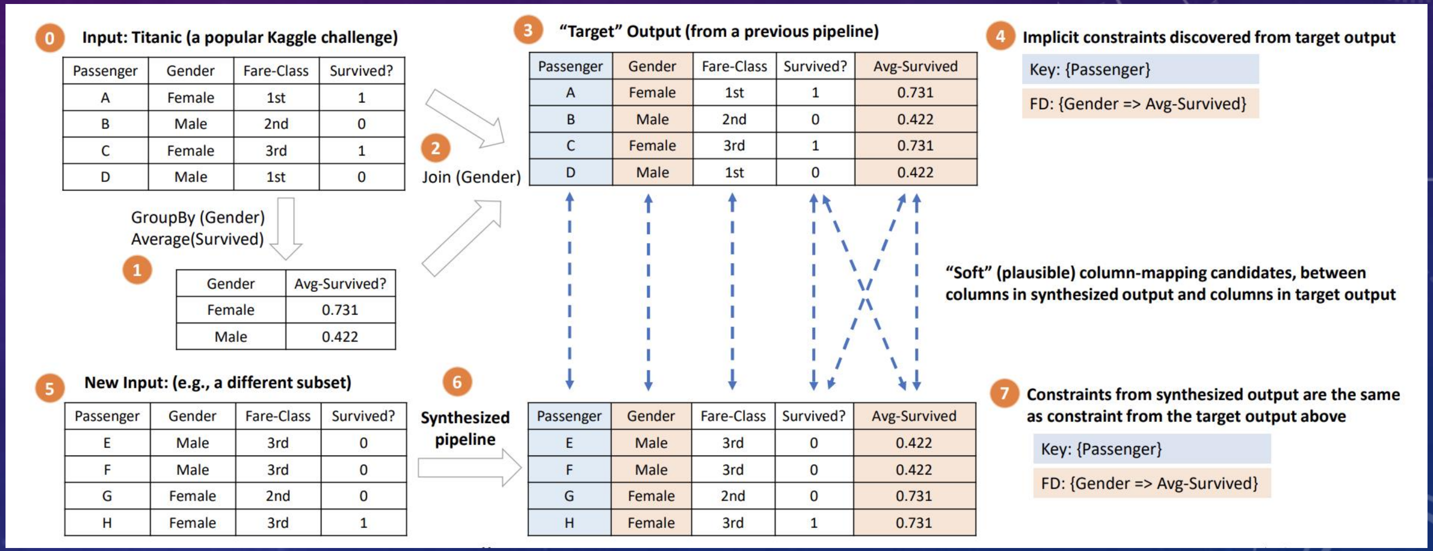


Figure 4. Filling missing values. The analyst populates empty cells by clicking the gray bar (Fig. 3) in the data quality meter above the "State" column, and then selecting a *fill* transform.

[LATER WORK] 2021 – AUTO PIPELINE

- Combine multiple operators
  - Table operators: Join, Group By, Pivot
  - String operators: Split, substring, Index
- Synthesize end-to-end pipeline using Reinforcement Learning
- “by-target” paradigm
- Dataset - Jupyter notebooks



[LATER WORK] 2022 – MACHINE LEARNING & DATA CLEANING – WHICH SERVES THE OTHER

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Researcher: Ting Yu

What is proven to be successful?

- **Jupyter notebooks offer valuable insights** into how data scientists work. The paper provides a hands-on framework on how to put such notebooks crawled from GitHub into use.
- Single-operator prediction: **useful heuristic metrics** that are proven to be effective in predicting single-operators
- Next-operator prediction: the **value of using the sequence of preceding operators** in improving predictions is proven when compared with single-operator prediction based purely on characteristics of input tables

Next step - Simplify further for non-technical analyst or auto ETL

Bit:

- Predicting a single data preparation step

Given:

- Original data from online Jupyter notebooks can be found.

Flip:

- Auto-generate a complete data preparation pipeline given tables at interest.
- We may do this by find notebooks that work on a “similar tables” (defined by some distance metric based on table characteristics).

Next step - Focus on other parts of Jupyter notebooks

Bit:

- Predictions help automate data preparation stage

Given:

- Many jupyter notebooks include code on data import, serialization, visualization using a few standard libraries.

Flip:

- Automate other stages such as the data exploration stage.
- In particular, we may predict parameters of matplotlib parameters to allow building graphs with tickers, titles, axis, graph types without having to specify them, all within one command “plt.autoplot(Data)”.

Next step - Generalize the method to other tools

Bit:

- Prediction for next Pandas operation

Given:

- Pandas dataframe is a rich super-set of SQL

Flip:

- Predict the next SQL query with SQL history.
- We may also translate Pandas into SQL queries, loosely treating all the notebooks the SQL history.

CS8803

Auto-Suggest: Learning-to-Recommend Data Preparation Steps Using Data Science Notebooks

Practitioner role: Jingfan Meng

10/5/2022

Georgia Tech

Why we need "self-service" data preparation?

- Data preparation is "the most time-consuming step in analytics".

 By Gartner

Two reasons:

- It takes expertise knowledge to know which operations to perform, and takes many trials to make a decision.
- If a bad decision is discovered at later stages, rolling back means a lot of wasted effort.

Georgia Tech

Why we need "self-service" data preparation?

- Auto Suggest learns how expert data scientists prepare data from existing Pandas scripts, and makes intelligent suggestions on which operation to perform on the tables.

- Two-fold benefit to our Data Analytics group:
- Less errors and increased productivity.
- Less training effort on newcomers.

Georgia Tech

Join, Group-by, Aggregation

- They are most widely-used operators in our codes. Hence, the advances will significantly improve productivity.

- Auto Suggest predicts join columns and group-by (dimension) columns than our current tool.
- It also has a new feature: Predict the join type (inner/outer).

Pivot and Unpivot

- Although not as frequent, these are the hardest operators for analysts.
- Some colleagues complain that they always have too many NULLs in the tables.
- Auto Suggest saves the day.

Ticker	Company	Year	Aerospace	Business Services	...	Utilities
AJRD	AEROJET ROCKETD	2006	6218.09	NULL	...	NULL
AJRD	AEROJET ROCKETD	2007	6342.45	NULL	...	NULL
AJRD	AEROJET ROCKETD	2008	7088.62	NULL	...	NULL
ATRO	ASTRONICS CORP	2006	1050.97	NULL	...	NULL
...	...	...	...	...	...	...
HHS	HARTE-HANKS INC	2006	NULL	2473.75	...	NULL
...	...	...	...	...	...	...
YORW	YORK WATER CO	2008	NULL	NULL	...	2168.7

Discussions

- Multi-operator prediction?
  - We can develop this feature after we finish and pilot single-operator predictions.
- Which training data to use?
  - Open source notebooks: Readily available, large in volume, but might not best suit our data and tasks.
  - Corporate code: Best suited for our task, but limited in volume. Need adaptation and permission.

Discussions (cont.)

- What if our analysts become reliant on Auto Suggest rather than domain knowledge?
- This is a legitimate issue. We need to know in which cases Augo Suggest can be improved by our domain knowledge. To this end, a possibmonitor feedbacks from users to see if this is an issue.

Georgia Tech

Thanks!

Contribution/Strengths

- Built a system to crawl jupyter notebooks and data pipelines at scale – could handle error cases including missing packages and absolute path issues.
- First data driven operator predictor which relies on real user data.
- Experiments also shed light on the differences between conventional wisdom and ad-hoc data preprocessing; for example, left-ness and val-range-overlap are more useful than value-overlap in predicting join columns.
- Extends prior work in automated suggestions to new operators such as pivot and unpivot

Limitation/Weaknesses

Training and maintenance challenges

- How frequently one needs to gather data to ensure the models are up to date with current data science trends
- Replaying is costly and not always feasible (for lack of data). It is possible to avoid replaying by analyzing the scripts themselves, or to analyze these features without actually running on real data, or to use some fictitious data when the original data is unavailable?
- If users come to rely on these predictions in the same way users rely on the results of a Google search, then there could be a chance that the incorrect parameters and operators could be routinely chosen reinforcing bad habits.

Limitation/Weaknesses

Bias/error in data

- Publicly crawled code can contain many bugs, especially since the authors make no attempt to curate their sources.
- Did not collect default parameters of methods
- It would also be interesting to examine the purposes of notebooks used as the training data and analyze any potential biases of using GitHub as the only crawling source. For example, are Trifacta users different from Pandas users as a result of having different user interfaces? If so, how will this difference affect the prediction task?
- Many commercial systems use black-box algorithms that are likely trained on data analytics workflows performed on their systems, there might exist a distribution shift in their training data and test data of Auto-Suggest. The poor performance of these systems might be attributed to their poor robustness on distribution shift instead.
- Some features (such as leftness) seems arbitrary. While it is possible that some users are prone to group-by left columns, I think it is more of a matter of personal preference. Using such features will introduce some preference bias to the prediction model.

Extensions/Open Questions

- Integrating the system with popularly used IDEs and collaborative editors for notebooks could be another future work (like GitHub copilot).
- This can also be extended to have a human-in-the-loop approach where the feedback from the user is then taken into account to improve the system recommendations.
- It might not be the most practical to recommend operations prior to a user actually exploring the data. So one open question I had was whether Auto-Suggest can be used as a standalone tool or requires some level of data exploration beforehand.
- Can you precompute suggested operations to reduce latency to users?

Next class

[Towards Effective Foraging by Data Scientists to Find Past Analysis Choices]

Author: Myna

Reviewer: Tanya, Siddhi

Archaeologist: Sahil

Practioner: Cangdi

Researcher: Ting