Scalable Content-based Modeling for Big Data Tasks

Dimitrios Tsoumakos

Associate Professor

Dept. of Informatics, Ionian University

Computing Systems Lab, NTUA

Data Processing and Analysis

MSC – Research Directions in Informatics, Spring 2020

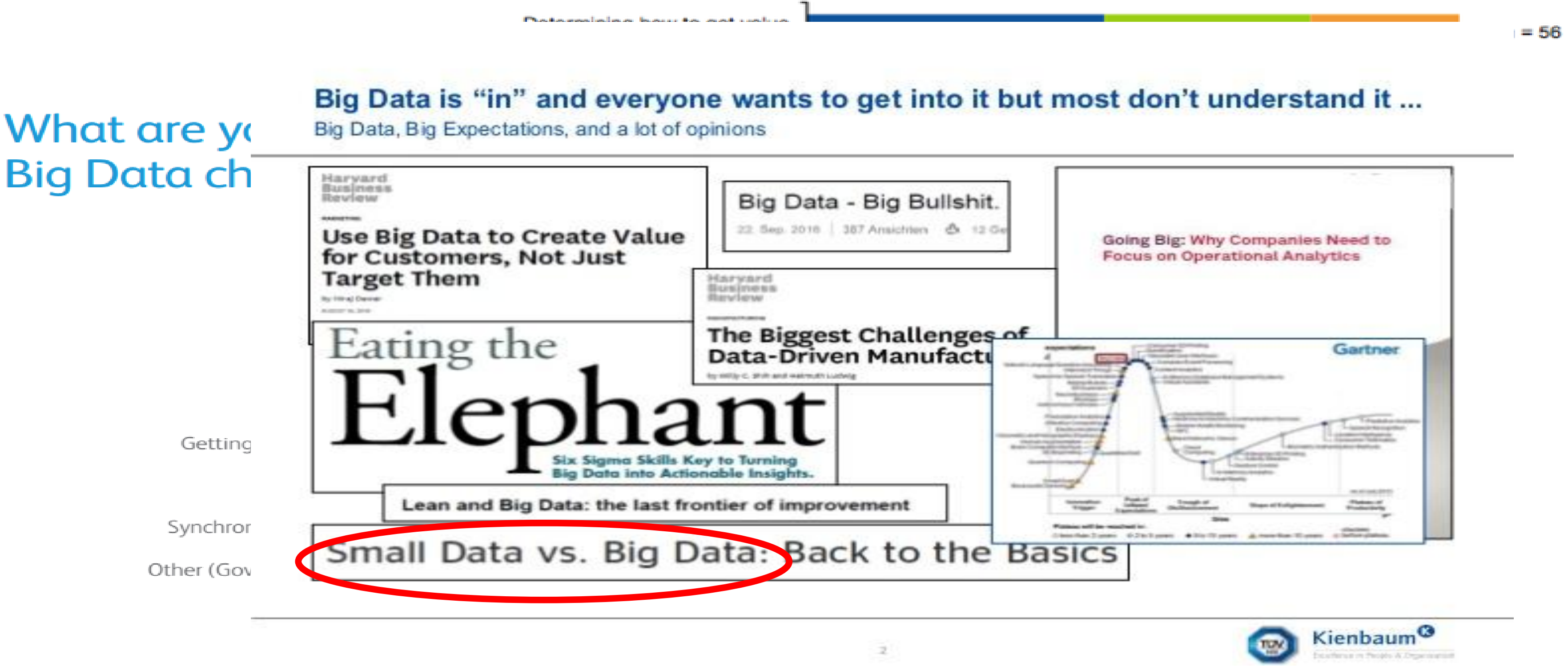
Big data era

Data is everywhere

- Social networks
- IoT devices/trackers
- Smartphones
- Data Lakes
- Business

Big data era (2)

Figure 7. Top Big Data Challenges



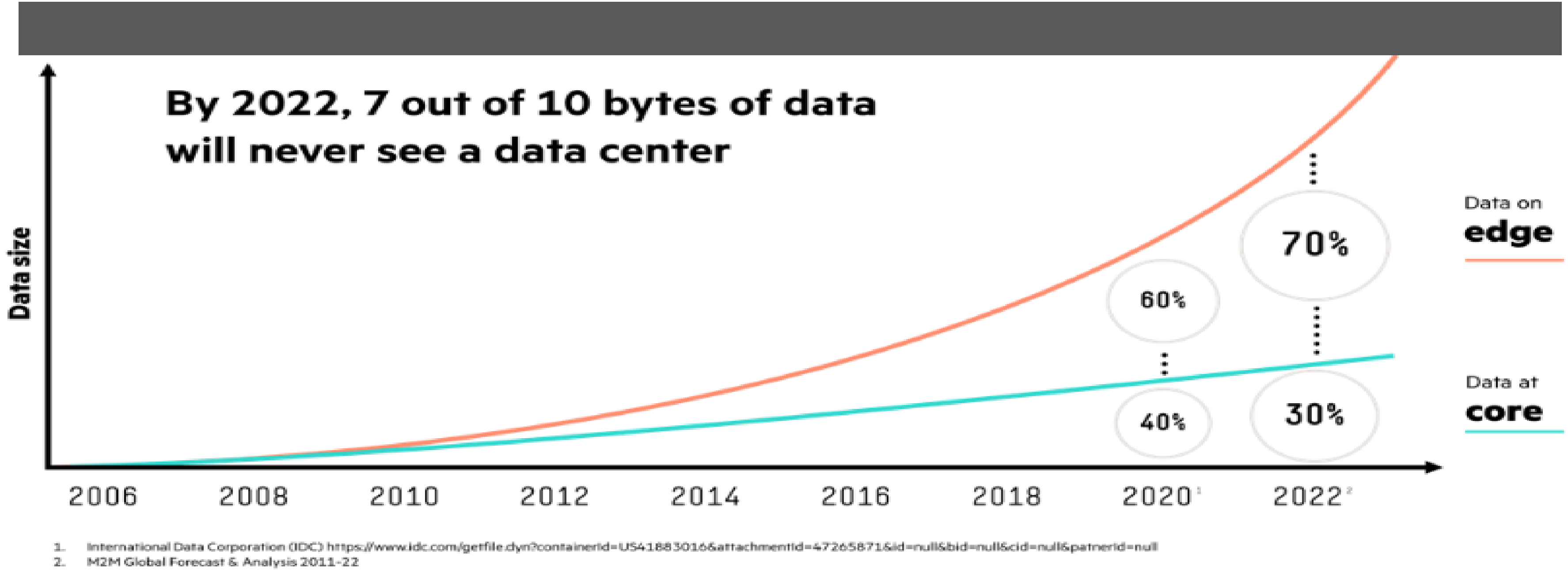
N = 687 (excludes "don't know" responses)

Source: Gartner (September 2013)

What’s really Big?

- Data is big
  - Crunching them is getting faster and faster
  - More resources, bigger speeds, better algorithms
- Heterogeneity dramatically increases complexity in executing a task!
  - #runtimes, #datastores
  - #resource configurations for deployment
  - #input datasets

What is really Big (2)



By 2022, seven out of every 10 bytes of data created will stay where they are created.

Data curve from IDC/EMC Digital Universe reports 2008-2017, Compute curve HPE analysis

40 years of Microprocessor Trend Data — Image: Karl Rupp

Big data Challenge

Big Data systems are harder to:

- Design
- Implement
- **Analyze**

HDFS Architecture

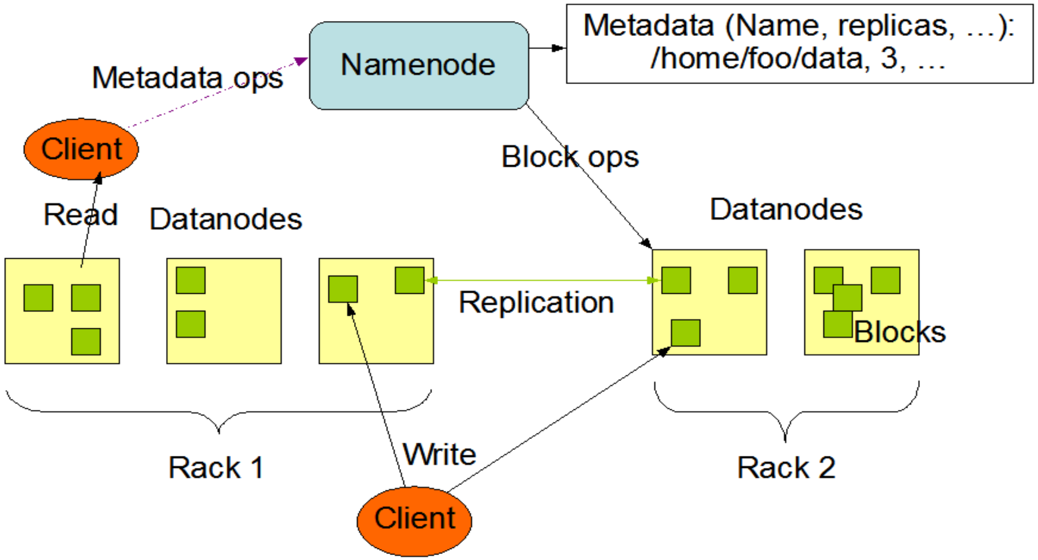


Image from [HDFS]

Modeling

Why care about modeling (in big data settings)?

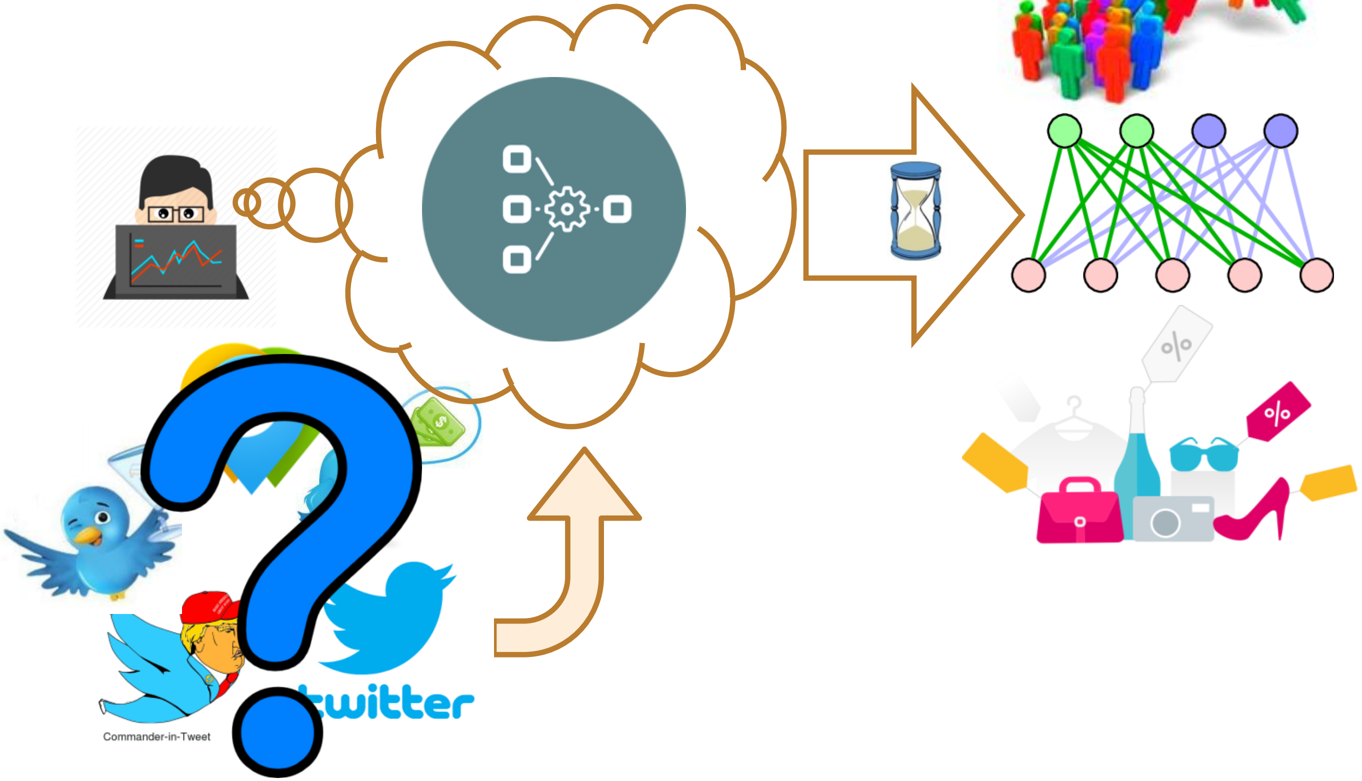
1. How does my app behave deployed under $\underline{x}$ amount of resources?
    a. Best deployment combo/Maximize cost-efficiency balance
    b. Elastic scaling capabilities/properties
    c. Improve architecture/identify bottlenecks
    d. Multi-engine execution environments
2. How does my app perform when consuming dataset(s) $\underline{y}$?
    a. Finding good training set for ML tasks
    b. Quickly spot dataset(s) of high interest/maximize accuracy of insights
    c. Targeted exploration without manual search

Content-Based Data Modeling for Analytics Operators

Discovering the “right” data

- A different type of challenge
    - Input data plays a huge role in achieving workflow goal(s)
    - Not size, but **content relevance** counts
- Examples:
    - Content-based marketing, web advertising, recommender systems
    - Healthcare (insurance, diagnosis, cost reduction)
    - Risk/credit analysis, fraud detection
    - Machine Translation
    - …

A marketer’s story



Interesting info for (any) data analyst

- What’s the expected output for a random (unseen) dataset?
- Rank all available datasets
- Which are the datasets that (for a given task):
    - Maximize accuracy, minimize time/cost…
    - Perform closest to a specific dataset
- But without testing each one of them
    - There are too many!
- And what if I change my workflow/task?
- New datasets arrive too (streaming mode)

Dataset-driven analytics profiling

- Predict operator performance over different input
- Operator-agnostic
  - Process largely independent of the analytics operators
- Scales for very big #datasets
  - Efficient + parallelizable process
  - Incremental updates (for unseen datasets)
- Extensible to other domains (graph data+operators now)
- Open source system implementation

Preliminaries

Problem statement

Given:

1. Operator F
2. Set of datasets $D = \{D_1, D_2, \ldots, D_n\}$

"Estimate the utility of each dataset D_i, $1 \le i \le n$, for the operator F."

or dually

"Find an approximation of the operator's output F when applied to all datasets D_i, $1 \le i \le n$."

Preliminaries

Challenges

- # of input datasets
  - *n* operator executions → too expensive in cost + time
  - Particularly for operators with high (computational) complexity

- # of different operators
  - Same datasets, different task applied
  - Repeat from scratch for each new operator

Methodology

- Observation
  - *Similar* datasets → *similar* operator outputs

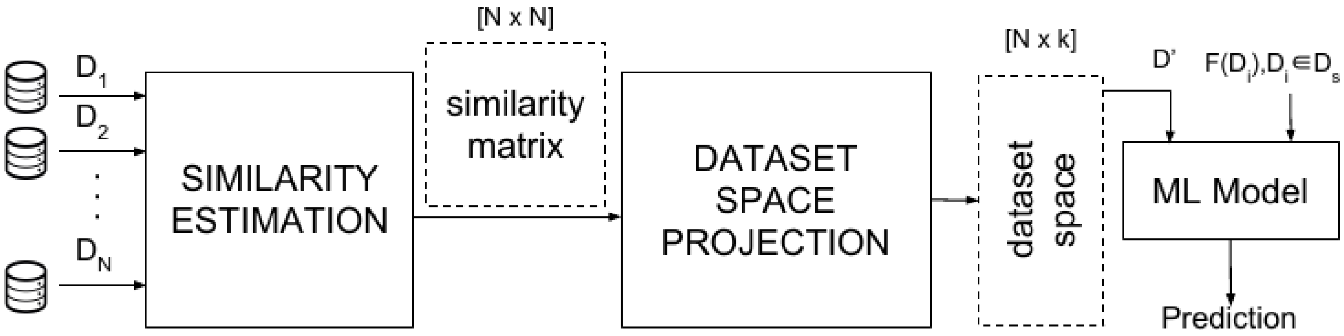
- Operator type:

$$F : D \rightarrow \mathbb{R}$$

- Data properties:
  - Statistical distribution
  - Dataset size
  - Tuple ordering

- Operator categories:
  - Aggregate functions (AVG, SUM, COUNT)
  - Density based (DBSCAN, Local Outlier Factor)
  - Linear Regression
  - Spectrum (Eigenvalue estimation)
  - Time-Series Forecast (*Holt-Winters*, *ARIMA*)

Methodology Workflow



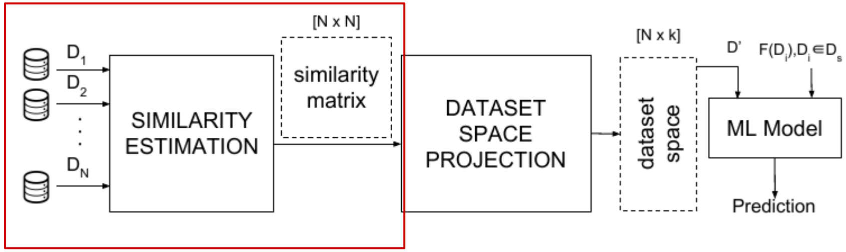
Methodology

Similarity Estimation - Distribution

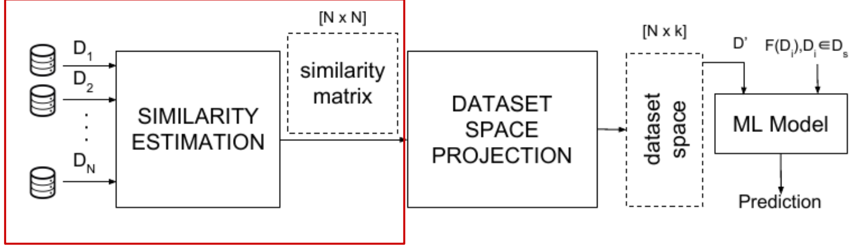
- Objective: quantify tuple-overlap among two datasets
- Normalized Bhattacharyya coefficient

$$Distribution(A, B) = \frac{\sum_{i=1}^{l} \sqrt{A_i B_i}}{\sqrt{|A||B|}}$$

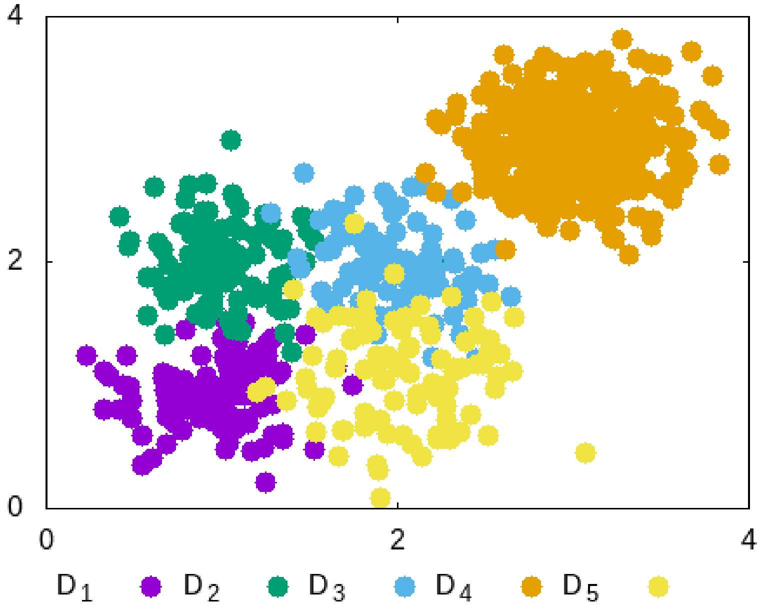
  - Partition the tuple space (*k-means partitioning*)
  - Count tuples cardinality for each partition for each dataset
  - Estimate Bhattacharyya coefficient for each pair of datasets



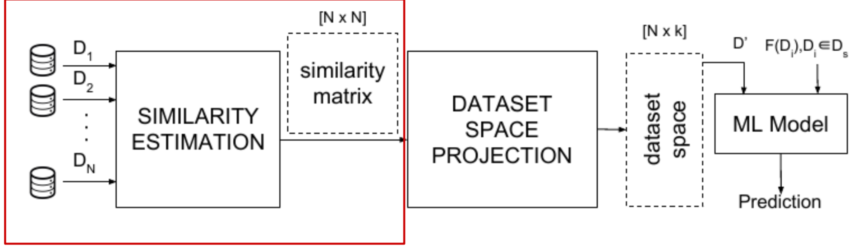
Methodology



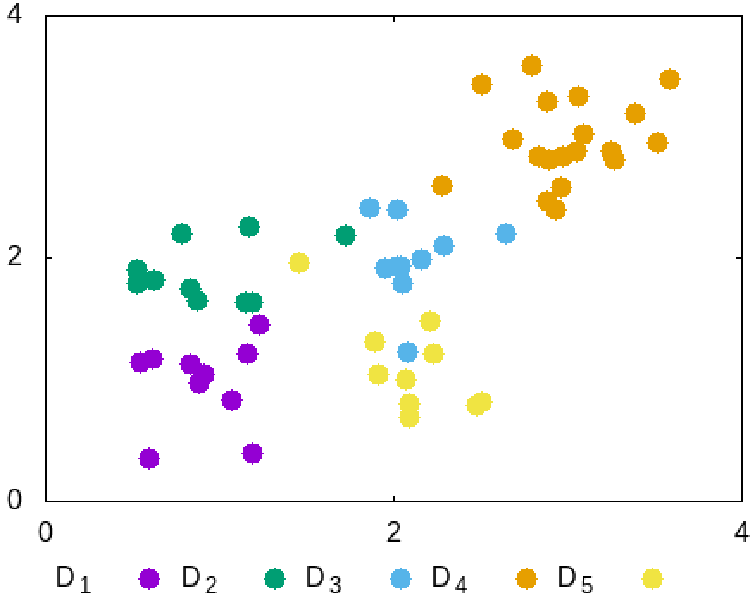
Similarity Estimation - Example



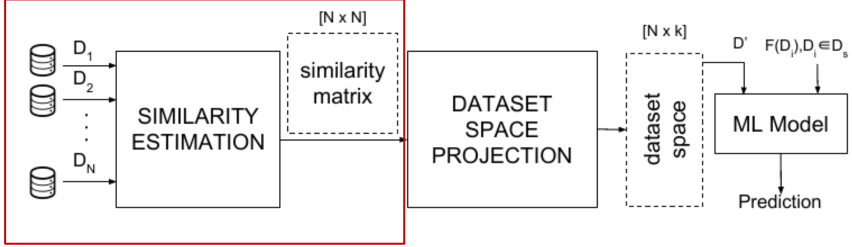
Methodology



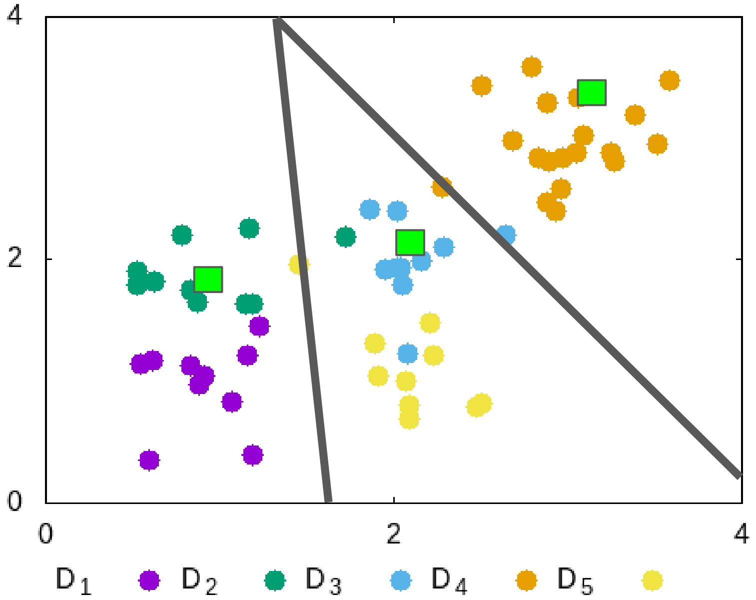
Similarity Estimation - Example



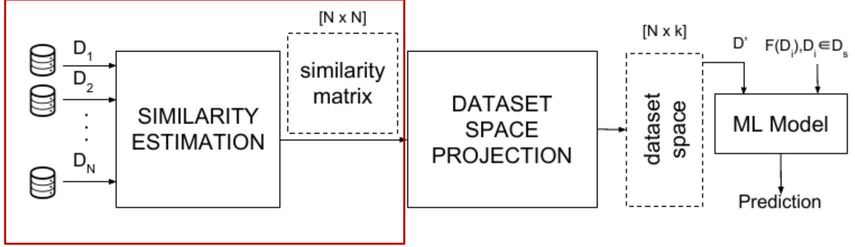
Methodology



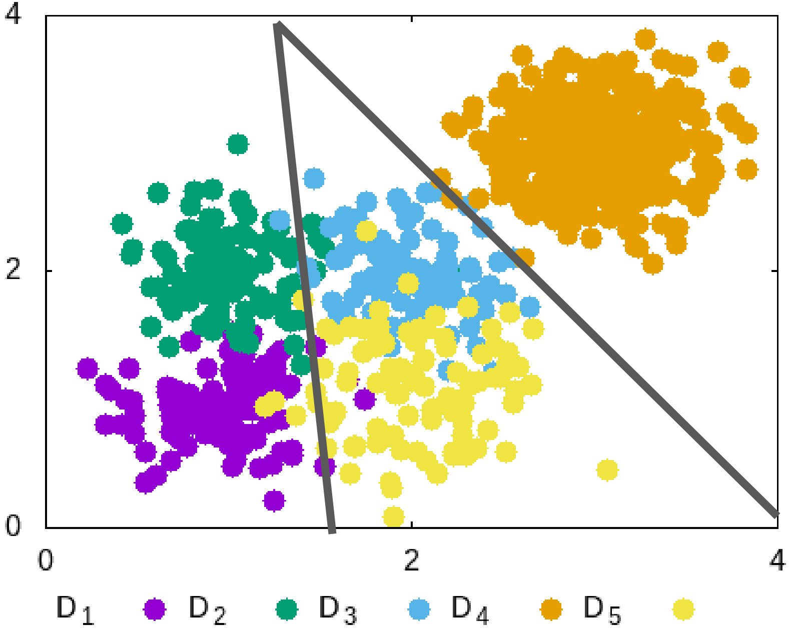
Similarity Estimation - Example



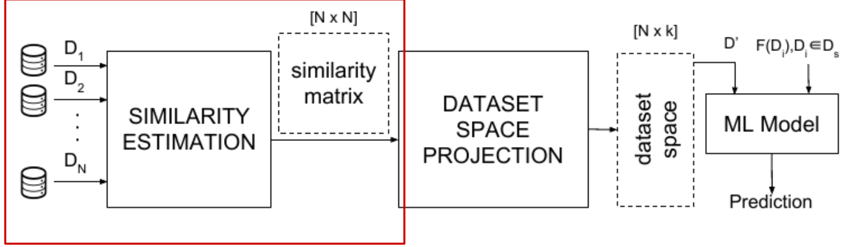
Methodology



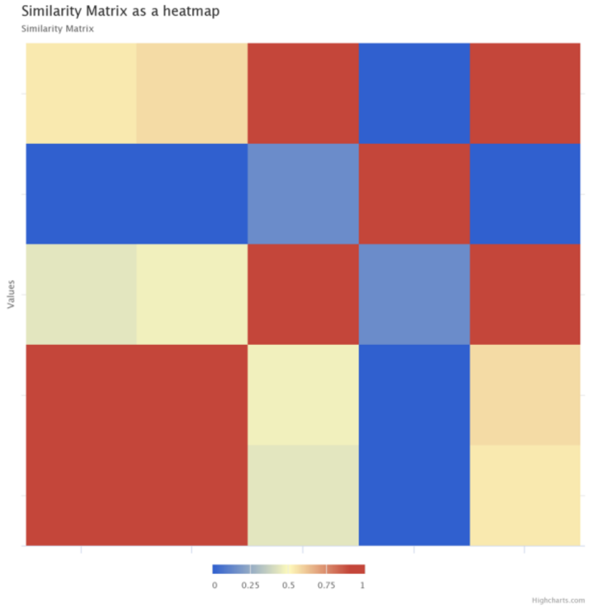
Similarity Estimation - Example



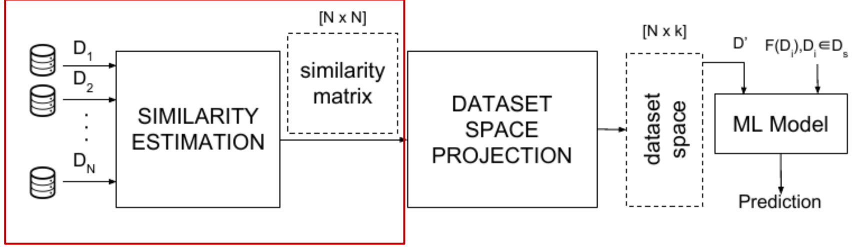
Methodology



Similarity Estimation - Example



Methodology



Similarity Estimation

Ordering

$$Order(A,B) = \frac{concord(a,b) - discord(a,b)}{n(n-1)} + \frac{1}{2}$$

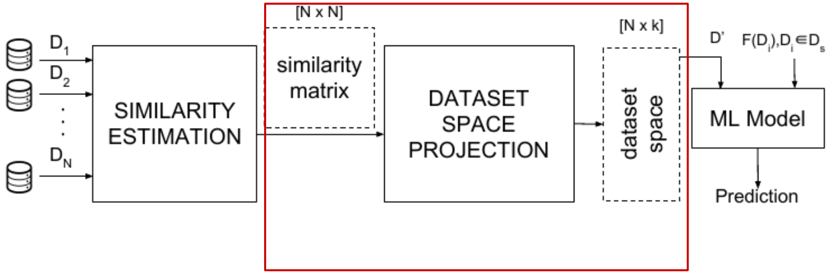
Size

$$Size(A,B) = \frac{\min(|A|,|B|)}{\max(|A|,|B|)}$$

And combinations:

- Linear combination of different Similarity Matrices

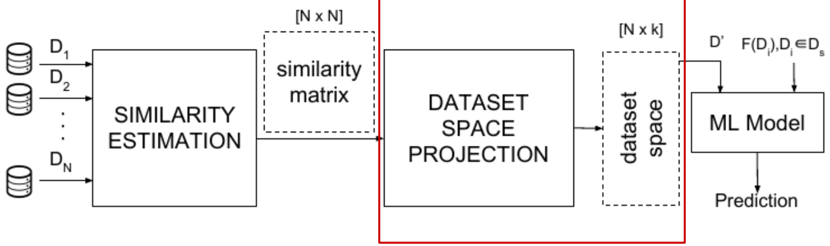
Methodology



Dataset Space Projection

- The similarity Matrix is useful, but:
  - Grows quadratically with # of datasets
  - Does not provide information at scale
  - Visualization with heatmap
- Idea: *transform Similarity Matrix to a low-dimensional space*
  - Each point represents a dataset
  - Similar datasets flock together in this space

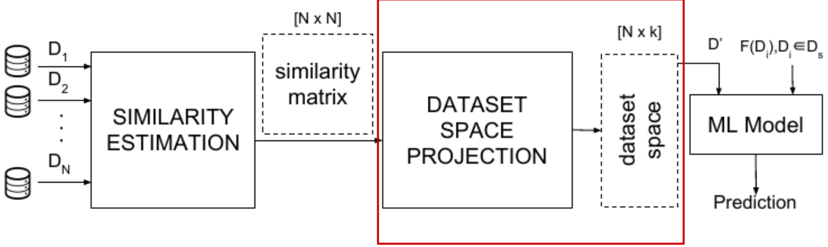
Methodology



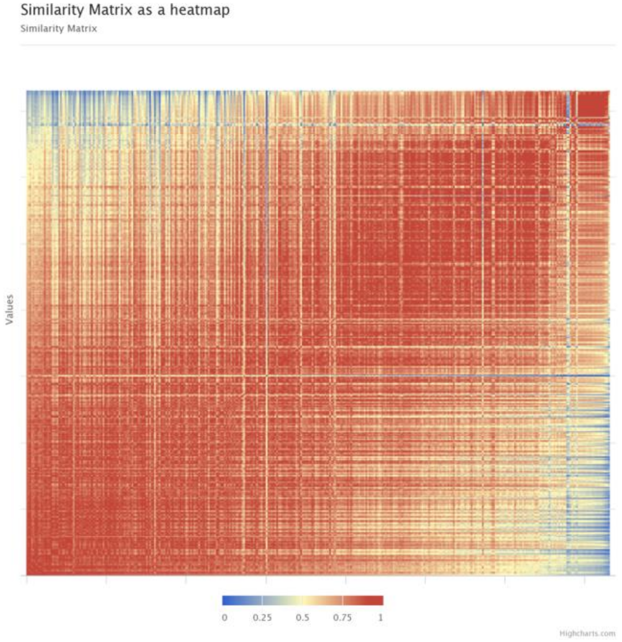
Dataset Space Projection

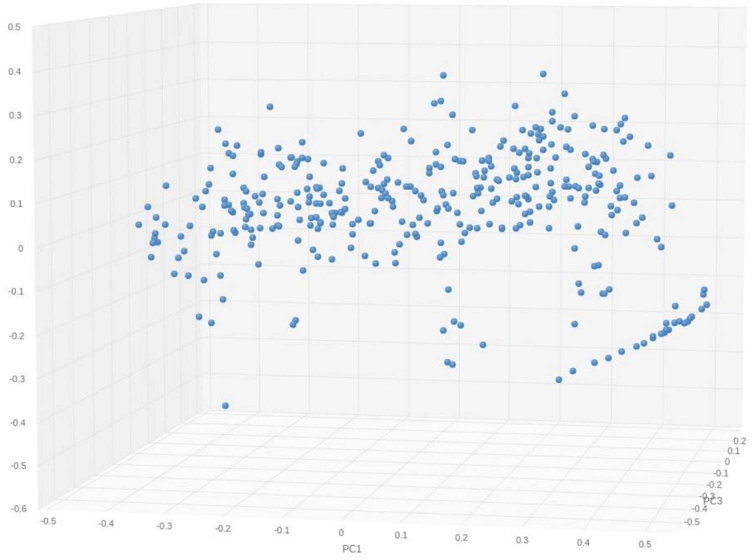
- Optimization problem:
  - *Given the pairwise distances between different points, find a set of k-dimensional coordinates that preserves these distances*
- Solution:
  - Eigenvalue optimization - **Multidimensional Scaling (MDS)**
    - Estimates space dimensionality (based on eigenvalues)
    - Estimates the set of coordinates
  - Non linear solution - **Sammon Mapping**
    - Starting off with a set of coordinates, slightly relocate points (datasets) to better fit the SM distances

Methodology

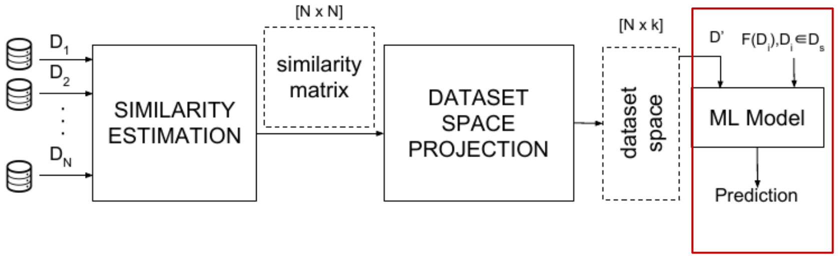


Dataset Space Projection - Example





Methodology



Modeling

- Execute $F(D_i)$ for a few datasets (e.g., 5% of them)

- Train a Machine Learning classifier to approximate operator values
  - 1-hidden layer Neural Network

Methodology

Key point:

Dataset space construction is *operator-agnostic*.

- We do not rely on operator output to create the space
- Examined data parameters are much less than the applicable operators

Let’s take a look

https://youtu.be/Bl9M-K8uwXw

Evaluation

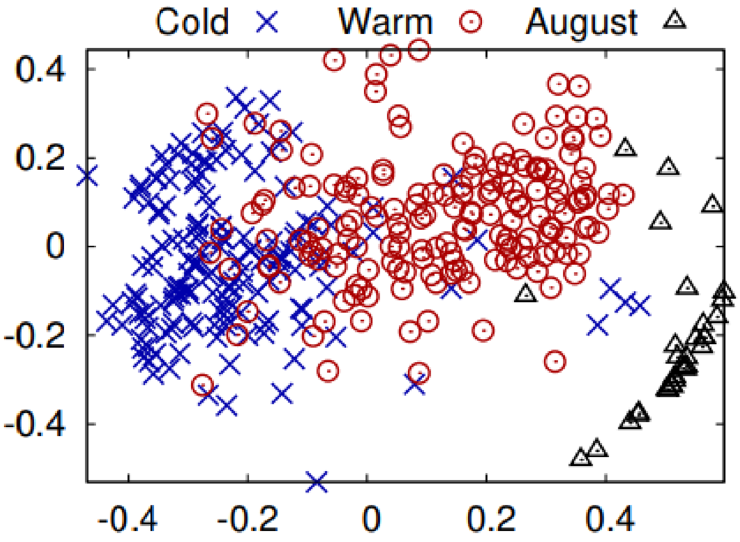
- Open Source Prototype in Go
- Experiments in private Openstack Cluster
  - Intel Xeon E5645 @2 .40GHz, 96G RM
- Evaluation
  - Modeling accuracy
  - Speedup
- Accuracy metrics:
  - NRMSE
  - MdAPE
- Space distortion
  - Goodness-of-Fit
  - Sammon Stress

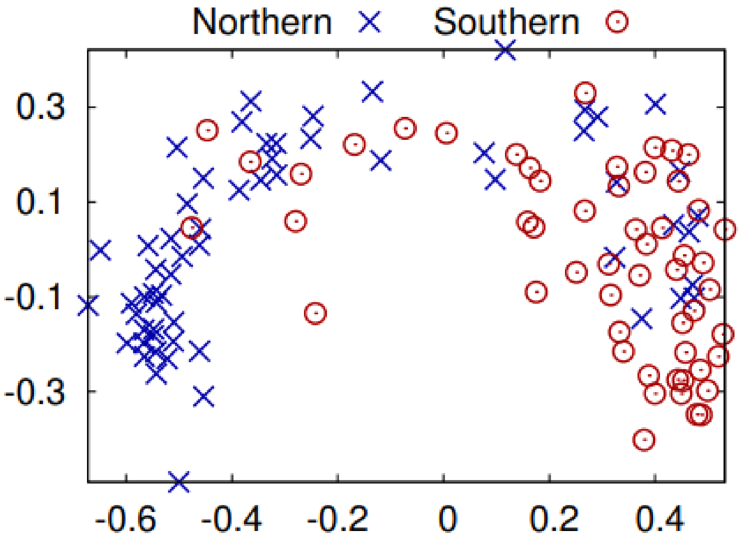
Operators		Affected by
Class	**Name**	
Aggregate Functions	AVG	Distribution
	SUM COUNT	Distribution + Size
Density	DBSCAN [23] Local Outlier Factor [18]	Distribution
ML	Linear Regression	Distribution
Spectrum	Eigenvalue Estimation	Distribution
Time-Series Forecast	Holt-Winters [19] ARIMA [17]	Distribution + Order

ID	Description	Datasets	Tuples	Operators
CLU	Google Cluster Monitoring [2]	4797	46 – 2188	**AVG**, **SUM**, COUNT (**CNT**), DBSCAN (**DBS**), Local Outlier F. (**LOF**), Eigenvalue (**EIG**), Regression (**REG**)
HPO	Household Power Consumption [35]	1442	1263 – 1440	
WEA	Weather Station Recordings [3]	552	300 – 8766	
NAS	NASDAQ Tech. Stocks [5]	231	252	Holt-Winters (**HOL**)
WIK	Wikipedia Page Visits [7]	4503	551	ARIMA (**ARI**)

Evaluation

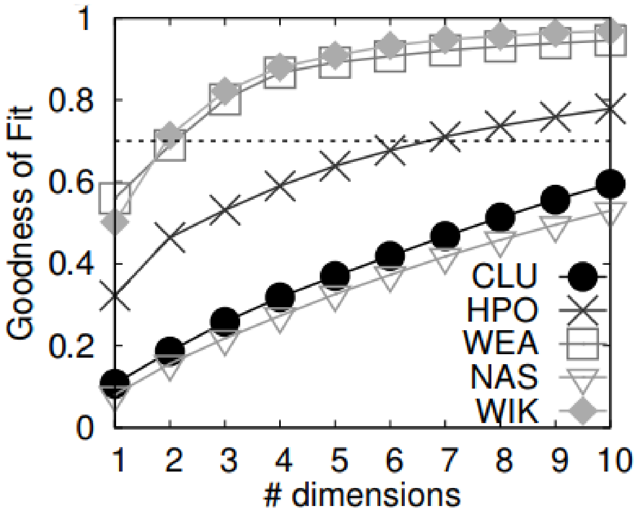
Dataset spaces

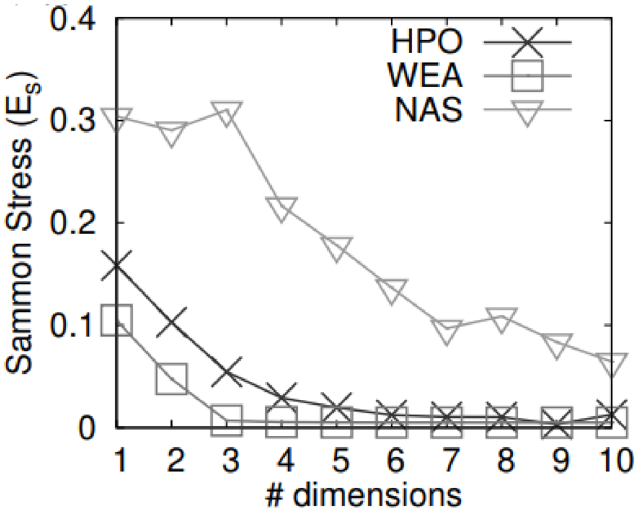




Evaluation

Dataset spaces





Evaluation

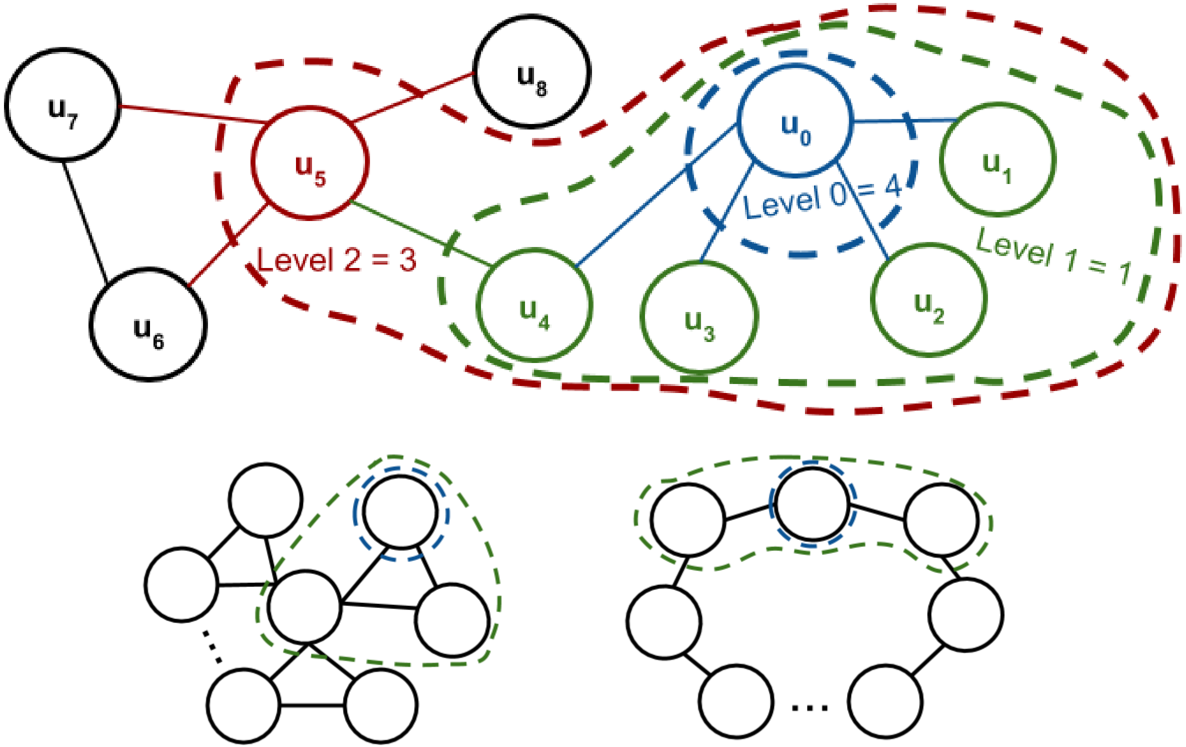
Operator	NRMSE				MdAPE				Speedup (×)				Amortized Speedup (×)			
	4%	8%	16%	32%	4%	8%	16%	32%	4%	8%	16%	32%	4%	8%	16%	32%
CLU-AVG	0.086	0.079	0.073	0.066	0.125	0.114	0.100	0.082	3.21	2.84	2.32	1.69	16.34	9.88	5.52	2.93
CLU-SUM	0.085	0.077	0.070	0.063	0.182	0.158	0.136	0.114	3.21	2.84	2.32	1.69				
CLU-CNT	0.115	0.108	0.104	0.097	0.433	0.401	0.377	0.339	3.21	2.84	2.32	1.69				
CLU-DBS	0.098	0.093	0.088	0.083	0.201	0.191	0.173	0.152	5.69	4.63	3.83	2.19				
CLU-LOF	0.082	0.074	0.070	0.066	0.146	0.136	0.125	0.110	12.13	8.17	4.94	2.76				
CLU-EIG	0.069	0.063	0.058	0.053	0.089	0.079	0.071	0.060	4.27	3.65	2.83	1.95				
HPO-AVG	0.104	0.096	0.088	0.084	0.013	0.012	0.011	0.010	3.93	3.4	2.67	1.87	20.27	11.20	5.91	3.04
HPO-SUM	0.070	0.065	0.056	0.051	0.149	0.135	0.122	0.113	3.93	3.4	2.67	1.87				
HPO-CNT	0.098	0.079	0.069	0.061	0.115	0.104	0.092	0.084	3.93	3.4	2.67	1.87				
HPO-DBS	0.124	0.119	0.114	0.111	0.146	0.141	0.133	0.128	8.30	6.23	4.16	2.50				
HPO-LOF	0.064	0.061	0.055	0.052	0.068	0.063	0.061	0.057	16.64	9.99	5.55	2.94				
HPO-EIG	0.071	0.069	0.067	0.065	0.065	0.063	0.059	0.055	7.33	5.67	3.90	2.72				
HPO-REG	0.073	0.071	0.071	0.069	0.162	0.150	0.134	0.124	11.33	7.80	4.80	2.72				
WEA-AVG	0.089	0.074	0.068	0.059	0.035	0.025	0.020	0.018	2.68	2.42	2.03	1.53	18.72	10.71	5.77	3.00
WEA-SUM	0.075	0.068	0.063	0.057	0.114	0.078	0.059	0.047	2.68	2.42	2.03	1.53				
WEA-CNT	0.119	0.106	0.091	0.080	0.324	0.284	0.244	0.214	2.68	2.42	2.03	1.53				
WEA-DBS	0.182	0.180	0.176	0.171	0.323	0.328	0.303	0.288	6.06	4.88	3.51	2.25				
WEA-LOF	0.126	0.123	0.115	0.110	0.118	0.113	0.107	0.093	16.71	10.02	5.56	2.94				
WEA-EIG	0.035	0.032	0.031	0.029	0.024	0.021	0.019	0.018	5.59	4.57	3.35	2.18				
NAS-HOL	0.093	0.090	0.086	0.084	0.700	0.445	0.333	0.283	0.65	0.63	0.60	0.55	3.45	3.03	2.44	1.75
NAS-ARI	0.095	0.090	0.085	0.084	0.773	0.548	0.341	0.262	2.94	2.63	2.17	1.61				
WIK-HOL	0.018	0.018	0.018	0.018	0.812	0.686	0.582	0.353	0.17	0.16	0.16	0.16	1.42	1.34	1.21	1.01
WIK-ARI	0.019	0.019	0.019	0.019	0.595	0.488	0.324	0.237	1.27	1.20	1.10	0.93				

Graph modeling

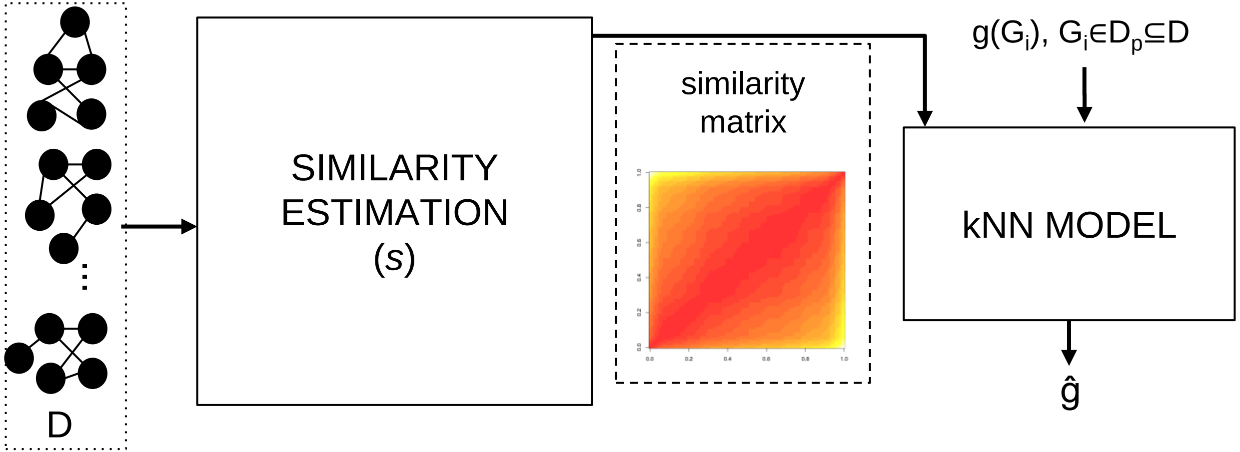
Apply the same idea to different types of data

- Let’s try graphs
  - Similarity Metrics:
    - Degree distribution (in different levels) + Size
    - D-similarity
    - Random Walk Kernel
  - Operators from different classes
    - Distance: {betweenness,edge betweenness,closeness} centrality
    - Spectrum: spectral radius, eigenvector centrality
    - Connectivity: PageRank

Graph modeling - degree distribution similarity



Graph modeling



Graph modeling

Dataset	Metric	MdAPE (%)			nRMSE			Speedup ×			Amortized Speedup ×		
		p=5%	p=10%	p=20%	p=5%	p=10%	p=20%	p=5%	p=10%	p=20%	p=5%	p=10%	p=20%
AS	**sr**	1.3	1.1	0.9	0.05	0.03	0.02	6.4	3.8	3.3	18.0	9.5	4.9
	ec	0.1	0.1	0.0	0.01	0.00	0.00	5.7	4.5	3.1			
	bc	1.4	1.2	1.1	0.04	0.03	0.03	15.7	8.8	4.7			
	ebc	3.1	2.7	2.4	0.04	0.04	0.04	17.3	9.3	4.8			
	cc	0.4	0.4	0.3	0.01	0.01	0.01	14.0	8.2	4.5			
	pr	0.9	0.8	0.7	0.05	0.04	0.03	5.7	4.4	3.1			
TW	**sr**	16.3	15.3	14.7	0.10	0.10	0.10	13.3	8.0	4.4	14.8	8.5	4.6
	ec	8.0	7.7	7.7	0.14	0.14	0.13	13.1	7.9	4.4			
	bc	17.8	17.5	16.8	0.16	0.15	0.14	13.0	7.8	4.4			
	ebc	29.5	29.8	28.6	0.12	0.12	0.12	13.5	8.0	4.4			
	cc	3.3	3.0	2.9	0.10	0.10	0.09	13.0	7.9	4.4			
	pr	9.2	7.7	7.2	0.07	0.06	0.05	13.2	7.9	4.4			
BA	**sr**	3.3	1.8	0.9	0.04	0.03	0.03	5.6	4.4	3.0	16.3	9.0	4.7
	ec	0.4	0.3	0.3	0.01	0.01	0.01	3.7	3.1	2.4			
	bc	10.3	10.1	9.6	0.10	0.05	0.02	12.6	7.7	4.4			
	ebc	10.9	9.3	8.5	0.10	0.09	0.01	13.6	8.1	4.5			
	cc	2.4	2.2	2.1	0.04	0.04	0.03	9.9	6.6	4.0			
	pr	6.7	6.1	5.9	0.06	0.05	0.05	3.6	3.0	2.3			

Conclusions

Modeling operator output

- Many operators, but only ***a few*** data properties
- Dataset spaces do ***make sense***
- ***Accelerate*** data analysis workflows

System is publicly available

- https://github.com/giagiannis/data-profiler