Graph Based Cardinality Estimation from Query Constraints

Group Member: Chun Wang, Wanning He, Changwen Xu

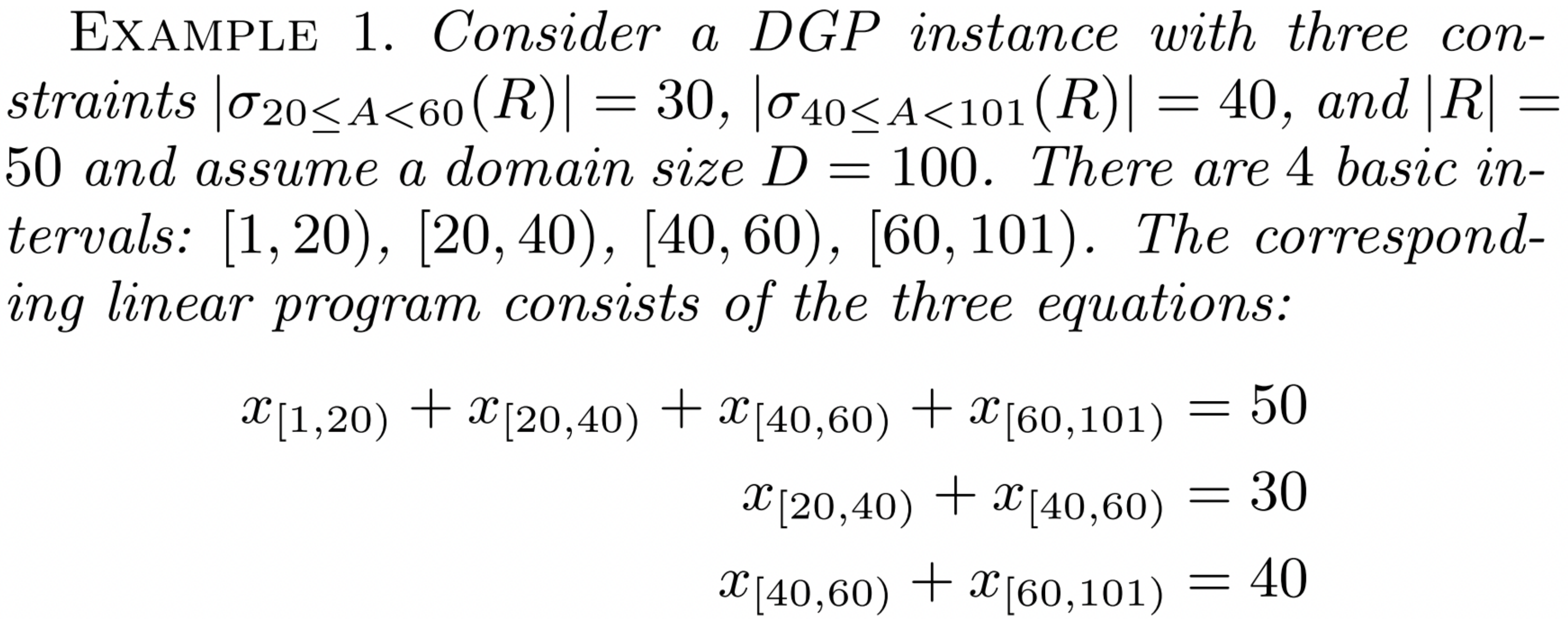
Cardinality Estimation Problem

- Cardinality estimation is a sub-problem of database query optimization
    - Given a database schema and a collection of query and their cardinality constraints $C_1, . . . , C_m$
    - Generate a database instance conforming to the schema that satisfies all the constraints.
    - Given future input queries and return their estimated cardinality.

Cardinality Constraints

- Single Table, Single Attribute (Intervalization)
  - $X_{[vi,vi+1)}$ represents the number of tuples t ∈ Dom(A) in R(A) that belong to the interval $[v_i, v_{i+1})$.

$$\sum_{i=p}^{q-1} x_{[v_i, v_{i+1})} = k_j$$



Example 1. *Consider a DGP instance with three constraints* $|\sigma_{20 \leq A < 60}(R)| = 30$, $|\sigma_{40 \leq A < 101}(R)| = 40$, *and* $|R| = 50$ *and assume a domain size* $D = 100$. *There are* 4 *basic intervals:* $[1, 20)$, $[20, 40)$, $[40, 60)$, $[60, 101)$. *The corresponding linear program consists of the three equations:*

$$\begin{aligned} x_{[1,20)} + x_{[20,40)} + x_{[40,60)} + x_{[60,101)} &= 50 \\ x_{[20,40)} + x_{[40,60)} &= 30 \\ x_{[40,60)} + x_{[60,101)} &= 40 \end{aligned}$$

Existing Works

- Symbolic-CSP-based
  - It starts with a symbolic database. It then translates the input AQPs to constraints over the symbols in the database, and invokes a black-box constraint satisfaction program (CSP) to identify values for symbols that satisfy all the constraints.
  - Limitation:
    - Not guaranteed to produce a single database instance.
    - The times invoking a CSP grows with the size of the generated database.
  - Related work: QAGen [SIGMOD, 2007]、MyBenchmark [VLDB 2010]

Existing Works

- ILP-CSP-based
    - Modeling the CCs as a system of linear equations and solve it using an Integer Linear Problem solver.
    - Limitation
        - The number of variables created can be exponential in the number of attributes.
    - Related work:
        - Data Generation using Declarative Constraints [Sigmod 2011]
        - Synthesizing Linked Data Under Cardinality and Integrity Constraints [Sigmod 2021]

Motivation

- Symbolic-CSP-based methods use greedy algorithm to identify values for symbols that satisfy all the constraints. They us can not effectively generate database satisfying lots of constraints (for workload with large amount of queries).

- The number of variables created by ILP-based methods can be exponential in the number of attributes.

- Therefore, we try to learn the joint distribution of origin database from the workload by Graph Neural Network according to the cardinality of queries in workload.

Machine learning Based Data Generation

- PDF: Probability density function based
  - Symbolic- and ILP- CSP based method
- CDF: Cumulative distribution function based
  - Ours graph method

PDF based

- Auto-regressive (ART) model
  - Product rule factorization can be the representation of a joint distribution, if we have three variable a, b and c, then the joint distribution:

$$P(a,b,c) = P(a) \cdot P(b|a) \cdot P(c|a,b).$$

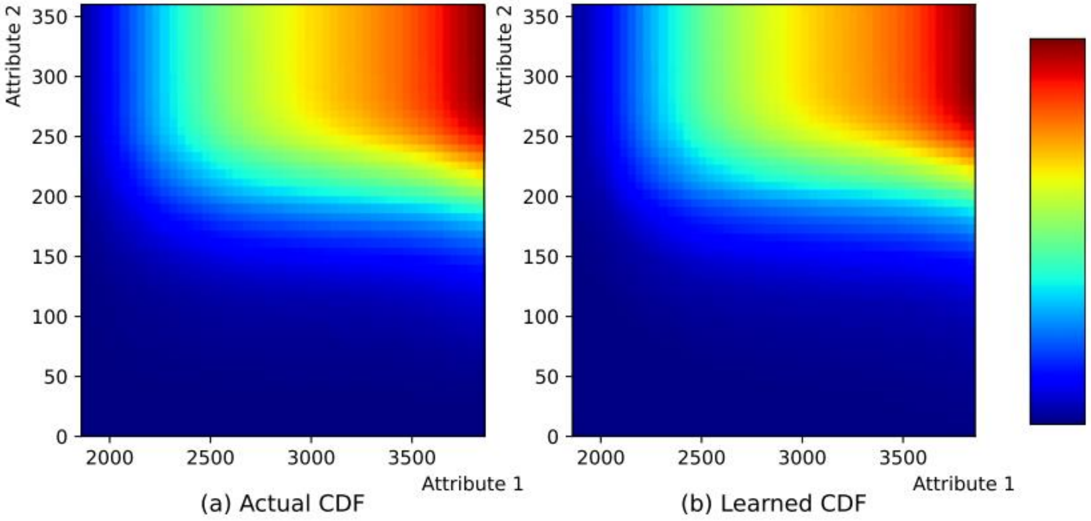
  - There are several AR models representing this rule such as Made and ResMade.
  - Existing works (Naru and Nerocard) propose to train the AR model by scanning the datasets for selectivity estimation, while our problem is to learn distributions purely by workload for data generation.

CDF-based

- Motivation
  - PDF model works well for categorial variable, but may fail in continuous cases.
  - We propose to use GNNs to learn the CDF when the domine size $Dom(Ai)$ is large, leading to a large model and a long training time.
  - The cumulative distribution function (CDF) of a *d*-dimensional real-valued random variable $X = (X_1, X_2, \ldots, X_d)$ is defined as:
  - $F_X(x) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_d \leq x_d)$, where $x = [x_1, x_2, \ldots, x_d]$ is a point of multi-dimensional space *D*.

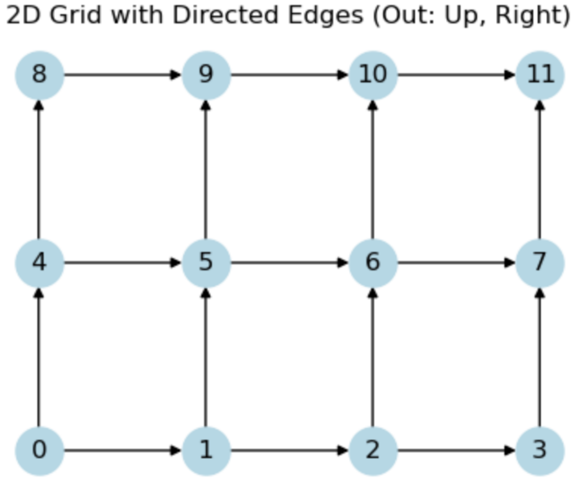
The CDF could be learned in a query-based fashion.

- An observation:
  - The actual joint CDF is monotonically increasing with each attribute
  - Thus we may use this to design our GNN architecture.



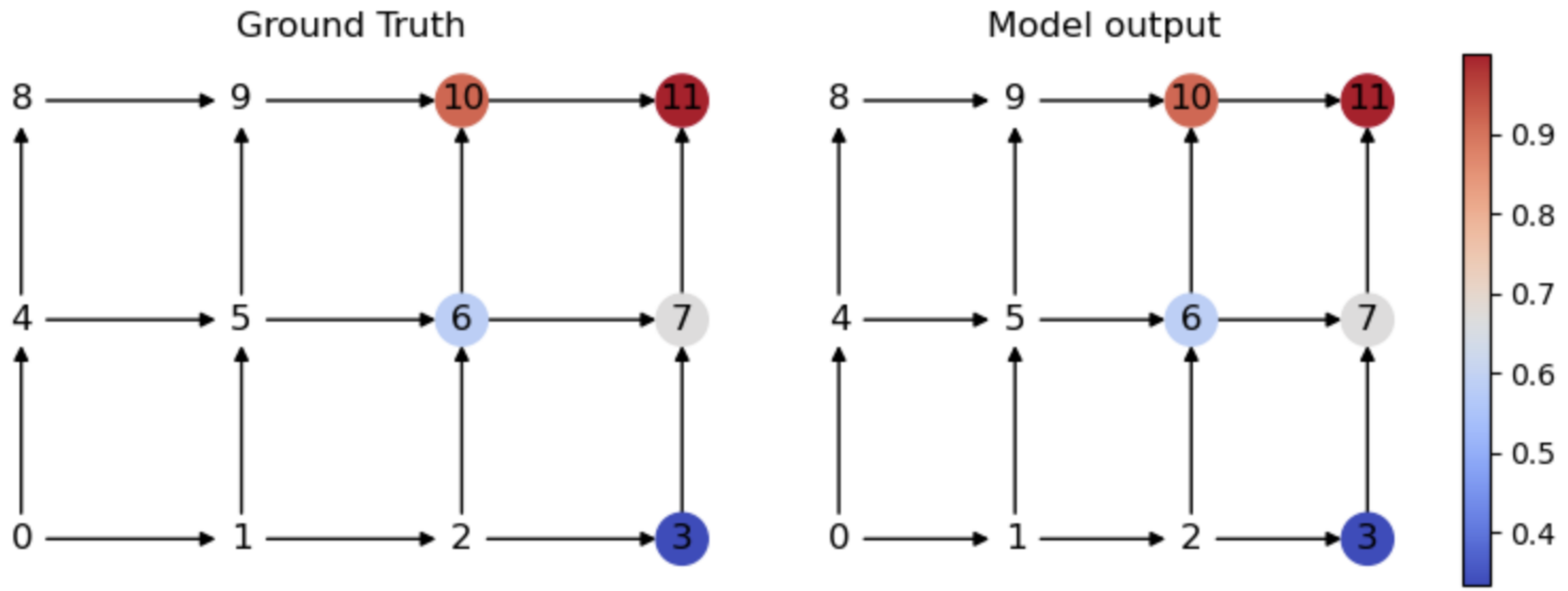
Workflow of Graph based methods

- How to connect and update the graph node?
  - Each node (xi, yi) represent F(xi, yi) = P(X<=xi, Y<=yi)
  - Each node is encoded by index, we use index to take corresponding nodes out of a 1D-array instead of maintain a multi-dimension array.
  - We implement the node in monotonical order and design a directed graph.
  - Each node takes max value (activation) of all its successors.



Workflow of Graph based methods

- How to define the training error without the true data as input?
  - We use the RMSE between true selectivity and predicted selectivity of selected nodes as the training loss.
  - Cross entropy loss or neg-log likelihood is also promising if we treat CDF as a PDF.



Challenges

- CDF to common query selectivity constraint
  - A common query can be represented by the 2-input range [li, ui] and the selectivity ki .
  - We need one step calculation to transform CDF to range selectivity
  - The cumulative probability of point x=[x1, x2, . . . , xd] is actually the selectivity of query with range of [0, x1] × [0, x2] × . . . × [0, xd]

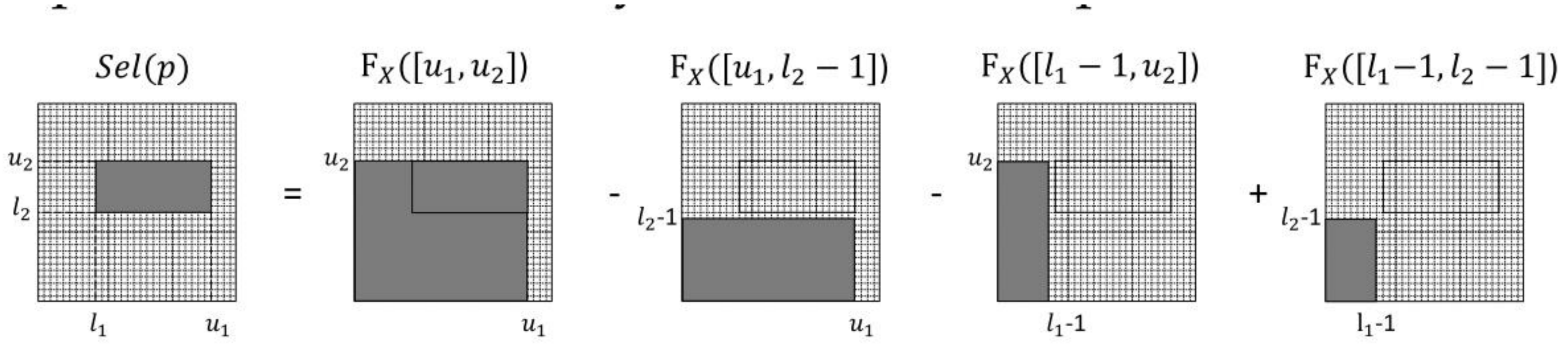


Fig. 1: Transformation from CDF to selectivity

Challenges

- Large-scale Graph
  - How to evaluate and limit the size of the graph? As the dimension increases, the smallest connection structure (Markov blanket) of each node becomes larger. Do it as conditional probability? AR model?
- Multiple tables
  - First learn the outer-join of n tables
  - Design an assignment matching algorithm to split each table.