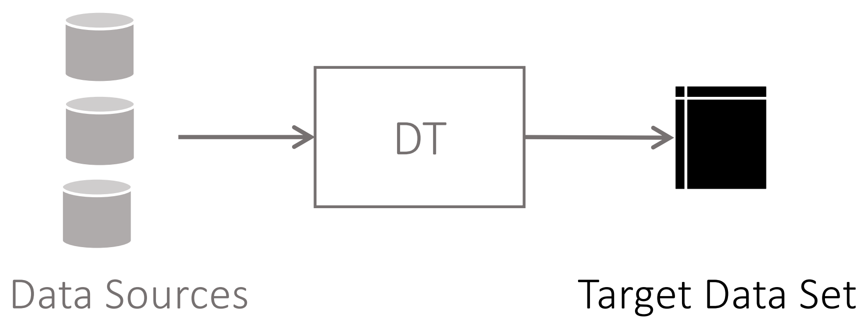
DISTRIBUTION TAILORING

MOTIVATION

- Distribution requirements on data sets
  - reducing model error (for feature slices)
  - showing adequate consideration of minority groups
- Sources of data
  - explicitly collected by the data scientist
  - secondary data, collected for some other purpose
- Can data from multiple sources be put together to build a data set with a desired distribution?
  - Data Distribution Tailoring (DT)

QUERY MODEL

- User's query: target schema and distribution requirements
- Target schema contains some sensitive attributes that identify the groups.
- A distribution requirement specified over some groups
  - Count requirements: group ratio + target size



Schema: movie_title, actor_name, gender, race, …

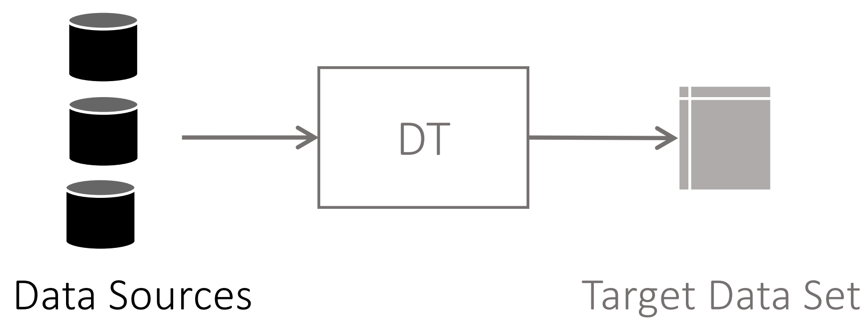
Distbn Requirements:
WM: 1K, NWF: 1K, …

DATA MODEL

- A collection of data sources
- Each source has the same schema as the user's query schema.
  - Each tuple of a source can be associated with a group.
- We assume a tuple-at-a-time access to a source.

Schema: movie_title, actor_name, gender, race, …

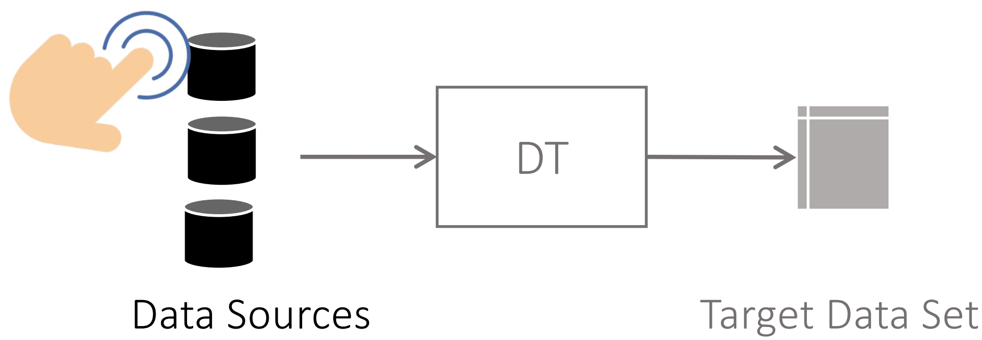
SPJ views over data lakes, web services, data markets, or data brokers



COST MODEL

- Obtaining samples from different data sources is not for free.
- Samples are associated with a cost: monetary, computation, memory or network access.

Cost of sampling k tuples at random



DATA DISTRIBUTION TAILORING (DT)

- Given sources $L = \{D_1,\ldots, D_n\}$ with their costs $\{C_1,\ldots, C_n\}$, and count requirements $\{Q_1, . . . , Q_m\}$ on groups $\{G_1, . . . , G_m\}$, our goal is to query different sources in L, in a sequential manner, in order to collect samples that fulfill the count requirement, while the expected total query cost is minimized.

DT ALGORITHM

Input: data sources $L=\{D_1, . . . , D_n\}$ and $\{C_1, …, C_n\}$
counts $\{Q_1, …, Q_m\}$ over $\{G_1, …, G_m\}$;
Output: O, the target data set

$O \leftarrow \{\}$, cost $\leftarrow$ 0
while(Q_j>0) do
    D_i, C_i $\leftarrow$ select_optimal_source()
    s $\leftarrow$ Query(D)
    j $\leftarrow$ Group(s)
    if(s $\notin$ O AND Q_j>0) then
        add s to O;
        $Q_j \leftarrow Q_j-1$
    cost $\leftarrow$ cost + C_i
return O

VERSIONS OF DT

- Known source distributions
- Unknown source distributions

DT: KNOWN DISTRIBUTIONS

- Notations
  - $Q = \{Q_1, \cdots, Q_m\}$: count requirements on m groups
  - C_i: cost of D_i
  - P_i^j : prob of collecting G_j from D_i
    - N_i: #tuples in data source D_i
    - N_i^j: #tuples in D_i that belong to G_j
  - F(Q): min expected cost of a target with counts Q

- How to compute F(Q)?
  - Think recursively. Consider the probability of obtaining a fresh and useful tuple. Include the case when a tuple is not useful.

KNOWN DT: COST FUNCTION

F(Q): min expected cost of a target with counts Q

$F_j(Q) = F(Q_1,\cdots,Q_j-1,\cdots,Q_m)$

- Take a sample from D_i

Cost of sample

prob. of a seen or useless sample of Qc

$$C_i + \sum_{j=1,Q_j>0}^{m} P_i^j F_j(Q) + (1 - \sum_{j=1,Q_j>0}^{m} P_i^j) F(Q)$$

Exp. cost of the rest of data collection if a fresh sample is obtained G_j from D_i

Exp. cost of the rest of data collection if sample is not fresh or does not help with target

Expected remaining cost

KNOWN DT: COST FUNCTION

- Source selection strategy

$$min_{\forall D_i}(C_i + \sum_{j=1,Q_j>0}^{m} P_i^j F_j(Q) + (1 - \sum_{j=1,Q_j>0}^{m} P_i^j)F(Q)$$

- What kind of assumption do we make here on P_i^j?
- Which algorithmic technique can we use to solve this optimization problem?

A Dynamic Programming Solution

cost groups

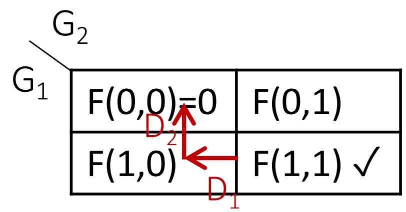
sources

	C_i	G_1	G_2
D_1	2	0.2	0.8
D_2	3	0.4	0.6

cost of obtaining a tuple of G_1 from D_1: 2/0.2=10
cost of obtaining a tuple of G_1 from D_2: 3/0.4=7.5

F(1,0) = min(2/0.2, 3/0.4) = 7.5 ⇐ D_2
F(0,1) = min(2/0.8, 3/0.6) = 2.5 ⇐ D_1

Query: G_1: 1 and G_2: 1
F(1,1): the cost of a target with G_1: 1 and G_2: 1



select D_1: 2 + 0.2 F(0,1) + 0.8 F(1,0)
select D_2: 3 + 0.4 F(0,1) + 0.6 F(1,0)

F(1,1) = min(2 + 0.2 F(0,1) + 0.8 F(1,0),
3 + 0.4 F(0,1) + 0.6 F(1,0)) = 8.4 ⇐ D_1

DP COMPLEXITY

- What is the complexity of this DP algorithm?

DP COMPLEXITY

- Pseudo-polynomial time complexity

$$O(n\, m\, \prod_{i=1}^{m} Q_i)$$

- Not practical for realistic settings

Equi-Cost Binary DT

- Let's consider a common and simple setting
- Groups $\{G_1, G_2\}$ with counts $\{Q_1, Q_2\}$ and all source costs are equal.
- P_i^j : prob of collecting G_j from D_i
- What is the cost of getting a fresh tuple of group G_j from D_i?
- What is the best source for group G_j ?

EQUI-COST BINARY DT

- Groups $\{G_1, G_2\}$ with counts $\{Q_1, Q_2\}$ and all source costs are equal.

- Cost of getting a fresh tuple of G_j from D_i (geometric distribution):
  - $\frac{N_i}{N_i^j - O_i^j}$, O_i^j: #seen tuples of G_j from D_i

- The best source for G_j: $D_{*j} = \underset{\forall D_i}{\operatorname{argm}ax}\left(\frac{N_i^j - O_i^j}{N_i}\right)$

EQUI-COST BINARY DT

- Groups $\{G_1, G_2\}$ with counts $\{Q_1, Q_2\}$ and all source costs are equal.

- Cost of getting a fresh tuple of G_j from D_i (geometric distribution):
  - $\frac{N_i}{N_i^j - O_i^j}$, O_i^j: #seen tuples of G_j from D_i

- The best source for G_j: $D_{*j} = \underset{\forall D_i}{\mathrm{argm}ax}\left(\frac{N_i^j - O_i^j}{N_i}\right)$

OPTIMAL EQUI-COST BINARY

- Which source we should pick in each iteration?

OPTIMAL EQUI-COST BINARY

- Hint: we can find the best source for each group: D_{*1} and D_{*2}

  $D_{*1} = D_i$ and $P_{*1} = \frac{N_i^1 - O_i^1}{N_i}$

  $D_{*2} = D_j$ and $P_{*2} = \frac{N_j^2 - O_j^2}{N_j}$

- Which incurs lower cost?

Optimal Equi-Cost Binary

- Find the best source for each group: D_{*1} and D_{*2}

$$D_{*1} = D_i \text{ and } P_{*1} = \frac{N_i^1 - O_i^1}{N_i}$$

$$D_{*2} = D_j \text{ and } P_{*2} = \frac{N_j^2 - O_j^2}{N_j}$$

Theorem. *Consider the DT problem under the availability of group distributions where there are two groups and the costs for querying data sources are equal. Let* G_1 *be the minority, i.e.* $P_{*1} \leq P_{*2}$. *Selecting* D_{*1} *to query at current iteration is optimal.*

DT for Other Settings

- General DT: non-binary case (m>2) with unequal source costs
  - approximation algorithm with cost upper bound analysis

- Unknown DT
  - An exploration-exploitation solution based on the Multi-Arm Bandit framework

Optimal Equi-Cost Binary

- Proof by contradiction
- Intuition: piggy-backing
  - while sampling from the minority group, we collect items of the majority group.

PROOF SKETCH

- Proof by contradiction
- Let $D_{*1} = D_i$. Suppose A_1 that select D_i is not optimal. Suppose the optimal algorithm A_2 selects $D_{i \neq j}$. We show that the expected cost of A_1 cannot be less than A_2. Let $P' = \frac{N_j^1 - O_j^1}{N_j}$. Note $P' \leq P_{*1}$.

$F_i(Q_1, Q_2) = P_{*1} F(Q_1\text{-}1, Q_2) + (1\text{-}P_{*1}) F(Q_1, Q_2\text{-}1)$

$F_j(Q_1, Q_2) = P' F(Q_1\text{-}1, Q_2) + (1\text{-} P') F(Q_1, Q_2\text{-}1)$

$B = F_j(Q_1, Q_2) - F_i(Q_1, Q_2)$

$= (P_{*1} - P')(F(Q_1, Q_2\text{-}1) - F(Q_1\text{-}1, Q_2))$

PROOF SKETCH

$F(Q_1-1, Q_2) = F(Q_1-1, Q_2-1) + F(0,1)$

$F(Q_1, Q_2-1) = F(Q_1-1, Q_2-1) + F(1,0)$

- Since G1 is the minority, F (0, 1) ≤ F (1, 0). Therefore $\boldsymbol{B \geq 0}$
- Since the expected cost of A_1 cannot be less that of A_2, selecting $D_i = D_{*1}$ to query at iteration i is an optimal solution.

Equi-Cost Binary DT Algorithm

Input: number of items from $Q = \{Q_1, Q_2\}$;

data sources $L=\{D_1, . . . , D_n\}$

Output: O, the target data set

1: $O \leftarrow \{\}$

2: **while**(Q_1>0 AND Q_2>0) **do**

3: D ← source with max ratio of undiscovered G_1

4: D’ ← source with max ratio of undiscovered G_2

5: D’’ ← source (D or D’) with the minority group

6: s ← Query(D’’)

…

GENERAL NON-BINARY DT

- Multiple groups $\{G_1, \ldots, G_m\}$ with count requirements $\{Q_1, \ldots, Q_m\}$ and source costs are not equal.

- Brainstorming for an algorithm for the general non-binary DT.

General Non-Binary DT

- Multiple groups $\{G_1, \ldots, G_m\}$ with count requirements $\{Q_1, \ldots, Q_m\}$ and source costs are not equal.

- For group G_j, what is the most cost-effective data source?
- How can we use the cost-effective data sources to fulfill the count requirements?

GENERAL NON-BINARY DT

- For group G_j, the most cost-effective data source is

$$D_{*j} = \underset{\forall D_i}{\operatorname{argmax}} \frac{N_i^j}{N_i . C_i}$$

General DT Algorithm

- Select the most cost-effective source for G_j (namely D_{*j}) and commit to it.
- Query the data source D_{*j} for group G_j
  - Maintain the tuples of other groups *(piggybacking)*
- Repeat until the target specified by the count description $[Q_1, . . . , Q_m]$ is collected.

PROJECT 2: VARORIATIONS OF DT

- Given sources $L = \{D_1,\ldots, D_n\}$ with their costs $\{C_1,\ldots, C_n\}$, and count requirements $\{Q_1, \ldots, Q_m\}$ on groups $\{G_1, \ldots, G_m\}$, our goal is to query different sources in L, in a sequential manner, in order to collect samples that fulfill the count requirement, while the expected total query cost is minimized.
- Generalize the problem to
  - fixed > 1 number of samples at each iteration
  - arbitrary number of samples at each iteration
  - count requirements on multiple groups (e.g. 100 of gender=F and 100 of gender=M as well as 100 of race=W and 100 of race=NW)
  - overlapping sources