CS 784: Foundations of Data Management **Spring 2017**

Lecture 6: Size Bounds for Joins

Instructor: Paris Koutris

As we discussed in previous lectures, the output size of a join query often dominates the running time, since the algorithm has to enumerate all the output tuples. Thus, being able to compute the output size, or even provide a good upper bound on the output size becomes an important task. In this lecture, we discuss the following two question:

1. Given a conjunctive query q, where each relation R_j has size N_j, what is the largest possible output?
2. Can we construct an algorithm that runs in time at most the largest output?

We start with two examples.

Example 6.1. *Consider the join query* $q(x,y,z) = R_1(x,y), R_2(y,z)$ *where the sizes of the relations are* N_1 *and* N_2 *respectively. The largest possible output is* $N_1 \cdot N_2$*, which occurs when the join behaves like a cartesian product (i.e. there is a single value of the y variable). One can also observe that we can construct a trivial algorithm that runs in time* $N_1 \cdot N_2$ *by considering all possible pairs of tuples and checking whether they join or not.*

Example 6.2. *Consider the triangle query* $\Delta(x,y,z) = R(x,y), S(y,z), T(z,x)$*, where relations have sizes* N_R, N_S, N_T*. A first straightforward bound is* $N_R \cdot N_S \cdot N_T$*. We can get a better bound by noticing that the join of any two relations is an upper bound on the total size, so we get an improved bound of* $\min\{N_R \cdot N_S, N_R \cdot N_T, N_T \cdot N_S\}$.

Can we do any better? We will see that another upper bound on the size of the query is $\sqrt{N_R \cdot N_S \cdot N_T}$*. Notice that, depending on the relation between* N_R, N_S, N_T*, this can be a better or worse bound than the above three quantities.*

6.1 The AGM Bound

We start by introducing some notation.

Definition 6.3 (Fractional Edge Cover)**.** *The* fractional edge cover *of a conjunctive query* q *is a vector* $\mathbf{u}$*, which assigns a weight* u_j *to relation* R_j*, such that for every variable* $x \in vars(q)$*, we have that* $\sum_{j:x \in vars(R_j)} u_j \geq 1$.

We say *fractional* edge cover to distinguish from the (integral) edge cover, which assigns to each relation a weight of 0 or 1. The value of the *minimum fractional edge cover* of a conjunctive query q is denoted by $\rho^*(q)$.

Example 6.4. *Consider again the triangle query Δ. A possible fractional edge cover is $u_R = u_S = 1$, and $u_T = 0$. In this case, the sum of the weights is 2. Another fractional edge cover is $u_R = u_S = u_T = 1/2$, which has a smaller sum $3/2$.*

The AGM inequality, first proved in [AGM08], bounds the output size of a join query w/o projections using any fractional edge cover of the query.

Theorem 6.5. *Let q be a full conjunctive query that takes as an input relations S_j with size at most N_j. For every fractional edge cover $\mathbf{u}$ of q, the output size is bounded as follows:*

$$|q(I)| \leq \prod_{j=1}^{\ell} N_j^{u_j}$$

Notice that in the case we have the same upper bound N on the sizes of the relations, i.e. $N_j = N$, we have that $|q(I)| \leq \min_{\mathbf{u}} N^{\sum_j u_j} = N^{\rho^*(q)}$. In other words, the best bound is achieved by the minimum fractional edge cover ρ^*.

Example 6.6. *For the triangle query, the fractional edge cover is $u_R = u_S = u_S = 1/2$ gives the $\sqrt{N_R \cdot N_S \cdot N_T}$ bound. The fractional edge covers $(u_R, u_S, u_T) = (1,1,0), (1,0,1), (0,1,1)$ give the $N_R \cdot N_S$, $N_R \cdot N_T$ and $N_S \cdot N_T$ upper bounds respectively.*

Example 6.7. *Consider the Loomis Whitney join LW_k, where each relation has size at most N:*

$$LW_k = R_1(x_2, \ldots, x_k), R_2(x_1, x_3, \ldots, x_k), \ldots, R_k(x_1, \ldots, x_{k-1})$$

The smallest fractional edge cover assigns an equal weight of $1/(k-1)$ to each R_j (observe that each variable belongs to exactly $k-1$ atoms). The bound we get then is

$$|LW_k(I)| \leq N^{\sum_{j=1}^{k} u_k} = N^{k/(k-1)}.$$

The AGM bound gives us an infinite number of upper bounds on the output size. Given the cardinalities of each relation, how can we find the best (minimum) possible bound? In the case of equal cardinalities N it suffices to find ρ^*, but in the general case we can achieve this by minimizing the quantity $\prod_{j=1}^{\ell} N_j^{u_j}$ by solving the following linear program:

$$\begin{aligned} \min \quad & \sum_j \log_2(N_j) \cdot u_j \\ \text{s.t.} \forall x \in vars(q) : & \sum_{j : x \in vars(R_j)} u_j \geq 1 \\ \forall R_j : & \, u_j \geq 0 \end{aligned}$$

The AGM bound is *tight*; in other words, we can always find a database instance I, such that $|q(I)|$ is equal to the the worst-case upper bound.

6.2 Worst-Case Optimal Joins

All of the join processing algorithms we have seen so far (e.g. for acyclic queries, or queries of bounded query width) have used Join-Project query plans. Consider the triangle query with relations of equal size N: the worst-case output size is $N^{3/2}$. Consider the three standard query plans to compute this query: $(R \bowtie S) \bowtie T$, $(R \bowtie T) \bowtie S$ and $(T \bowtie S) \bowtie R$. We can construct an instance such that any of these plans needs time $\Omega(N^2)$ to run (since the intermediate size of the join will be that large). In fact, even if we add projections to the plan, we can show that the running time will always be $\Omega(N^2)$ in the worst case. The question now is: can we design an algorithm that *always* runs in time linear w.r.t. the worst-case output, which in our case is $O(N^{3/2})$?

The answer to this question is yes; there exists a worst-case optimal algorithm that matches the worst-case output in running time [NRR13,V14].

References

[Alice] S. ABITEBOUL, R. HULL and V. VIANU, "Foundations of Databases."

[AGM08] A. ATSERIAS, M. GROHE and D. MARX, "Size bounds and query plans for relational joins," *FOCS 2008*.

[NRR13] H. NGO, C. RE and A. RUDRA, "Skew Strikes Back: New Developments in the Theory of Join Algorithms," *SIGMOD Record, 2013*.

[V14] T. VELDHUIZEN, "Leapfrog Triejoin: a worst-case optimal join algorithm," *ICDT, 2014*.