Access path selection in a relational database management system [**?**].

1 Abstract / Introduction

1. What opportunities/changes that make this work useful and timely? 2. Why existing approaches fail to make use of these opportunities? 3. How do you propose to do better? 4. Why this problem is relevant to the course? (1-2 sentences each)

SQL is a declarative alternative to imperative data manipulation languages used by IMS and proposed by CODASYL. SQL translates into relational algebra (*RA*), which is built on top of a theoretically sound relational data model. SQL is good because it makes the programmer's life easier by providing data independence so that changes in how the data is physically stored does not affect the SQL queries that developers write to access and manipulate their data.

This development is important because developers currently spend most of their time writing and rewriting their applications whenever there is a change to the schema of the data or how it is stored. This means developers don't have time to build new features and improve their applications. SQL can help address this issue, but current SQL execution engines are unacceptably slow and nowhere near the performance of hand-tuned IMS/CODASYL queries.

SQL can translate into many different, but semantically equivalent, relational algebra statements, and each statement can be executed by many possible *query plans*, it is desirable to pick the plan that is fastest to run. This project proposes a way to estimate the cost of executing a query plan, and automatically search the space of all valid query plans for a SQL query to find one that is fast no execute.

This problem is relevant to the course because data visualization systems are often built on top of data management systems, so making the data management system faster would thus make the data visualization faster.

2 One Sentence Summary

> *Describe your project in one sentence, in other words, your hypothesis.*

We will devise a cost model to give each query plan a score, and use existing equivalence rules to define and search the space of alternative plans to find one with low cost. Our hypothesis is that this will be competitive with hand-coded execution plans.

3 Audience and Needs

> *Who are the audiences for this project? How does it meet their needs? What happens if their needs remain unmet?*

This project will impact the academic community as well as every application developer. The community benefits because it will validate the hypothesis that declarative lanugages such as SQL can run quickly, and open up a new field of research in data management.

Application developers benefit because they will not need to write imperative data manipulation code, worry about how to hand optimize the code to be fast, and can focus on building application features.

4 Approach

> *What is your approach? Why do you think it's a good approach and will be successful?*

The problem of entity resolution bears some resemblance to that of facial recognition. In both cases, we are less concerned about what a record is or who a face belongs to and more interested in whether two records or two faces are the same. Recent innovations in facial recognition technology

uses neural networks trained on a triplet loss function. In a triplet loss network for facial recognition, rather than predicting the class of an input image X, the network outputs a multi-dimensional vector embedding $f(X)$ of that image with the constraint that the Euclidean distance between two embeddings of images of two different people is significantly large. That is, let $\mathbb{I}(X)$ be the identity of the person whose face appears in X. If $\mathbb{I}(X) = \mathbb{I}(Y)$ and $\mathbb{I}(X) \neq \mathbb{I}(Z)$, then $\|f(X) - f(Z)\| \gg \|f(X) - f(Y)\|$.

A training instance for such a network consists of a triplet of three images (A, P, N), an anchor, positive, and negative image, respectively, that satisfies $\mathbb{I}(A) = \mathbb{I}(P)$ and $\mathbb{I}(A) \neq \mathbb{I}(N)$. Given said triplet, the loss that they incur on a network is $\mathcal{L}(A, D, P) = \max(0, \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha)$, where α is a threshold hyperparameter defined prior to training.

The above loss function has proven successful for face recognition, and we believe it can be applied similarly to entity resolution. To generate triplets, we will use a *discriminator model* to predict whether pairs of records refer to the same entity. The mispredictions of the discriminator model will either consist of false positives which can be used as the A, N records within a triplet or false negatives which can be used as the A, P records within a triplet. Ideally, we would like to show that given an initial discriminator model, using its mispredictions as training examples for a triplet loss network results in a neural model that outperforms the discriminator.

5 (Best Case) Impact

In the best-case scenario, what would be the impact statement (ideal outcome and conclusion) for this project?

We can show that the optimizer picks reasonable plans that are comparable to hand-optimized plans selected by expert developers.

6 Milestones

1. Identify datasets tested in previous methodologies that are computational feasible to train on and that we can use as benchmarks.

2. Develop an adversarial procedure for generating triplets. This will entail developing a tree-based ensemble and corresponding featurization procedures to generate false positives and false negatives. The accuracy of this procedure may also serve as a benchmark.

3. Implement a neural network based on the triplet loss function and compare against benchmark used to generate "hard" negatives.

4. Develop data augmentation procedures to generate positive pairs and improve the performance of the triplet loss network.

5. Run experiments on various datasets and compare results with current widespread methods.

7 Obstacles

*Major obstacles are situations where we would consider **killing** the project. Minor obstacles are situations that would delay the project or increase the overall cost in energy, time, people, and money.*

7.1 Major obstacles

- If we cannot show that a triplet loss network can at the very least improve on the performance of the initial model used to generate false positives and negatives, it is highly unlikely that said network can compare favorably to the state of the art.

7.2 Minor obstacles

- The original purpose of the triplet loss function was to enable facial recognition systems to compare new images of faces against a database of faces the system has already trained on. In other words, a facial recognition system is responsible for determining whether a new face is the same as one it has been trained to recognize, but it is not responsible for determining whether two new faces are the same. It is possible the triplet loss function is effective for only the former and not the latter, in which case a triplet loss network would require significantly more training data and computational resources to be effective.

- A typical dataset or even pair of datasets with duplicate entities largely consists of negative pairings. While an initial model such as a random forest can be used to pick out negative pairings most effective for a tripet loss network, we may be constrained by the number of positive

pairings, in which case we will need to devise data augmentation methods to generate additional positive pairings to increase the number of training triplets.

- We may discover that the neural network architecture needed to effectively embed records as vectors is prohibitively large for the computational resources available. If so, we should consider training on datasets with fewer attributes or using simpler benchmark models for initially detecting false positives and negatives.

8 Additional Resources

What additional resources do you need to complete this project?

- Some computational time to run our optimizer algorithm to generate some query plans.
- Access to a machine where we can install and run experiments using our current database prototype.

9 Literature Review

List 5 major publications that are most relevant to this project, and how they are related.

- *Background for the project:* This work builds on prior work on relational algebra and the relational model [**?**], and on new relational database systems [**?**, **?**]
- *Work the project relies and builds on:* Some preliminary work has suggested a language for specifying query plans [**?**] that we could borrow from. Also, Recent work [**?**] on different ways to store and index data lend credence to the need for different cost models for access data in relations.
- *Direct competitors:* We could not find existing works on alternative techniques to automatically optimize query plans.
- *Alternatives to achieve the broader goal:* As stated above, IMS and CODASYL [**?**] are the main alternative data management systems, but they do not have any automated query optimization.

10 Define Success

When / How do you know if you have succeeded in this project? In other words, what is the minimum finding that would make this project a success and publishable?

Simply developing a set of cost models and search heuristics for query plans should be publishable, because an automated optimizer of any sort does not yet exist.