Type Hierarchy Inference

First we need to form a concrete problem description

Fabian Klopfer
Department of Computer Science
University of Konstanz

Abstract

Contents

First we need to form a concrete problem description

Given:

Set of Relations $R_1, \ldots, R_n,\ n \in \mathbb{N}$ with attributes of variable numbers, including primary keys in the case of relational DBs and triplets in the form id, key, value in the case of triplet stores (RDF) and graph DBs like Neo4J (Id + key value triplets for nodes, properties; quadlets for edges with id, type, key, value).
So the complete structure (w.r.t. the available information of the DB) of the type is gained by querying all attributes of a certain id over all available relations or all triplets of a certain id. This means all available information on an entity consists of the possibly available fields and their values.
More formal, the type of an entity is defined by all it's attributes contained in the relations:

$$\text{Type of entity with } id_x \equiv \forall r \in \text{ Relations } R_1, \ldots, R_n : \{\text{attribute } r.a | r.id = id_x\}$$

Relational data bases mostly already model such types or attributes describing entities of one type, whereas non-relational data stores rely on labels which may be arbitary.

Problems: Query optimization, Data set unification & Type hierarchy construction
The last one shall be some kind of "unsupervised learning" without doing it in NNs but rather outside as an analysis of what the NNs classified for simple properties to construct higher order traits, types and kinds (e.g. from color and shape of a subpart of the body traits like has face instead of directly detecting faces (using feature vector output vs using (one-hot) category output

- Given a data set with non- or poorly specified types of entities, infer the type from the structure of it's properties and label each record accordingly
- Given mutliple data sets containing information on the same type of entities with primary keys that are matching, construct a common type containing all information and find sub types and relevant specialized sub sets
- Given perceived simple properties of an object, e.g. shape, color, some contextual information, infer a type for classes of objects

Examplatory Applications:

-
  - Given the property graph of data set D, find all entities of the same type and convert it to a relational scheme
    Given a graph of cell properties, build a data base schema to store the data in a relational database (for unification purposes). Given a data set in a certain schema, generate labels and properties to illustrate them in a graph data base.
  - Given the property graph of data set D, infer a type hierarchy to do type checking e.g. the database contains entities which specify cell types in animals and some features of the animal containing the cell, where which cell is embodied. etc. Some animal nodes specify fur color, some specify feather color, some specify number of eyes and so on.
    Select all hair cells of mamals from my graph with more than 2 legs.
    Query without type information one must specify exactly which animal types are

necessary e.g. exclude the hairs of spiders or find the common subset of properties by hand. With type information one can structure queries closer to the mental model like the query mentioned above. With dependent type information it's even possible to do querries in subcategories e.g. show me all hair cells of dogs of breed X instead of having to specify how a dog looks and what specification the breed implies.
If one wants to check queries before submitting them, correctness checks are hardly feasable without catagorization of hat is to be inserted. E.g. if one wants to build a compiler for the cypher query language, type information is essential for propper checking.

- Given two graph data bases on the topic meteorology containing similar properties for each entities, merge the two data sets to gain a more fine grained resolution and additional information if available. Mark incomplete records after the merge or discard duplicate information automatically. Emit on that basis specialized data sets with subsets of properties that are important for a special case, e.g. air preasure, humidity, other indicators for natural catastrophs from standard meteorological data sets for day-to-day prediction

- (Basically irrelevant here, sorry couldnt resist)
  Given a multi-task classifier for semantic segemntation, color, shape, relative orientation
  I. regarding objects: identify sub-parts, identify compositions/objects
  II. regarding the background, assign properties and values to cluster property and value based contexts. E.g. in the house *Rightarrow* property sun angle not present, e.g. in kitchen nested object with color and shape x,y (e.g. fridge, oven) are present
  III. build context like SLAM binding the constructed object and scene hierarchy
  IV. use context to build dependent type hierarchy
  V. include external knowledge on the dependent type hierarchy by adding properties and removing iff below p ¡ 0.05 (or sth.)
  VI. learn to categorize more significant features by adding new structures to NN based on knowledge

Chapter 1

Introduction

1.1 Preface

1.1.1 Motivation & Objective

1.1.2 Contributions

1.1.3 Statement of Originality

1.2 Related Work

1.2.1 Type Theory

1.2.2 Clustering

1.2.3 Functional Dependency Extraction

Chapter 2

Proposed Method

2.1 A Word on the Input & Output Format

2.2 Deriving the Hierarchy: Property-based Clustering

2.3 Refining the Hierarchy: Value-based Sub-Clustering

2.4 Complexity Analysis

2.4.1 Computation

2.4.2 Memory

2.4.3 I/O

Chapter 3

Discussion

3.1 Benchmarks & Results

3.2 Applications

3.3 Summary

3.4 Future Work

Chapter 4

Milestones

TO BE REMOVED WHEN FIXED;

Milestone 1: Fix project including used algos, adjustions to be made, create repo $\rightarrow$ 14.01
Milestone 2: Preprocessing, distance function, chameleon adjustions $\rightarrow$ 01.03
Milestone 3: Integration of CFDs and Value-based clustering, usable outputs for Leo $\rightarrow$ 01.04

OK? Start Thesis and refinements : Do error analysis & Rework methodology;