1 Introduction

1.1 Contenido:

Typically, machine learning algorithms take as input a set of objects, each described by a vector of numerical or categorical attributes, and produce (learn) a mapping from the input to the output predictions: a class label, a regression score, an associated cluster, or a latent representation, among others. In relational learning, relationships between objects are also taken into account during the learning process, and data is represented as a graph composed of nodes (entities) and links (relationships), both with possible associated properties.

The fact that relational learning methods can learn from the connections between data makes them very powerful in different domains [1-4]. Learning to classify profiles in social networks based on their relationships with other objects [5, 6], characterising proteins based on functional connections that arise in organisms [7], and identifying molecules or molecular fragments with the potential to produce toxic effects [8] are some prominent examples of relational machine learning applications.

There are two basic approaches to relational learning, the latent feature or connectionist approach and the graph pattern-based approach or symbolic approach. [9]. The connectionist approach has proven its effectiveness in many different tasks [10-15]. In comparison, the pattern-based approach has been less successful. Two of the most important reasons for this fact are the computational complexity arising from relational queries and the lack of robust and general frameworks to serve as a basis for this kind of symbolic relational learning methods. On the one hand, most of the existing relational query systems are based on graph isomorphisms and their computational complexity is NP-complete, which affects the efficiency of learning methods using them [16]. On the other hand, most existing query systems do not allow atomic operations to expand queries in a partitioned manner, preventing learning systems from efficiently searching the query space [17].

The novel graph query framework presented in this paper attempts to solve these two fundamental problems. The goal is to obtain a query system that allows graph pattern matching with controlled complexity and provides stepwise pattern expansion using well-defined operations. A framework that satisfies these requirements is suitable for use in relational machine learning techniques because, combined with appropriate exploration techniques, it allows the automatic extraction of characteristic relational patterns from data.

Computational capacity needed to assess the performance of graph query methods is significant. Our study centres on formalising an efficient graph query system and defining a set of operations to refine queries. However, it does not conduct an extensive analysis of performance or efficiency in comparison to other methods. The primary result of our study is a mathematical formalisation for a graph query system that enables: The graph query system must fulfill three characteristics: (1) conducting atomic operations (refinements) to expand queries in a partitioned manner, (2) assessing any substructure in a graph (beyond isolated nodes or complete graphs), and (3) evaluating cyclic patterns in polynomial time. To the best of our knowledge, no other approach meets these requirements.

The paper is structured as follows. Section 1 provides an overview of related research. Section 2 introduces a novel graph query framework, outlining its main definitions and properties that guarantee its utility. Representative query examples and an analysis of the computational complexity arising from the model are also presented. Section 3 describes the implementation of the framework to perform relational machine learning. Finally, Section 4 presents the conclusions that can be drawn from this investigation and identifies potential avenues for future research.

1.2 Resumen:

The introduction of the research paper discusses how machine learning algorithms typically process data by taking a set of objects and their attributes as input, and producing predictions. However, in relational learning, relationships between these objects are also considered, with data represented as graphs composed of nodes and links. The paper highlights two basic approaches to relational learning: the latent feature or connectionist approach and the graph pattern-based approach or symbolic approach, with the former being more effective than the latter due to its ability to handle connections efficiently.

The paper introduces a novel graph query framework that attempts to solve the computational complexity issues associated with existing relational query systems, which are based on graph isomorphisms and have NP-complete complexity. The proposed framework allows for controlled complexity in graph pattern matching and well-defined operations for stepwise pattern expansion. The primary goal of this study is to provide a mathematical formalization of the graph query system that meets these requirements, with no other approach meeting them to the best of their knowledge.

The paper's structure includes an overview of related research, an introduction of the novel graph query framework, a description of its implementation for relational machine learning, and conclusions drawn from the investigation along with potential avenues for future research.

1.3 Evaluación:

The section provides a clear overview of the research topic, relational learning, and its significance. The introduction highlights the potential benefits of using graph-based methods for machine learning tasks. However, there are some areas where improvements can be made:

Motivation:

* Clarity: The motivation section could benefit from more specific examples to illustrate the importance of relational learning in real-world applications. For instance, the section could mention how relational learning has been used in social network analysis or recommendation systems.
* Novelty:

Originality: The introduction does not explicitly compare the proposed approach with existing methods, which could be done to emphasize its novelty. Additionally, the section could benefit from more details on what makes the proposed framework unique and different from other graph-based learning methods.

Clarity:

* Comprehension: Some of the sentences in the introduction are quite long and complex, making them difficult to understand. For example, the sentence "Learning to classify profiles in social networks based on their relationships with other objects [5, 6], characterising proteins based on functional connections that arise in organisms [7], and identifying molecules or molecular fragments with the potential to produce toxic effects [8] are some prominent examples of relational machine learning applications." could be broken down into simpler sentences for better comprehension.
* Grammar and Style: The section contains a few grammatical errors, such as missing articles and incorrect verb tenses. Additionally, some of the sentences could be rephrased for better style and readability.

Typos and Errors:

* Accuracy: There is one typo in the section: "The primary result of our study is a mathematical formalisation for a graph query system that enables." should be "The primary result of our study is a mathematical formalization for a graph query system that enables."

Overall, the introduction provides a good overview of the research topic and its significance. However, there are some areas where improvements could be made to enhance clarity, novelty, and accuracy.

2 Related work

2.1 Contenido:

A common approach to executing relational queries entails developing patterns in an abstract representation of data and searching for their occurrences in actual datasets\nocite{phdthesis,Barcelo,gupta2015neo4j,Segaran:2009:PSW:1696488,graphlog,van2016pgql} This working method falls under the scope of Graph Pattern Matching, an area of study that has been actively researched for more than three decades. Depending on various aspects to consider, there are customary distinctions in pattern matching methods. (a) Structural, semantic, exact, inexact, optimal, and approximate are distinctions that can be made in matching relations between patterns and subgraphs [1]. Additionally, graph pattern matching can be based on isomorphisms, graph simulation, and bounded simulation, among other methods [2-4]. While systems for querying based on graph isomorphism present NP complexity, those based on simulations present polynomial complexity [5, 6]. However, both types are based on relations between the set of elements in the query and the set of elements in the graph data, which prevents the evaluation of the non-existence of elements. Our proposal is within the scope of semantic, exact, and optimal graph pattern matching implemented with an approach similar to simulations.

As stated above, there are two fundamentally different types of relational learning models [7]. The first type, known as 'the latent feature approach', is founded upon latent feature learning, for example, tensor factorization and neural models, and generally performs well when handling uncertainty via probabilistic approximation [8-10]. The second approach, known as the graph-pattern based approach, automatically extracts relational patterns, also called observable graph patterns, from data [11, 12]. Since this work pertains to the second approach, our focus in the subsequent discussion will be on the review of relational learning techniques that utilise the graph-pattern based approach and the query systems upon which they rely.

Most of the pattern-based relational learning methods are derived from Inductive Logic Programming (ILP) [13]. ILP does not inherently offer relational classifiers, though it does permit the automatic creation of logical decision trees capable of managing relational predicates, provided that data relationships have been properly transformed into logical predicates. Binary decision trees are logical decision trees, in which all tests in internal nodes are expressed as conjunctions of literals of a prefixed first-order language. TILDE (Top-down Induction of Logical Decision Trees) is one of the representative algorithms that can learn this type of decision tree from a given set of examples [14]. TILDE provides a framework for generating logical decision trees that can be further adapted for relational decision trees. Nevertheless, it does not cater to relational learning and therefore fails to offer certain operations for refining relational queries. We shall refer to atomic operations as those that bring about minor structural modifications to the query (typically the addition or deletion of a node or an edge, or some of their characteristics).

Multi-relational decision tree learning (MRDTL [11]) is a learning algorithm for relationships and is supported by Selection Graphs [15], a graph representation of SQL queries that selects records from a relational database based on certain constraints. Selection graphs enable atomic operations to enhance queries, but they lack the ability to distinguish between query elements that constitute the query result and those that relate to objects that should or should not be linked to the query result. Consequently, queries performed using selection graphs yield records that satisfy the given selection graph conditions, but cannot identify subgraphs. The refinement operations presented on the selection graphs are: adding positive conditions, adding negative conditions, adding present edges and opening nodes, adding absent edges and closing nodes. This set of operations does not allow for the construction of cyclic patterns.

Another noteworthy pattern-based method for relational learning is Graph-Based Induction of Decision Trees (DT-GBI [16]), which is a decision tree construction algorithm for learning graph classifiers using graph-based induction (GBI), a data mining technique for extracting network motifs from labelled graphs by connecting pairs of nodes. In DT-GBI, the attributes (referred to as patterns or substructures) are generated during the execution of the algorithm [12].

As we have seen, some pattern-based approaches are able to learn to classify complete graphs, and some others construct node classifiers; our proposal supports learning from general subgraphs as base cases. Moreover, our technique can execute cyclic queries, hence allowing for extraction of cyclic patterns from data during the learning process.

2.2 Resumen:

The Related work section in the paper titled "Logical-Mathematical Foundations of a Graph Query Framework for Relational Learning" discusses different approaches to executing relational queries, particularly focusing on graph pattern matching. The study highlights various distinctions in pattern matching methods such as structural, semantic, exact, and approximate matchings. It also mentions that graph pattern matching can be based on isomorphisms, graph simulation, and bounded simulation. The paper further distinguishes between two relational learning models: the latent feature approach and the graph-pattern based approach.

The paper then delves into some of the existing methods for pattern-based relational learning, such as Inductive Logic Programming (ILP), Multi-relational decision tree learning (MRDTL), and Graph-Based Induction of Decision Trees (DT-GBI). ILP is noted to not inherently provide relational classifiers but can create logical decision trees. MRDTL utilizes selection graphs for query enhancement, while DT-GBI generates attributes during the execution of the algorithm.

Finally, the paper highlights that their proposed method supports learning from general subgraphs as base cases and can execute cyclic queries, allowing for extraction of cyclic patterns from data during the learning process.

2.3 Evaluación:

1. Motivation:

The section provides a clear explanation of the significance and relevance of the study, highlighting the importance of relational learning in graph-based data and the limitations of existing approaches. The text also justifies the need for a new approach that can handle cyclic queries and extract cyclic patterns from data during the learning process.

YES

2. Novelty:

The section effectively describes the proposed approach's novelty and originality, particularly in its ability to execute cyclic queries and extract cyclic patterns from data. The text also highlights the differences between our proposal and existing approaches, such as those based on selection graphs that cannot handle cyclic queries.

YES

3. Clarity:

The section is well-written and easy to understand, using appropriate terminology and avoiding ambiguity. However, some complex sentences could be restructured for improved clarity.

Can be Improved

4. Grammar and Style:

The section is generally free of grammatical and stylistic errors, but there are a few instances of unclear phrasing and missing articles.

Can be Improved

5. Typos and Errors:

There are no typos or other errors in the section.

NO

Overall, the section provides a clear and motivated description of the related work in the field of relational learning, highlighting the novelty and potential benefits of the proposed approach. The text could be improved by restructuring complex sentences for enhanced clarity and correcting minor grammatical errors.

3 Relational machine learning

3.1 Contenido:

In this section, we shall leverage the advantages of the framework presented to acquire relational classifiers on graph data sets. To elaborate, we shall initiate from a labelled subgraph set within a graph data set then develop a pattern search technique founded on information gain to obtain typical patterns for each subgraph class.

Information-gain pattern mining

To obtain characteristic patterns of subgraph classes using the previous graph query framework, a top-down decision tree induction will be conducted to explore the pattern space. Within the trees' internal nodes, graph queries will serve as test tools. The best refinement sets will be identified during the tree construction process, resulting in queries that define classes within the graph dataset.

The training set, , consists of pairs (S, y), where S denotes a subgraph of G and y represents its associated class. Every node n in the resulting decision tree is linked to:

- a subset of the training set: \subseteq ,
- a query Q such that: S (S \vDash Q).

The procedure for tree learning is standard: a tree is initialized comprising one node (the root) linked to the entire set of training, . The initial query, Q, corresponds to all its constituents (S , S \vDash Q). The subsequent stage involves determining which refinement set generates the maximum information gain while separating , and applying it to Q. For each query in the refinement set, a corresponding child node is created, and samples are transmitted through it. A child with a matching associated query is guaranteed to exist since it is a refinement set of Q. The recursive process continues for each new node until a stop condition is met. At that point, the node becomes a leaf associated with a class. Note that the decision trees derived from this approach are not predominantly binary, unlike the prevalent trees in the literature.

Relational tree learning examples

Here, we introduce some practical instances to demonstrate the process of performing relational learning by using the query framework and refinement sets. The refinement operations will be as mentioned in Section 1. A critical factor is that all subgraphs in a decision node belong to the same class, which we require as the stopping condition. Initially, we will focus on node classification problems before proceeding to classify more intricate structures.

Consider the small social network illustrated in Figure 2, portraying users and items in a graph. The objective is to classify the nodes based on the patterns extracted from the dataset.

```
\begin{figure}[h!]
\centering
\includegraphics[scale=0.6]{png/FIG8.pdf}
\caption{Social Network toy}

\end{figure}
```

Beginning with a training set composed of all nodes in the graph, Figure 3 displays the relational decision tree acquired through the process elucidated in Section 4. Negative nodes/edges are identified with a cross, while nodes with predicate (v,S):= v S are larger and white in hue. This tree accurately assigns types (User A, User B, or Item) to all nodes in the graph by exploiting relational information from the network. Furthermore, on the leaves of the tree, distinctive patterns are acquired for each node type, which can be used to directly assess nodes and clarify future classifications.

```
\begin{figure}[h!]
\centering
\includegraphics[scale= 0.3]{png/FIG6.pdf}
\caption{Node type classifier}

\end{figure}
```

Similarly, by utilizing each character node in the Star Wars toy graph (Figure 5) and the corresponding specie property as a training dataset, the relational decision tree shown in Figure 6 categorizes and explains each character's species in the graph. The leaf patterns of the tree characterize each species: human characters are born friends of Luke, while droids are unborn friends of Luke, wookies are those born in Kashyyk, etc.

```
\begin{figure}[htb]
\begin{center}
\includegraphics[scale=0.3]{png/FIG7.pdf}
\end{center}
\caption{Character specie classifier}

\end{figure}
```

3.2 Resumen:

This section of the manuscript focuses on leveraging relational machine learning for graph data sets. The process begins with a labelled subgraph set within a graph data set, and develops a pattern search technique based on information gain to obtain typical patterns for each subgraph class. To achieve this, a top-down decision tree induction is conducted, where graph queries serve as test tools. The best refinement sets are identified during the tree construction process, resulting in queries that define classes within the graph dataset.

The training set consists of pairs (S, y), where S denotes a subgraph of G and y represents its associated class. Each node n in the resulting decision tree is linked to a subset of the training set and a query Q such that S   (S \vDash Q). The procedure for tree learning follows standard tree construction methods, with initial queries corresponding to all constituents and subsequent stages involving determining which refinement sets generate maximum information gain while separating the training set.

The section also provides practical examples of relational learning using the query framework and refinement sets, such as classifying nodes in a small social network and categorizing characters' species in the Star Wars toy graph. These examples demonstrate the effectiveness of the decision tree method for obtaining distinctive patterns that can be used to directly assess nodes and clarify future classifications.

3.3 Evaluación:

Based on the provided section text, here is an evaluation of how well it fulfills each of the criteria for a good scientific research article within the technology domain:

Motivation:

* YES: The section clearly explains the study's significance and relevance. It justifies the problem's importance and its wider impacts, providing specific examples from the text.

Novelty:

* Can be improved: The section does not explicitly describe the proposed approach's novelty or originality. It would benefit from more explicit comparisons with related work and highlighting unique contributions.

Clarity:

* YES: The section is well-written and easy to understand, using appropri-

ate terminology and avoiding ambiguity.

Grammar and Style:

* Can be improved: While the section is generally grammatically correct, there are some instances of awkward phrasing or word choice that could be improved for clarity and concision.

Typos and Errors:

* None found.

Overall, the section provides a clear and well-motivated introduction to the proposed approach, but it could benefit from more explicit discussion of its novelty and unique contributions relative to existing work. Additionally, some minor improvements in grammar and style could help to enhance clarity and readability.

4 Conclusions and future work

4.1 Contenido:

The paper's main contribution lies in a novel framework for graph queries that permits the polynomial cyclic assessment of queries and refinements based on atomic operations. The framework's ability to apply refinements in relational learning processes was also demonstrated. In addition, the presented framework fulfils several essential requirements. The system utilises a consistent grammar for both queries and evaluated structures. It allows the assessment of subgraphs beyond individual nodes and supports cyclic queries within polynomial time (where the length of the query path is limited). The system offers a controlled and automated query construction via refinements, and the refinement sets constitute embedded partitions of the evaluated structure set, making them effective tools for top-down learning techniques.

Graph isomorphism-based query systems exhibit exponential complexity when presented with cyclic queries. Additionally, if a projection is necessary for pattern verification, evaluating the non-existence of specific elements becomes difficult or even impossible. However, the query graph framework offered here assesses the existence/non-existence of paths and nodes in a graph rather than demanding isomorphisms, thus enabling the evaluation of cyclic patterns in polynomial time.

After conducting an initial and fully functional proof-of-concept implementation, the graph query framework's capabilities have been demonstrated through experimentation. This methodology has been explicitly applied in relational learning procedures, as demonstrated in section 1, and the results of these experiments have shown that interesting patterns can be extracted from relational data. This is of great significance in both explainable learning and automatic feature extraction tasks. The results' graphs were obtained via our proof-of-concept implementation on a graph database and employing the matplotlib library [1].

Despite the presented query definition utilizing binary graph data sets (rather than hypergraphs), it can be implemented on hypergraph data as well. This is due to the fact that the concept of a path, which connects pairs of nodes, is independent of the edge arity involved. For the sake of simplicity, and due to the absence of true hypergraph databases, our queries have been limited to the binary case. Nevertheless, they can be adapted to more universal cases once the usage of hypergraphs becomes more widespread.

Also, in Section 2, a basic and reliable set of refinement operations have been provided. However, they should not be considered the most suitable solution for all types of learning tasks. To achieve complex queries and to prevent plateaus in the pattern space, more complex refinement families can be established. For example, it is possible to combine the operations add edge and adding property to an edge into one step, thereby reducing the number of steps required. If executed properly, unifying the refinements based on the frequency of structural occurrences in a graph, for instance, can lead to faster versions of learning algorithms at the expense of covering a broader query space. This work provides theoretical tools to support the accuracy of new refinement families. Future research will focus on developing automated methods to generate refinement sets based on a given learning task and the specific characteristics of the graph dataset. Extracting statistics from the graph data for automatic generation of such sets can result in significant optimizations.

It is concluded that it is feasible to establish effective techniques for matching graph patterns and learning symbolic relationships, resulting in systematic exploration of the pattern space, a high degree of expressiveness in queries and computational cost of implementations kept in reasonable orders.

Patterns associated with the leaves in obtained decision trees can be used to characterize subgraph categories. Moreover, the path from the root node to the corresponding leaf of the decision tree for a particular input can be used to justify decisions, a beneficial feature in various sensitive applications as in decision trees. In addition, patterns obtained from the graph learning procedure presented can serve as features in other machine learning methods. Once the patterns have been acquired, they can serve as Boolean features for subgraph modeling, enabling non-relational machine learning methods to learn from them.

Learning of relational decision trees can be utilised by ensemble methods (such as Random Forest), and although the explanatory power is diluted when multiple trees are combined, its predictive power can be greatly enhanced. Therefore, it is essential to investigate the probabilistic amalgamation of queries to generate patterns that can be interpreted as probabilistic decision tools.

Furthermore, while a relational decision tree learning technique has been employed, additional machine learning algorithms can be evaluated alongside this query framework to investigate more opportunities for relational learning.

4.2 Resumen:

The paper introduces a novel graph query framework that enables polynomial cyclic assessment of queries and refinements based on atomic operations, demonstrating its applicability in relational learning processes. The system utilizes consistent grammar for both queries and evaluated structures, allowing the assessment of subgraphs beyond individual nodes and supporting cyclic queries within polynomial time. The paper's proof-of-concept implementation has been demonstrated through experimentation, showcasing that interesting patterns can be extracted from relational data.

The query definition utilizes binary graph data sets but can be adapted to hypergraph data once the usage of hypergraphs becomes more widespread. A basic and reliable set of refinement operations is provided in Section 2, with future research focusing on developing automated methods for generating refinement sets based on a given learning task and graph dataset's specific characteristics.

The paper concludes that it is feasible to establish effective techniques for matching graph patterns and learning symbolic relationships, resulting in systematic exploration of the pattern space and high-degree expressiveness in queries with reasonable computational cost. Patterns obtained from graph learning can be used as features in other machine learning methods, enabling non-relational machine learning methods to learn from them.

Future research will investigate ensemble methods for utilizing learning techniques and additional machine learning algorithms alongside the query framework, aiming to enhance predictive power and interpret patterns as probabilistic decision tools.

4.3 Evaluación:

Based on the provided section, here is the evaluation:

Motivation:

* Clarity: The motivation section clearly explains the study's significance and relevance. The problem of efficiently querying graph data is well-justified, and the proposed approach's potential impact on explainable learning and automatic feature extraction tasks is highlighted. (Provide specific examples from the text.)

Originality:

* Novelty: The section clearly describes the proposed approach's novelty or originality. The framework's ability to assess cyclic queries in polynomial time and its potential for top-down learning techniques are highlighted. However, it could be improved by explicitly comparing with related work and highlighting unique contributions.

Clarity:

* Comprehension: The section is well-written and easy to understand. It uses appropriate terminology and avoids ambiguity. However, some complex sentences could be restructured for better clarity.

Grammar and Style:

* Correctness: The section is generally free of grammatical and stylistic errors. However, there are a few minor errors that could be corrected, such as missing articles and inconsistent capitalization.

Typos and Errors:

* Accuracy: There are no typos or other errors in the section.

Overall, the motivation section provides a clear explanation of the study's significance and relevance, and it highlights the proposed approach's novelty and potential impact. However, there is room for improvement in terms of clarity and comparisons with related work.

Evaluation Level: Must be Improved