This dissertation has been submitted by Pedro Almagro Blanco in partial fulfillment of the requirements for the Degree of Doctor of Philosophy by the University of Seville.

D. Fernando Sancho Caparrini

Seville, April 2017.
Contents

1 Introduction .......................................................... 1
   1.1 Property Graphs ............................................... 2
   1.2 Machine Learning on Graphs ................................. 3
   1.3 Thesis Goals .................................................. 5
   1.4 Contributions ............................................... 6
   1.5 Thesis Structure ............................................. 7

2 Conclusions and Future Work ................................... 9
   2.1 PQG and its use for PQG-ID3 ............................... 11
   2.2 Semantic Embedding ......................................... 14
   2.3 Future Work .................................................. 16

Bibliography .......................................................... 42
Man’s curiosity to understand the environment in which he lives has allowed him to advance in his way of dealing with the unknowns and day to day problems. This primary need to understand the behavior of the phenomena that surround him leads him to create increasingly efficient methods used to discover and explain the workings of what he can perceive through his senses. For example, the enunciation and understanding of universal laws of physics allows him to emulate hypotheses about the past conditions of phenomena, to reason about their relation to the present that he perceives, and to predict their future evolution, going ahead in the decisions about events that have not yet occurred. Without possessing a perfect system of inference, it is certainly the capacity that differentiates him the most from the rest of living beings with which he shares his existence.

However, he must repeatedly redo and expand the discovery mechanisms he uses, adding new tools that allow him to continue to expand the frontiers of knowledge. Among past tools, the Scientific Method, and the scientific paradigms that are established by its application, stand out as great referents, being that they are characterized by demanding a rational justification of principles that needed to be proved and by rejecting absolute affirmations, assuming that any of them is susceptible to being refuted. The oldest scientific paradigms are the empirical (validate a hypothesis through experimental repetition) and the theoretical ones (to prove something through logical derivations). Later on, immersed in the computer age, a new scientific paradigm emerged that has come to be called a computational paradigm (to validate something through computational simulations that complement experimental observations). In recent years, and as a consequence of the use of computer systems for mass data processing, a new paradigm of data-based inference has emerged, which is giving rise to an emerging discipline called Data Science which, despite not being new, only in the last few decades has an effort been made to endow it with solid theoretical
foundations and viable implementations, which has placed it in the spotlight of many economic, social and research interests.

Among the phenomena that can be approached with these types of mechanisms, those in which both the description of the elements and the interactions between them are important are especially interesting, especially when there may be different types of interactions or when we have to consider the internal properties that characterize them. The data structures that allow to express this type of phenomena usually are oriented to the description of its elements, to the relations between them, or have a hybrid orientation that allows a similar level of expressiveness for both.

Most methods in Data Science are oriented to work with structures that naturally express the properties of the elements, but usually have certain limitations to express their interactions in a natural way. For example, vectors or tables, with which the most widely used machine learning algorithms usually work, are well suited for describing elements, but not for expressing the complex relationships between them. Classic databases provide mechanisms for expressing complex relationships between elements, but the available query and modification tasks are not optimized to work on features derived from these relationships.

The mathematical structure that seems to better express the interactions between elements of a system is the graph, and recently extensions of these have started to appear, such as property graphs, that allow to express in a natural way all the components that make up a system: its elements and the relationships between them. Based on this conceptual structure graph databases were developed, which allow to store the information in such a way that both the storage process and the subsequent query process are efficient.

1.1 Property Graphs

As previously mentioned, mathematical objects usually used to model system’s elements tend to have some linearity in their structure and have good computational implementations (vectors, registers, tables, ...). For those cases where it has been necessary to have less linear structuring tools, ad-hoc implementations have been created that have covered the desired requirements (XML, RDF, ...). In search of a common mathematical object that models this type of data structures, property graphs stands out as a concept that provides a good balance between correctly expressing the description of the elements of the system and that of their relationships. This structure, which has not been formally defined until a few years ago, does not yet have a robust theory to
support it, although, as we shall see, a correct definition can make it contain different types of abstract structures that support databases.

Historically, uni-relational graphs (those in which all relationships are of the same type) formed the basis of classical mathematical graph theory. Multi-relational graphs (those in which different relationships may have different types) are closer to the intuitive description of a system. Property graphs allow each element (node or relation) to have an indeterminate number of associated heterogeneous properties. With these premises, this structure is able to contain practically any other relational structure and it can be adapted to the level of detail required for the system according to the needs of later analysis.

For these reasons, in this work we will use this type of mathematical objects as a base structure, both in the previous phase of modeling and storage of information and in the later phase of analysis and inference about it.

1.2 Machine Learning on Graphs

Once we have selected a suitable mathematical structure to describe the systems that interest us, we need tools that allow us to make inferences automatically about them, usually to be able to predict them. Traditionally, the various scientific paradigms have made use of tools based on logical, geometric, algebraic, analytical, etc., and more recently they make extensive use of algorithmic tools.

In the same way as in the scientific method, the enunciation of hypotheses after the observation of phenomena allows extracting general laws from experience, the use of computational techniques can be useful to obtain general laws from examples analyzed with algorithmic tools. This process can be done at various levels. We can make use of computational tools as a means to aid the process of inference of the researcher, or we can try to create an algorithm capable to make the complete inference and enunciate laws itself, automatically. In the latter case, we say that we are dealing with a machine learning process (discussed more fully in Chapter 2).

Originally, most of these techniques had been devised to generalize from a series of isolated examples, elements described through a series of predetermined properties, usually expressed in the form of registers or tables. This way of structuring the examples coincides with the first type of structures presented in the previous section, those that allow to correctly express the properties of the elements that make up a system, but not their relations. This limits learning ability as it does not explicitly consider an important part. If in the phenomenon under study (and which is intended to learn from) the interactions seem to be determining in the understanding of the same, we must choose a
mathematical structure that reflects them correctly. However, the efforts in the
development of machine learning seem to have left these considerations behind.

Designing algorithms that learn from structured data in the form of property
allows to learn in a more natural way both on the basis of the properties of the
elements of a system and the relationships between them. In addition, although
some information is always lost from the real phenomenon, the flexibility of
graphs allows a greater adaptation without making too many transformations
that would break from reality.

If we allow machine learning algorithms to work with graphs as the natural
structure from which to learn, they can manipulate relations explicitly in the
same way as they do with the properties of the elements. This type of learn-
ing has come to be called relational learning (multi-relational in case there
are several types of relationships between data) and luckily, despite not being a
focus of attention, has made great strides and has been an active research area
for many years. It is usual in literature to find relational learning divided into
three blocks: (1) Statistical Relational Learning (SRL), within which develop-
ments like the Markov logical networks would be included, that uses a coding
of multi-relational graphs making use of probabilistic models; (2) Path Ranking
Methods, which explicitly explore the relationship space through random paths;
And (3) Immersion Based Models, which obtain a vector representation of the
graph through matrix / tensor factorization, Bayesian clustering or neural net-
works. In addition to these three blocks, we can include the algorithms that
perform the discovery of relational patterns by refining a hypothesis through
a series of steps, in this last block we could include algorithms like Top-Down
Induction of Logical Decision Trees, Multi-Relational Decision Tree Learning
or Graph Based Induction Decision Tree, which we will detail in Chapter 4 of
this report. As can be observed, advances have been made in relational learning
using decision trees, neural networks and probabilistic models, among others,
but there is still a long way to go, and there are other algorithms that, even
though they have demonstrated great potential in learning from non relatio-
nal data, have not yet been exploited from this perspective, as is the case of
Random Forest.

Usually, according to various criteria, machine learning models are classified
in Supervised vs. Unsupervised (by type of learning carried out), Regression,
Calsification or Ranking (by the type of expected output), etc. In addition, and
with respect to the interpretability of the results by a human, we can classify
the models in those that are able to offer an explanation that accompanies and
justifies the result they provide (white box models) and those that sacrifice such
justification for better efficiency (black box models).

With regard to the white box methods, one of the most representative mo-
1.3 Thesis Goals

Given the small number of methodologies that perform relational machine learning, the main objective of this research has been to provide new methods to carry it out, as well as to optimize some of the existing ones. In order to carry out this task, and without naming objectives related to bibliographical or comparative revisions between models and implementations, a series of concrete objectives to be covered are proposed:

1. Define flexible and powerful structures that allow phenomena modeling based on the elements that compose them and the relations established between them. Such structures must be able to express naturally complex properties (continuous or categorical values, vectors, matrices, dictionaries, graphs,...) of the elements, as well as heterogeneous relationships between them that in turn may possess the same level of complex properties. In addition, such structures must allow to model phenomena in which the relationships between the elements do not only occur between pairs, but also between any number of them.

2. Define tools to build, manipulate and measure such structures. However powerful and flexible a structure is, it will be of little use if you do not have the right tools to manipulate and study it. These tools should be efficient in their implementation and should cover construction and consulting tasks.

3. Develop new black box relational machine learning algorithms. In tasks related to automatic classification and regression black box models can be used, since the goal is not to obtain explanatory models, this characteristic can be sacrificed for better efficiency.

4. Develop new white box relational machine learning algorithms. When an explanation about the operation of the systems being analyzed is needed we will look for white box models.

5. Improve query, analysis and repairing tools for databases. Some of the queries in databases are computationally expensive, preventing us
from proper analysis in some information systems. In addition, graph databases lack methods that allow to normalize or to repair the data automatically or under the supervision of a human. It is interesting to develop tools that carry out this type of tasks increasing efficiency and offering a new layer of query and standardization that allows to cure the data for a more optimal storage and recovery.

All marked objectives must be developed on a solid formal basis usually based on information theory, learning theory, artificial neural network theory or graph theory. This basis will allow the results obtained to be sufficiently formal so that the contributions made can be easily evaluated. It is also sought that the developed abstract models be easily implemented on real machines to be able to verify experimentally its operation and to offer to the scientific community useful solutions of this type in a short space of time.

1.4 Contributions

The work carried out has meant an incursion into the formalization of graphs and relational machine learning and, as reflected in this report, has allowed to unify and condense different perspectives in these areas. In addition, it has allowed the development of new techniques to carry out this type of tasks using more general formalizations as well as making use of new learning methods that are able to work with property graphs as basic structure from which to learn.

We describe below the contributions that can be found in this work:

1. **Generalized Graph**, simple mathematical structure that generalizes virtually all classical definitions of graph, from uni-relational graph to property hypergraph. In this paper, concepts belonging to graph theory are redefined from this new perspective and a solid, simple and flexible basis is provided to support systems in which both the description of the elements and that of the relations are important.

2. **Extension of the measures** already defined for uni-relational graphs to the generalized graph concept. Because this structure generalizes most of the range of definitions for graphs, this extension allows to make measurements on the different types of graphs existing.

3. **Property Query Graph**, query tool for generalized graphs that allows to evaluate structures based on their content and the elements with which it is related. Query languages such as Cypher and other query tools such
as Selection Graphs are specific cases of this tool. PQG are expressed through a generalized graph, allowing to work with the same structure for both the information source and the query.

4. **PQG-ID3**, relational machine learning algorithm that allows to discover patterns in enriched structures of data, and to construct decision trees to classify subgraphs from a set of classified examples. The relational patterns extracted by this algorithm are expressed through a generalized graph, allowing its easy interpretation by any human / machine.

5. **Methodology for the embedding of generalized graphs in vector spaces**, maintaining the original semantics and allowing the discovery of new information, retrieval of missing information, automatic classification of data and providing improvements in other tasks such as long distance queries.

6. **Implementations** of the tools developed throughout the work, and detailed in the Implementation Appendix.

In addition to the points indicated, no less significant are the following contributions: a first step towards a tool for the normalization of graph structured data through analysis of vector structures obtained from embedding; A first family of refinements that allow automatic manipulation of complex predicates on graphs. In addition, throughout this report, you can find other minor conceptual and technical contributions that have not been named in this section.

## 1.5 Thesis Structure

The content of this report is described below, describing the various chapters of which it consists and that pretend to cover the research goals marked previously.

In Chapter 2, **Fundamentals**, we introduce the fundamental theories that serve as a transversal axis to all the work done, we present the common structures for the different chapters of this report. We present a framework for graphs that unifies definitions as a uni-relational graph or graph with properties through the concept of generalized graph, redefining concepts belonging to Graph Theory. In addition, an introduction to machine learning is presented and the models that will be used throughout the memory are briefly presented.

In Chapter 3, **Property Graph Pattern Matching**, a study about the evaluation and extraction of information in relational structures is made, presenting a review of the technologies and foundations that make this task possible. In addition, we present the Property Query Graphs (PQG), our proposal to
evaluate structures immersed in a property graph, and which works as predicates on subgraphs, so that they are ideal as a basis for discovery tasks. The chapter concludes by showing some concrete examples of PQG.

In Chapter 4, Decision trees for property graphs, we face the problem of automatically constructing subgraph classifying trees in property graphs. It begins by reviewing the operation of decision tree induction algorithms, from those that learn object (described through a set of properties) classifier trees, to those who learn to classify records from a relational database taking into account their relationships. The central part of the chapter presents the PQG-ID3 algorithm, our proposal to build multi-relational decision trees based on PQG. We conclude by showing a collection of examples of trees constructed using this tool and analyzing the resulting semantic patterns.

Chapter 5, Semantic Embeddings of Property Graphs, makes a different approach to multi-relational learning, this time making use of neural networks to learn a vectorial encodings of property graphs. This encoding allows to make use of the usual machine learning methods designed to work naturally with objects described vectorially. A review is made of the learning methods that make use of neural networks, and we analyze methods that have been used to obtain encodings of other types of structures. We present a methodology that makes use of neural encoders to carry out property graph embeddings in vector spaces, experiments are carried out on real data to verify that the obtained projection allows to capture properties present in the original graph. In addition an empirical demonstration is carried out, proving that the proposed immersion methodology allows to successfully perform machine classification tasks, information extraction, missing information retrieval and long-distance queries.

Finally, in the chapter Conclusions and Future Work, we present the conclusions obtained from the research process structured in accordance with the chapters of this report, as well as the conclusions obtained globally across the work. In this last chapter we also present possible future lines of work, made possible by this research and that are related to the formalization of graphs, relational machine learning, graph pattern matching and discovery procedures in databases.
Conclusions and Future Work

As discussed in various sections of this report, and despite its potential, relational machine learning has been in the background in relation to the more standard machine learning, which makes use of non relational information, usually in form of tables and other regular structures. After the work done, we can identify certain reasons that have led to this situation.

On the one hand, we have detected that the scientific community has some inertia in its research, which leads it to prioritize the optimization and modification of existing algorithms over the creation of new ones, or the use of new data structures from which to learn. Certain scientific publications get impact by overcoming well-founded earlier results and this may be a reason why entering a new method is not as fruitful (academically speaking) for the researcher as continuing a methodology that has already proven to be well founded and useful. The creation of completely new models usually does not find the adequate platforms for the presentation of its work or simply requires a greater validation effort.

On the other hand, the most commonly used information systems, in which we store most of the studied phenomena, make use of schemas and systems based on relational databases. As many studies have shown, these classical databases do not show an optimal performance when working with complex relationships, which is one more reason for the impediment of developing methodologies oriented to work well with elements and their relations.

In addition, the greater expressive richness of the more complex information imposes difficulty in making new algorithms and provides, at least in the first approximations, results less striking than the more refined and more traditional methods.

Finally, there is a Rich gets richer process, the more conventional methods are better known by the scientific community and, therefore, are known by
more researchers, which causes that the majority of them to work with non relational machine learning methods.

We would like to emphasize that to reason about the formal structures used to store the information concerning the systems we analyze is essential if we want to carry out this type of tasks in an optimal way. Researchers often transform data obtained from a system into one of regular formats, usually vectors or tables, missing facets of important relational information whose expression is not natural in these formats. When analyzing a system through techniques derived from Machine Learning can become as important the data structure used to express the information as the algorithm used. After this research, we consider that there is no proportional effort in the area between optimizing the structures from which to learn and the chosen learning methods. In this paper we have explored the learning capacity from structured data in the form of property graphs.

With respect learning from property graphs, it is possible to emphasize that there are several lines of work that transform the original data towards other structures that algorithms are able to handle of more natural way (because they were created to work specifically with such structures). This is the case of graph embeddings in vector spaces. In our view, these are valid approximations that should continue to be investigated, but other options should be considered, such as working directly with the graph structure, which has been one of the lines of research followed in this work and which has proved to be valid.

At times, efforts to work with data are focused on obtaining automatic predictions that quantitatively improve previous results, usually measured through benchmarks. However, there are methods related to prediction that can provide results (quantitative and qualitative) that are not easily measurable through this type of techniques. In addition, there are other interesting tasks, not related to prediction, that can be carried out with data and that have not received the attention they deserve. For example, analyzes related to the semantic purity of a data set can be interesting to evaluate the structure of the data set and to detect inconsistencies such as overlap between data types or redundancy in the data or schema, as well as improving the efficiency of queries on these datasets.

Another related task, is the discovery of information through white box models. If we analyze the trend in learning, relational or not, we can observe that in recent years many efforts have been invested in black box methods, with explanatory methods remaining in the background. Explanatory algorithms like MRDTL emerged at the end of the 1990s and it seems that their development has stalled in recent years. Undoubtedly, black box models are showing shocking, unexpected results, through their greatest exponent, Deep Learning, but whose interpretation is too diffuse to be understood by a human (although
there are many efforts to create tools that could narrow this gap). We con-
sider that this fact can be dangerous, black box methods allow us to predict
the evolution of systems to a certain extent, but prevents humans from realizing
real learning about how the system works, simply providing a tool that predicts
it but does not add additional knowledge to the researcher. In this way they
become useful tools for Engineering (which justifies the work done on them),
but not for Science. If we want to know the phenomena that surround us and
advance in the understanding of our surroundings, we must find ways to under-
stand it. Therefore, we consider that a superior effort should be made in the
study and development of white box machine learning methods.

We will now take a more detailed look at some conclusions that can be
derived from the different approaches in this work.

2.1 PQG and its use for PQG-ID3

Chapter 3 addressed the goal of obtaining a tool to evaluate subgraphs in pro-
perty graphs that can be used in discovery procedures in relational information.
To achieve this goal, several requirements had to be fulfilled. On the one hand,
it was necessary to have a grammar to express the queries in a way close to the
structures on which it is going to work. And thanks to the expressive capacity of
property graphs, we have presented a query tool that can be expressed naturally
by means of a property graph. In addition, it was necessary to provide queries
that when used as logical predicates on graphs, they behaved consistently and
robustly. In addition, it was necessary, since we will also use them to generate
machine learning methods, that the queries could be modified in a controlled
way by means of atomic operators that translated the topological control into
a logical control. In this sense, a first family of refinements (a refinement acts
as a query partition) has been introduced that allow an ordered collection of
queries to be constructed from an initial query (which may be empty).

Any relational data structure can be viewed as a graph and any query can
be viewed as pattern matching, thus, most query languages in databases can
be viewed as Graph Pattern Matching (perhaps primitive) tools in property
graphs. In Chapter 3 we have also analyzed some of the existing Graph Pattern
Matching tools as well as the feasibility to be used in automatic procedures.
One of the tools analyzed, Selection Graph, allows to evaluate registers in rela-
tional databases using acyclic patterns that can be refined through basic ope-
rations, allowing to obtain complementary patterns in each case. It does not
require an exact projection of the pattern representing the selection graph on
the subgraph to be evaluated, but rather the fulfillment of a series of predica-
tes expressed through said pattern. It should be noted that if a projection is
required when carrying out the verification of a pattern, the task of evaluating the non-existence of certain elements is complicated. Specifically, the selection graphs evaluate the existence / non-existence of paths that are incident to the registry under evaluation (they are only able to evaluate individual records). It is verified if a conjunction of predicates on paths that depart from the analyzed registry is fulfilled, which can be seen as the evaluation of the existence of a tree rooted in the node that represents the registry under evaluation.

Property Query Graph, the tool presented in Chapter 3, extends the concept of Selection Graph allowing the evaluation of general subgraphs, beyond a single node, using predicates through the definition of a language on the elements of the graph and allowing cyclical patterns. As it becomes a requirement not to use a projection for the verification of a pattern, these objectives have been achieved by extending the form of evaluation, which can be seen as the evaluation of a tree rooted by each node present in the pattern. Although each node of a PQG evaluates the existence of a node that fulfills the conditions imposed by its predicate and the edges in which it participates, by allowing the edges to be identified with paths in the graph (Regular Pattern Matching) there is the evaluation of one tree per node, not a single star. It is through the intersections that occur between the various trees and the constraints imposed on the nodes as the evaluation of cyclical patterns in PQGs is allowed.

Like the selection graphs, the PQG can be modified and constructed from refinements, but unlike the simple case of selection graphs, refinements are usually not binary, since their application can modify more of one predicate in the pattern, resulting in sets of size $2^k$ (where $k$ is the number of modified predicates). As shown in Chapter 4, this is not a problem when it comes to building learning models, such as decision trees, since they do not have to be binary. Through the definition of certain operations of simplification and equivalence, the refinements shown can be simplified giving rise to simple tools that allow to express complex queries in graphs.

In general, refinements result in partitions of the structures they evaluate, making them ideal tools for white-box procedures. After carrying out a first (but fully functional) proof-of-concept implementation, it has been experimentally demonstrated that PQGs are viable under mild conditions and meet the stated objectives.

An explicit use of these capabilities is shown in Chapter 4, with the presentation of the algorithm PQG-ID3, which makes use of the Property Query Graphs as test tools for the construction of a decision tree following the foundations of ID3 algorithm. In the results of the experiments carried out, it is shown that PQG-ID3 is able to extract interesting patterns that can be used in complex learning tasks. PQG contained in the leaves can be considered as new
attributes discovered by the algorithm. In this way, in addition to constructing a classifier tree, the algorithm is able to discover patterns that characterize different structures in the graph (Graph Pattern Mining) and that can be used as attributes of the structures that classify in later tasks (Feature Extraction).

MRDTL algorithm can be seen as a particular case of the algorithm PQG-ID3 in which only PQG with tree form are allowed (since they use selection graphs) and where it learns only from structures formed by a single node. In this sense, PQG-ID3 is a leap forward in a line of work started years ago and considered open since then. As a curiosity, we have to say that the work done on PQG-ID3 was done in a completely independent way, and it was only when writing this memory that we could relate it to the selection graphs and the MRDTL algorithm.

As we saw throughout the chapter, the main problem presented by multi-relational decision tree construction algorithms is that the hypothesis space is extremely large. To solve this problem several solutions can be proposed. On the one hand (and as an extension to the proposal in MRDTL-2), the frequency of occurrence of certain structures can be analyzed in a statistical way with the purpose of reducing the number of possible refinements to be applied in each case and thus to reduce the cost of the search for the best refinement. All of this prior analysis makes use of the various measures introduced in Chapter 2 (and which extend the simpler frequency measures used in the case of MRDTL-2). On the other hand, you can create more complex refinement families (for example, combining the refinement add edge with adding property to an edge in a single step) to reduce the number of steps to get complexes PQG. If this last option is carried out properly (unifying the refinements according to the frequency of occurrence of structures in the graph), the algorithm can be brought closer to the solution faster. In both cases, an improvement in efficiency is achieved by sacrificing the possibility of covering a wider hypothesis space (but probably offering alternatives in which the impurity reduction is smaller). In this sense, a minimal set of well-constructed refinements has been offered in this paper, but it should be borne in mind that they are not offered with the intention of being optimal for all learning tasks.

The second major problem with the PQG-ID3 algorithm (and inherited by all algorithms inspired by ID3) is the inability to undo the decisions made during the construction of the tree. In such a way that the options of refinement in a determined step of the algorithm depend on the refinements chosen in previous steps. To solve this problem, it is usual to use some backtracking procedure to undo decisions if they have resulted in a bad result or a use a Beam-Search procedure as used in the GBI algorithm that allows you to make several decisions in parallel and finally select the one that has resulted in a better solution.
Consequently, queries on graph based on PQG allow us to obtain powerful and simple tools suitable for automatic construction and to be used in white-box tasks on multi-relational information with controlled complexity, due in part to good properties related to complementarity and containment of queries. In addition, the combination of the PQG decision tree type through an aggregate model, such as Random Forest, can achieve very good results when performing automatic classification (although this will reduce the interpretive capacity of the models obtained).

2.2 Semantic Embedding

The purpose of this last chapter has been to offer the possibility of performing relational machine learning tasks through more traditional algorithms by making an automatic feature selection. In this way, and in addition to the approach presented in the two previous chapters, we try to analyze what options traditional algorithms offer when we want to not lose the enriched structures of relational information.

If there is an element (a subgraph) that is immersed in a database (a property graph) the task of constructing attributes for learning from the relationships that it presents in the global structure can be very complicated. The approximation presented in this chapter consists of constructing a vector representation of each element in the system from a sampling of the information present in the network. In this way, we avoid, on the one hand, the manual work of selecting the attributes to be taken into account and, on the other hand, we obtain a learning algorithm which feeds from a representation of the elements of a graph obtained from global information.

Compared to other machine learning tasks, there are few jobs that have used neural encoders to perform property graph embeddings in vector spaces. Our methodology has sought to use simple architectures to obtain vector representations that maintain the semantic and topological characteristics of the original graph. In addition, it has been demonstrated experimentally that with the obtained embedding one can obtain semantic connections that do not appear explicitly in the original graph (due to incompleteness in the stored data, or to inconsistencies), or even to optimize queries in databases.

We have verified that the geometric characteristics of the structures formed by the nodes and edges in the new vector space can help to assign missing types or properties to the original graph elements (using measures related to distance, linearity, or clustering, among others), or may even help identify new relationships between elements that are not explicitly present in the original
2.2. Semantic Embedding

This functionality can be very useful in processes that work with large sets of relational data, where incompleteness of data is a common obstacle.

In addition, as has been observed from the evaluation tests, the performance and accuracy of machine learning tasks on these vector representations can provide information about the semantic structure of the data set itself, and not only about the algorithms in use. For example, the confusion of some nodes / edges in classification tasks can give us information about the need to make an adjustment in the data schema to reflect the semantic characteristics correctly. A detailed report on how different types, properties, and clusters overlap and confuse in the resulting embedding would be useful for making decisions related to the standardization of data schemas, something that almost all current analysis proposals lack.

It is evident that the size of the training set and of the selection window positively influence the application capacity of the resulting embedding, but these influences must be studied deeper, since they can throw keys for the automation of the embedding parameters.

In addition, this chapter has explored how vector structures can be used to retrieve information from property graphs, as shown in Entity Retrieval and Typed Paths experiments. Looking for complex structures in the projected space can be simpler than in the original one. In fact, the use of a second layer of learning models after neural encoding can improve the results of various tasks related to the retrieval of information in semantic graphs. The results show that this is a line of research that is worth considering. Although not enough experiments have been carried out on long-distance queries through the representative vectors in the new space, the results obtained show that the query times can be reduced considerably, sacrificing the optimality. This type of queries are very expensive in the databases, and although graph databases have helped to reduce their computational cost they continue to present great problems of efficiency.

Compared to other approaches in the same direction, this paper presents the novelty of working with more general semantic contexts, and not only with random paths, which assume a linearization of the original graph structure. But these are not the only options to carry out property graph embeddings through neural networks. As will be discussed in 2.3, we can get continuous embeddings of property graphs using neural autoencoders, so that the neuronal encoder will learn the identity function for the elements of the graph, allowing the encoding to work without any bias imposed by any function that relates the elements to their context.

With this work we have given an initial framework to perform machine learning tasks from property graphs in which we take into account information
from the complete graph to encode each element. This new representation of property graphs allows to work with relational data stored in almost any system of persistence in a vectorial way, taking advantage of the power that the processors and GPUs currently have to work with this type of structures.

It should be noted that during the review of this document new tools based on the Word2Vec architectures have been published, they optimize the process of learning latent semantics from natural language [34]. In spite of the probable improvement that these tools would suppose in our methodology, we have decided not to take them into account since they do not imply a change in the fundamental part of our results, although it would possibly improve the associated computational cost.

2.3 Future Work

In this last section we want to show some of the new lines opened by this research. Some are already being studied, while others represent simple ideas that have emerged during the work and have been targeted to be addressed whenever possible. As far as possible, we will try to maintain the natural order that has been followed in this report.

The power obtained through the definition of a language on the elements of a graph such as the one presented in this work has allowed to construct a discovering tool that generalizes and enhances the standard in multi-relational decision tree construction. However, the restriction that we have imposed that such predicates can only evaluate nodes, edges or paths, implies that the global pattern represents a predicate that is finally a conjunction of relatively simple ones. These types of patterns are constructed through some structure in the form of a graph that unifies said predicates, several trees in the case of PQG-ID3. In a PQG, the predicates that compose it have a node perspective (the PQG is constituted by a predicate for each node it owns), but PQG could be constructed to evaluate another structure, for example, PQG in which assigns a predicate for each cycle of length 3 (triangles perspective). This limitation suggests that the concept of pattern, as hitherto conceived, is not general enough to express powerful and flexible predicates about relational data. A recursive definition of pattern, which could lead to a redefinition of the concept of graph through the recursion of structures by levels, could allow to express a larger set of patterns without losing power, flexibility, or the capacity to be constructed through complementary refinements such as those presented.

In PQGs, the sign of nodes and edges have a clear intuitive interpretation: positive elements are must be found in the graph under evaluation, and negatives
impose non-existence constraints. Because adding constraints to a non-existence condition resulting in a less restrictive condition, negative elements have not been amenable to being refined through refinements presented. However, there is no reason not to investigate possible ways of refinement through the negative elements. For this, it would be enough to propose this type of predicates based on disjunctions, in this way the situation would be the inverse and the negative restrictions would be susceptible of being refined. Thus, in order to improve the utility of PQGs and the tools derived from them, a way of working should be in generating families of refinements that expand the expressive capacity of PQGs that can be built automatically.

The advances made in multi-relational decision trees through PQG should be used by the family of ensemble methods and in particular by Random Forest. As we have discussed, the interpretative ability of decision trees is diluted when several trees are combined to explain the same result, but they can greatly expand their predictive capacity. Another option derived from the use of methods combined with decision trees using PQG is that the trees obtained have tests in their leaf nodes that evaluate complete semantic patterns, so an option would be combining them in a probabilistic way to give rise to combined patterns that can be interpreted as probabilistic decision tools, opening up an interesting line in white box relational machine learning.

PQG presented in chapter 3 represent predicates on property subgraphs that are capable of evaluating characteristics beyond the structural and semantic properties of the subgraph under evaluation, since they allow to express restrictions in the surroundings of said subgraph (this surroundings can become all the graph in which it is immersed, if the appropriate predicate is used). This feature, in addition to the ones already discussed, make the PQG into descriptors of relational structures (whether they were built automatically or manually designed by experts in the area), erected as suitable candidates to be used as additional attributes in relational learning tasks. In addition, as already mentioned, the complexity in the PQG-ID3 method can be reduced by using statistical analysis to evaluate the frequency of occurrence of different patterns in the graph and in this way reduce the possible refinements available in each step, or combine several refinements into one.

The efficiency improvements in long-distance queries deserve to be evaluated in greater depth and compared with other similar methods. Some results related to the semantic analysis of property graphs have not been carried out in depth and have not been presented in this report although they are expected to be presented in later works. Options such as sampling the context of the edges, perform a embedding of the same and from this infer an embedding for the nodes have not been taken into account and can offer interesting results. With respect to property graph embeddings in vector spaces through neuronal
encoders, it must be taken into account that, having inspired us in the architectures corresponding to Word2Vec, the function that this encoder tries to learn relates each node with its context and this therefore determines the distribution of the obtained embedding. In this work it has not been considered that this encoder could learn other functions, but we consider that it is a point to take into account since the function that learns the encoder is determinant in the use of the subsequent embedding. For example, if we use the identity function (in this case the network would be an autoencoder) we could get an aseptic immersion, not determined by any previous criteria. This would allow to avoid the problems derived by the definition of the contexts and in other works related to graph embeddings through random walks. Another option could be to use patterns like PQG to perform the encoding. Given a structure, the function to be learned by the coder will relate it to its associated PQG, so the obtained embedding would reflect the semantics associated with the PQG used, and the resulting representation could be optimal if supervised (classification) or non-supervised (clustering) learning is later related to the structure shown in the PQG. Undoubtedly, the possibilities of mixing the expressiveness provided by patterns such as PQG and the efficiency and performance provided by the vectorial representations shown are broad and promising.

During the conception, implementation and experimentation of this work new research lines have been opened that can be considered to analyze the characteristics of the obtained embeddings.

A first consideration is how to construct the training set that is consumed by the neuronal encoder to obtain the vector representation of a property graph. In the experiments carried out the construction of the training set has been totally random, that is, all nodes have the same probability of being sampled, as well as all their properties and neighbors. This may not be the most appropriate way depending on the type of activity to be performed with the resulting embedding. For example, it may be beneficial to construct the training set so that those nodes with a greater semantic richness are more likely to appear in it, which may contribute to regions that are less likely to be considered.

Another line to take into account is to build a neural network that works with the contexts of an element as input (in one-hot format) and learn to return a particular property of the element as an output, i.e. connect a neuronal classifier directly with the encoder, in order to learn the proper encoding and classification simultaneously. Similarly, it would be interesting to think of neuronal encoders that make use of recurrent neural networks to be able to analyze the behavior of dynamic relational information.

It should also be noted that the possibility of working with continuous properties in nodes and edges is open, this feature is not present in the datasets
used, but should be considered to expand the capacity of presented methodology. For both PQG and continuous embedding there are direct mechanisms to include the presence of continuous properties, it is still a matter of work to begin by testing these more obvious mechanisms and then to measure the extent to which other approaches can be taken into account.

In summary, presented research has opened numerous lines of work in various connected areas. The most evident have been in the formalization of relational structures, formalization of procedures for constructing queries about them, relational machine learning and relational knowledge discovery, feature extraction, representation learning, and analysis and normalization of data. Some of them have been presented here but, undoubtedly, new challenges will arise in the form of ideas from this work. For this reason, we are pleased to present a thesis in which, despite having meticulously addressed the initial objectives, more questions have been opened that roads have closed. Undoubtedly, this profusion of possible ways of continuity shows that the study of relational information systems can become a fruitful line of research worth paying attention to.
Bibliography

[6] Xml 1.0 origin and goals. https://www.w3.org/TR/REC-xml/#sec-origin-goals


[48] Kai-Wei Chang, Scott Wen-tau Yih, Bishan Yang, and Chris Meek. Typed tensor decomposition of knowledge bases for relation extraction. In


[140] Juan Li. Improved multi-relational decision tree classification algorithm.


[252] Xiaoxin Yin, Jiawei Han, Jiong Yang, and Philip S. Yu. *CrossMine: Efficient Classification Across Multiple Database Relations*, pages 172–195. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[253] Xiaoxin Yin, Jiawei Han, Jiong Yang, and Philip S. Yu. Efficient classification across multiple database relations: A crossmine approach. *IEEE Trans. on Knowl. and Data Eng.*, 18(6):770–783, June 2006.


