RULE MINING IN FEATURE SPACE

Stefano Teso & Andrea Passerini
Department of Information Engineering and Computer Science
University of Trento
Trento, Italy
{teso, passerini}@disi.unitn.it

ABSTRACT

Relational embeddings have emerged as an excellent tool for inferring novel facts from partially observed knowledge bases. Recently, it was shown that some classes of embeddings can also be exploited to perform a simplified form of rule mining. By interpreting logical conjunction as a form of composition between relation embeddings, simplified logical theories can be mined directly in the space of latent representations. In this paper, we present a method to mine full-fledged logical theories, which are significantly more expressive, by casting the semantics of the logical operators to the space of the embeddings. In order to extract relevant rules in the space of relation compositions we borrow sparse reconstruction procedures from the field of compressed sensing. Our empirical analysis showcases the advantages of our approach.

1 INTRODUCTION

Knowledge Bases (KB) capture relational knowledge about a domain of choice by modelling entities and facts relating them. In so doing, KBs allow for rich answers to user queries, as happens with the knowledge panels powered by the Google Knowledge Graph. Furthermore, KBs can be mined for rules, i.e. patterns of relations which are frequently found to hold in the KB. Mining theories from data is the task of Rule Mining (Dzeroski & Lavrac, 2000) and Inductive Logic Programming (Dzeroski & Lavrac, 1994; Muggleton et al., 1992).

Classical ILP methods mine theories by searching over the (exponentially large) space of logical theories, resorting to language biases and heuristics to simplify the learning problem. While powerful, pure ILP methods do not scale to large relational datasets, preventing them from mining Web-scale KBs such as YAGO (Hoffart et al., 2013) and DBpedia (Auer et al., 2007). Further, purely logical methods can not gracefully deal with noise. Next-generation miners that specialize on large KBs, such as AMIE (Galárraga et al., 2015), work around these issues by trading off theory expressiveness for runtime efficiency.

A general strategy for processing huge datasets is *dimensionality reduction*: instead of working on the original KB directly, one first squeezes it to a summary of manageable size, and then performs the required operations on the summary itself. Common summarization techniques for relational data include relational factorization (Nickel et al., 2011; London et al., 2013; Riedel et al., 2013) and representation learning (Bordes et al., 2011; Socher et al., 2013). The core idea is to learn compressed latent representations, or *embeddings*, of entities and relations able to reconstruct the original KB by minimizing a suitable reconstruction loss. Until recently, relational embeddings have been mostly employed for link prediction and knowledge base completion (Nickel et al., 2016).

However, Yang et al. (2015) have shown that low-dimensional representations can also be exploited to perform a simplified form of theory learning. Their paper shows that, under reasonable assumptions, a simple nearest neighbor algorithm can recover logical rules directly from the *fixed size* embeddings of a KB, with potential runtime benefits. Furthermore, since the embeddings generalize beyond the observed facts, the rules are implicitly mined over a *completion* of the KB. Despite the novelty of their insight, their proposed method has several major downsides. First, their simple approach is limited to extracting rules as conjunctions of relations, with no support for logical disjunction and negation. Second, the rules are mined independently of one another, which can lead to redundant theories and compromise generalization ability and interpretability.

Building on the insights of Yang et al. (2015), we propose a novel approach to theory learning from low-dimensional representations. We view theory learning as a special *sparse recovery* problem. In this setting, a logical theory is merely an algebraic combination of embedded relations that best reconstructs the original KB, in a sense that will be made clear later. The recovery problem can be solved with specialized compressed sensing algorithms, such as Orthogonal Matching Pursuit (Pati et al., 1993) or variants thereof. Our approach offers two key advantages: it automatically models the inter-dependency between different rules, discouraging redundancy in the learned theory, and it supports all propositional logic connectives, i.e. conjunction, disjunction, and negation. Our empirical analysis indicates that our method can mine satisfactory theories in realistic KBs, demonstrating its ability to discover diverse and interpretable sets of rules. Additionally, our method can in principle be applied to "deeper" embeddings, that is, embeddings produced by deep models that take into consideration both relational and feature-level aspects of the data.

The paper is structured as follows. In the next section we introduce the required background material. We proceed by detailing our approach in Section 3 and evaluating it empirically in Section 4. We discuss relevant related work in Section 5, and conclude with some final remarks in Section 6.

2 BACKGROUND

In this section we briefly overview the required background. Let us start with the notation we will use. We write column vectors $\boldsymbol{x}$ in bold-face, matrices X in upper-case, and third-order tensors $\mathcal{X}$ in calligraphic upper-case. X^k is the kth frontal slice of the tensor $\mathcal{X}$, and **vec**(X) is the vectorization (flattening) of X. We denote the usual Frobenius matrix norm as $\|X\|_F := \sqrt{\sum_{ij} x_{ij}^2}$, the number of non-zero entries as $\|X\|_0$, the set $\{1, \ldots, n\}$ as $[n]$ and the Cartesian product of ℓ sets $\{1, \ldots, n\}$ as $[n]^\ell$. We reserve typewriter fonts for logical entities `Ann` and relations `motherOf`.

Knowledge Bases and Theories. A *knowledge base* (KB) is a collection of known true facts about a domain of interest. As an example, a KB about kinship relations may include facts such as $(\texttt{Ann}, \texttt{motherOf}, \texttt{Bob})$, which states that `Ann` is known to be the mother of `Bob`. In the following we will use n and m to denote the number of distinct entities and relations in the KB, respectively. With a slight abuse of notation, we will refer to logical constants and relations (e.g. `Ann` and `motherOf`) by their index in the KB ($e \in [n]$ or $r \in [m]$, respectively). Triples not occurring in the KB are unobserved, i.e. neither true nor false.

Given an input KB, the goal of theory learning, also known as Inductive Logic Programming (Muggleton et al., 1992), is to induce a compact logical *theory* that both explains the observed facts and generalizes to the unobserved ones. Most ILP methods extract theories in definite clausal form, which offers a good compromise between expressiveness and efficiency. A theory in this form is an implicitly conjoined set of Horn rules, i.e. rules like:

$$\forall\, e, e' \in [n] \;\; (e, \texttt{uncleOf}, e') \Leftarrow \exists\, e'' \in [n] \;\; (e, \texttt{brotherOf}, e'') \,\wedge\, (e'', \texttt{parentOf}, e')$$

Here $\Leftarrow$ represents logical entailment. The left-hand side is called the *head* of the rule, while the right-hand side is the *body*. The semantics of Horn rules are simple: whenever the body is satisfied by a given set of entities and relations, so is the head. The length of a rule is the number of relations appearing in its body; the above is a length 2 rule.

Classical ILP approaches cast theory learning as a search problem over the (exponentially large) space of candidate theories. When there are no negative facts, as in our case, the quality of a theory is given by the number of true facts it entails. In practice, learning is regularized by the size of the theory (number and length of the rules) to encourage compression, generalization and interpretability. Due to the combinatorial nature of the problem, the search task is solved heuristically, e.g. by searching individual Horn rules either independently or sequentially, or by optimizing surrogate objective functions. A language bias, provided by a domain expert, is often employed to guide the search toward more promising theories. Please see (Dzeroski & Lavrac, 1994; Muggleton et al., 1992) for more details.

Relational embeddings. Relational embedding techniques learn a low-dimensional latent representation of a KB. In order to ground the discussion, we focus on a prototypical factorization method, RESCAL (Nickel et al., 2011; 2012); many alternative formulations can be seen as variations or generalizations thereof. We stress, however, that our method can be applied to other kinds of relational embeddings, as sketched in Section 6. For a general treatment of the subject, see Nickel et al. (2016).

In RESCAL, each entity $e \in [n]$ in the KB is mapped to a vector $\boldsymbol{x}^e \in \mathbb{R}^d$, and each binary relation $r \in [m]$ to a matrix $W^r \in \mathbb{R}^{d \times d}$. These parameters are learned from data. Here $d \in [n]$ is a user-specified constant (the *rank*) controlling the amount of compression. The key idea is to model the plausibility, or *score*, of each fact as a function of its embedding. In particular, in RESCAL the score of a fact (e, r, e') is given by the bilinear product:

$$\text{score}(e, r, e') := (\boldsymbol{x}^e)^\top W^r \boldsymbol{x}^{e'} = \sum_{i=1}^{d} \sum_{j=1}^{d} \boldsymbol{x}_i^e W_{ij}^r \boldsymbol{x}_j^{e'}$$

The bilinear product measures how similar $\boldsymbol{x}^e$ and $W^r \boldsymbol{x}^{e'}$ are: the higher the dot product, the higher the score.

The embeddings can be expressed compactly in tensor form by grouping the entity vectors side-by-side into a matrix $X \in \mathbb{R}^{d \times n}$, and stacking the relation matrices into a tensor $\mathcal{W} \in \mathbb{R}^{d \times d \times m}$. The embeddings $(X, \mathcal{W})$ are learned so as to reconstruct the original KB as accurately as possible, modulo regularization. More formally, let $\mathcal{Y} \in \{0,1\}^{n \times n \times m}$ be a tensor such that $Y_{ee'}^r$ evaluates to 1 if the fact (e, r, e') appears in the KB, and to 0 otherwise. The learned embeddings should satisfy $Y_{ee'}^r \approx \text{score}(e, r, e')$ for all possible triples (e, r, e'). Learning equates to solving the optimization problem:

$$\min_{\mathcal{W}, X} \sum_{r=1}^{m} \|Y^r - X^\top W^r X\|_F^2 + \lambda \left(\|X\|_F^2 + \sum_{r=1}^{m} \|W^r\|_F^2 \right) \tag{1}$$

The second summand is a quadratic regularization term, whose impact is modulated by the $\lambda > 0$ hyperparameter. Note that the entity embeddings X are shared between relations. Choosing $d \ll n$ forces RESCAL to learn more compressed latent features, that hopefully better generalize over distinct facts, at the cost of a potentially larger reconstruction error. While the optimization problem is non-convex and can not be solved exactly in general, RESCAL pairs clever initialization with an alternating least squares procedure to obtain good quality solutions (Nickel et al., 2011).

In the next section we will see how theory learning can be generalized to work directly on the embeddings produced by RESCAL and analogous models.

3 Rule mining in feature space

In this section we detail our take on rule mining. Given a knowledge base in tensor form $\mathcal{Y}$, our goal is to learn a theory T that (1) entails many of the observed facts and few of the unobserved ones, and (2) is composed of few, diverse rules, for improved generalization.

The theory T includes rules for all possible relations $h \in [m]$, where the relation is the head of the rule and the body is an "explanation" of the relation as a (logical) combination of relations. Let T^h be the set of rules for head h. In our setting, T^h is a conjunction of Horn rules, where each rule is at most ℓ long [1]; ℓ is provided by the user. Following Yang et al. (2015), we require the rules to be *closed paths*, i.e. to be in the following form:

$$(e_1, h, e_{\ell+1}) \Leftarrow (e_1, b_1, e_2) \ \wedge \ (e_2, b_2, e_3) \ \wedge \ \ldots \ \wedge \ (e_\ell, b_\ell, e_{\ell+1}) \tag{2}$$

Here h is the head relation, and $b_1, \ldots, b_l$ are the body relations; quantifiers have been left implicit. Formally, a Horn rule is a closed path if (i) consecutive relations share the middle argument, and (ii) the left argument of the head appears as the first argument of the body (and conversely for the right argument). This special form enables us to cast theory learning in terms of Boolean matrix operations, as follows.

[1] For the sake of exposition, in the following we only consider rules *exactly* ℓ long; as a matter of fact, the miners we consider can return rules of length ℓ or shorter.

Let $\mathcal{Y}$ be a knowledge base and $h \in [m]$ the target head relation. Note that the conjunction of Horn rules with the same head relation h amounts to the disjunction of their bodies. Due to requirement (1), the set of rules targeting h should approximate the truth values of h, i.e.

$$Y^h \approx \bigvee_{B \in T^h} \bigwedge_{b \in B} Y^b$$

Here B is the body of a rule, and the logical connectives operate element-wise. In order to learn T from $\mathcal{Y}$, we define a loss function that encourages the above condition. We define the loss $\Delta(Y^h, T^h)$ as the accuracy of reconstruction of Y^h w.r.t. T^h, written as:

$$\Delta(Y^h, T^h) := \left\| Y^h \oplus \bigvee_{B \in T^h} \bigwedge_{b \in B} Y^b \right\|_0 \tag{3}$$

where $\oplus$ is the element-wise exclusive OR operator and $\| \cdot \|_0$ computes the misclassification error of T^h over Y^h. Minimizing Eq. (3) unfortunately is a hard combinatorial problem. We will next show how to approximate the latter as a continuous sparse reconstruction problem.

The relaxed reconstruction problem. Our goal is to approximate Eq. (3) in terms of algebraic matrix operations over the relation embeddings $\mathcal{W}$. First, we replace conjunctions with products between the embeddings of the relations along the path in the body of the rule, i.e.

$$\bigwedge_{b \in B} Y^b \approx X^\top \left(\prod_{b \in B} W^b \right) X$$

The idea is that a linear operator W^b maps the embedding of the left argument of relation b to vectors similar to the embedding of the right one, as per Eq. 1. For instance, $W^{\texttt{motherOf}}$ will map the embedding of `Ann` to a vector with high dot product w.r.t. the embedding of `Bob`. The closed path represented by the conjunction of the relations in the body B is emulated by composition of embeddings and obtained by repeated applications of this mapping (Yang et al., 2015).

Second, we replace disjunctions with sums:

$$Y^h \approx X^\top W^h X \approx X^\top \left[\sum_{B \in T^h} \prod_{b \in B} W^b \right] X$$

Intuitively, each path should represent an alternative explanation for the head relation, so that two entities are in relation h if at least one path (approximately) maps the left entity to the right one. Diversity between these alternatives will be enforced by imposing orthogonality between the mappings of the corresponding paths during the mining procedure, as explained later on in the section.

Clearly, the set of rules T^h is unknown and needs to be learned in solving the reconstruction problem. We thus let the summation run over all possible paths of length ℓ, i.e. $[m]^\ell$, adding a coefficient α^B for each candidate path. The problem boils down to learning these alphas:

$$\min_{\boldsymbol{\alpha}} \left\| X^\top W^h X - \sum_{B \in [m]^\ell} \alpha^B X^\top \prod_{b \in B} W^b X \right\|_F \tag{4}$$

In principle, the coefficients α^B should be zero-one; however, we relax them to be real-valued to obtain a tractable optimization problem. This choice has another beneficial side effect: the relaxed formulation gives us a straightforward way to introduce negations in formulas, thus augmenting the expressiveness of our approach beyond purely Horn clauses. The idea builds on the concept of set difference from set theory. A relation like `brotherOf` can be explained by the rule "a sibling who is not a sister". This could be represented in the space of the embeddings as the difference between the `siblingOf` mapping (accounting for both brothers and sisters) and the `sisterOf` one. More specifically, `siblingOf` $\wedge\neg$ `sisterOf` would be encoded as $W^{\texttt{siblingOf}} - W^{\texttt{sisterOf}}$. We thus allow α to also take negative values, with the interpretation that negative bodies are negated and conjoint (rather than disjoint) with the rest of the formula.

The last step is to get rid of the instances X, and mine rules for head h only in terms of their ability to reconstruct its embedding W^h. This is justified by the observation (Yang et al., 2015; Gu et al., 2015; Neelakantan et al., 2015; García-Durán et al., 2015) that the embeddings are learned so that their composition is close to that of the embedding of the head.

Putting everything together, we obtain an optimization problem of the form:

$$\min_{\boldsymbol{\alpha}} \left\| W^h - \sum_{B \in [m]^\ell} \alpha^B \prod_{b \in B} W^b \right\|_F \tag{5}$$

	# triples	**# entities**	**# relations**
Nations	3243	14	56
Kinship	10790	104	26
UMLS	6752	135	49
Family	5984	628	24

Table 1: Number of entities and relations of all datasets.

for each target head h. Upon finding the coefficients $\boldsymbol{\alpha}$, we convert them into a logic theory based on their sign and magnitude. First, only bodies with absolute coefficients larger than a threshold $\tau > 0$ are retained. Each body is then converted to the conjuction of the relations it contains. Bodies with positive coefficients are disjunctively combined with the rest of the formula, while bodies with negative coefficients are added as conjunctions of their negations. The final theory for the mined rule can be written as:

$$Y^h \approx \left(\bigvee_{B:\alpha^B>\tau} \bigwedge_{b\in B} Y^b\right) \wedge \neg \left(\bigvee_{B:\alpha^B<-\tau} \bigwedge_{b\in B} Y^b\right) \tag{6}$$

Solving the reconstruction problem. Equation 5 is a *matrix recovery* problem in Frobenius norm. Instead of solving it directly, we leverage the norm equivalence $\|A - B\|_F = \|\mathbf{vec}(A) - \mathbf{vec}(B)\|_2$ to reinterpret it as a simpler *vector recovery* problem. Most importantly, since most of the candidate paths B can not explain the head h, the recovery problem is typically *sparse*. Sparse recovery problems are a main subject of study in compressed sensing (Candès et al., 2006), and a multitude of algorithms can be employed to solve them, including Orthogonal Matching Pursuit (OMP) (Pati et al., 1993), Basis Pursuit (Chen et al., 1998), and many recent alternatives. In Appendix A we show how minimizing the sparse recovery problem in Eq. 5 equates to minimizing an upper bound of the total loss.

Two features of the above problem stand out. First, if the target theory is sparse enough, existing recovery algorithms can solve the reconstruction to global optimality with high probability (Candès et al., 2006). We do not explicitly leverage this perk; we leave finding conditions guaranteeing perfect theory recovery to future work. Second and most importantly, reconstruction algorithms choose the non-zero coefficients α^B so that the corresponding path embeddings $\prod_{b\in B} W^b$ are mutually *orthogonal*. This means that similar paths will not be mined together, thus encouraging rule diversity, as per requirement (2).

4 EMPIRICAL EVALUATION

We compare our method, dubbed Feature Rule Miner (FRM for short), against two variants of the kNN-based theory miner of Yang et al. (2015) on four publicly available knowledge bases: Nations, Kinship and UMLS from Kemp et al. (2006), and Family from Fang et al. (2013). The KB statistics can be found in Table 1. Given that FRM requires the relational embeddings $\mathcal{W}$ to be normalized (with respect to the Frobenius norm), we compare it against both the original kNN-based miner, which mines the unnormalized embeddings, and a variant that uses the normalized embeddings instead, for the sake of fairness.

The miners were tested in a 10-fold cross-validation setting. We computed the relational embeddings over the training sets using non-negative RESCAL (Krompaß et al., 2013) [2] variant with the default parameters (500 maximum iterations, convergence threshold 10^{-5}). The size of the embeddings d was set to a reasonable value for each KB: 100 for Family, 25 for Kinship and UMLS, and 5 for Nations. We configured all competitors to mine at most 100 rules for each head relation. The kNN distance threshold was set to 100 (although the actual value used is chosen dynamically, as done by Yang et al. (2015)). The desired reconstruction threshold of OMP was set ot 10^{-3}. Finally, the coefficient threshold τ was set to 0.2.

We evaluate both the *F-score* and the *per-rule recall* of all the methods. The F-score measures how well the mined rules reconstruct the test facts in terms of both precision and recall. The per-rule recall is simply the recall over the number of rules mined for the target head; it favors methods that

[2] Standard RESCAL tends to penalize the kNN-based competitors.

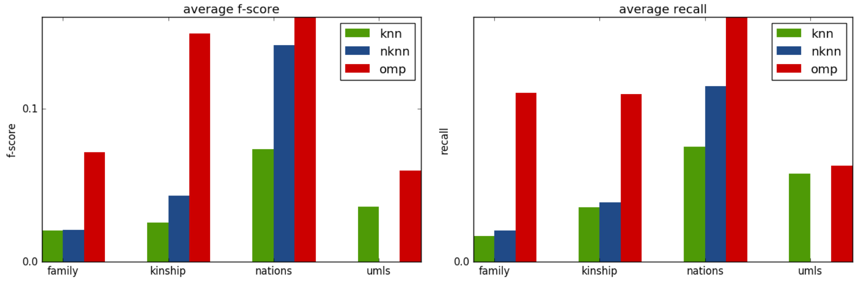

Figure 1: Results of all methods on the four datasets for max rule length 2. Average F-score is reported on the left, average recall over number of rules on the right.

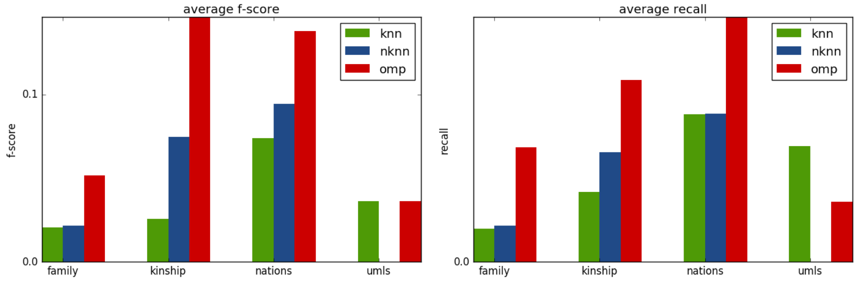

Figure 2: Results of all methods on the four datasets for max rule length 3. Average F-score is reported on the left, average recall over number of rules on the right.

focus on few rules with high coverage, and penalizes those that mine many irrelevant rules. The results on the four KBs (averaged over all target relations) are reported in Figures 1 and 2, and an example of mined rules in Figure 3. More detailed per-head results can be found in Appendix B. (Unfortunately, the normalized kNN method failed to work with the UMLS dataset; we left a blank in the plots.)

The plots show a clear trend: FRM performs better than the kNN-based methods in all four knowledge bases, both in terms of F-score and in terms of per-rule recall. Further, the normalized kNN variant tends to outperform the original, unnormalized version, providing support for our use of normalized relation embeddings.

Notably, the three methods mine similar amounts of rules. While OMP stops automatically when the mined body reconstructs the target head sufficiently well, the kNN methods compensate for the lack of a proper termination criterion by employing a distance-based pruning heuristic (as discussed by Yang et al. (2015)). Rather, the poor per-rule recall performance of the kNN methods can be imputed to insufficient rule diversity. The kNN miners discover the rules independently of each other, leading to theory redundancy. This is a well known problem in rule mining. On the contrary, OMP avoids this issue by enforcing orthogonality between the mined bodies. The resulting theories performs much better especially in terms of per-rule recall.

The phenomenon is also visible in Figure 3. The theory found by FRM contains many diverse bodies, while the one found by kNN does not. The two rules also show the power of negation: the FRM theory includes the “perfect” definition of a brother, i.e. `siblingOf` $\wedge$ $\neg$`sisterOf` (as well as an obvious error, i.e. that a brother can not be a sibling of a sibling). In contrast the theory found by kNN completely ignores the complementarity of `brotherOf` and `sisterOf`, and includes the rule `brotherOf` $\Leftarrow$ `sisterOf`.

$$\begin{aligned} \texttt{brotherOf} \quad &\Leftarrow \quad (\texttt{siblingOf} \lor (\texttt{siblingOf} \land \texttt{brotherOf}) \lor (\texttt{siblingOf} \land \texttt{sisterOf})) \\ & \quad\;\; \land \; \neg(\texttt{sisterOf} \lor (\texttt{siblingOf} \land \texttt{siblingOf})) \end{aligned}$$

$$\begin{aligned} \texttt{brotherOf} \quad &\Leftarrow \quad \texttt{siblingOf} \lor (\texttt{siblingOf} \land \texttt{siblingOf}) \lor (\texttt{siblingOf} \land \texttt{brotherOf}) \\ & \quad\;\; \lor \; (\texttt{childOf} \land \texttt{parentOf}) \lor (\texttt{sonOf} \land \texttt{parentOf}) \\ & \quad\;\; \lor \; \texttt{sisterOf} \lor (\texttt{siblingOf} \land \texttt{sisterOf}) \end{aligned}$$

Figure 3: Example rules for the `brotherOf` relation mined by FRM (top) and kNN (bottom).

5 Related work

There is a huge body of work on theory learning, historically studied in Inductive Logic Programming (Dzeroski & Lavrac, 1994; Muggleton et al., 1992). For the sake of brevity, we focus on techniques that are more closely related to our proposal.

The core of most ILP methods, e.g. FOIL (Quinlan, 1990), Progol (Muggleton, 1995), and Aleph[3], is a search loop over the space of candidate theories. Bottom-up methods start from an initially empty theory, and add one Horn rule at a time. Individual rules are constructed by conjoining first-order relations so as to maximize the number of covered positive facts, while trying to keep covered negative facts to a minimum. After each rule is constructed, all covered facts are removed from the KB. These methods are extremely expressive, and can handle general $n$ary relations. Instead, FRM focuses on binary relations only, which are more common in today's Web-centric knowledge bases. ILP methods are designed to operate on the original KB only; this fact, paired with the sheer magnitude of the search space, makes standard ILP methods highly non-scalable. More recent extensions (e.g. kFOIL (Landwehr et al., 2006)) adopt a feature-space view of relational facts, but are still based on the classical search loop and can not be trivially adapted to working on the relational embeddings directly. Finally, rule elongation can be hindered by the presence of plateaus in the cost function.

Our path-based learning procedure is closely related to Relational Pathfinding (RP) (Richards & Mooney, 1992). RP is based on the observation that ground relation paths (that is, conjunctions of true relation instances) do act as support for arbitrary-length rules. It follows that mining these paths directly allows to detect longer rules with high support, avoiding the rule elongation problem entirely. There are many commonalities between RP and FRM. Both approaches are centered around relation paths, although in different representations (original versus compressed), and focus on path-based theories. The major drawback of RP is that it requires exhaustive enumeration of relation paths (up to a maximum length), which can be impractical depending on the size of the KB. FRM sidesteps this issue by leveraging efficient online decoding techniques, namely Online Search OMP (Weinstein & Wakin, 2012).

To alleviate its computational requirements, a lifting procedure for RP was presented in Kok & Domingos (2009). Similarly to FRM, lifted RP is composed of separate compression and learning stages. In the first stage, the original KB is "lifted" by clustering functionally identical relation paths together, producing a smaller KB as output. In the second stage, standard RP is applied to the compressed KB. A major difference with FRM is that lifting is exact, while RESCAL is typically lossy. Consequently, lifted RP guarantees equivalence of the original and compressed learning problems, but it also ignores the potential generalization benefit provided by the embeddings. Additionally, the first step of lifted RP relies on a (rather complex) agglomerative clustering procedure, while FRM can make use of state-of-the-art representation learning methods. Note that, just like lifted RP, FRM can be straightforwardly employed for structure learning of statistical relational models.

The work of Malioutov & Varshney (2013) is concerned with mining one-level rules from binary data. Like in FRM, rule learning is viewed as a recovery problem, and solved using compressed sensing techniques. Two major differences with FRM exist. In Malioutov & Varshney (2013) the truth value matrix is recovered with an extension of Basis Pursuit that handles 0-1 coefficients through

[3] `http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/`

a mixed-integer linear programming (MILP) formulation, which is however solved approximately using linear relaxations. BP however requires the dictionary to be explicitly grounded, which is not the case for FRM. Additionally, their method is limited to one-level rules, i.e. either conjunctions or disjunctions of relations, but not both. An extension to two-level rules has been presented by Su et al. (2015), where BP is combined with heuristics to aggregate individual rules into two-level theories. In contrast, FRM natively supports mining two-level rules via efficient online search.

The only other theory learning method that is explicitly designed for working on embeddings is the one of Yang et al. (2015). It is based on the observation (also made by Gu et al. (2015)) that closed path Horn rules can be converted to path queries, which can be answered approximately by searching the space of (type-compatible) compositions of relation embeddings. They propose to perform a simple nearest neighbor search around the embedding of the head relation, W^h, while avoiding type-incompatible relation compositions. Unfortunately, rules are searched for independently of one another, which seriously affects both quality and interpretability of the results as shown by our experimental evaluation.

6 Conclusion

We presented a novel approach for performing rule mining directly over a compressed summary of a KB. A major advantage over purely logical alternatives is that the relational embeddings automatically generalize beyond the observed facts; as consequence, our method implicitly mines a completion of the knowledge base. The key idea is that theory learning can be approximated by a recovery problem in the space of relation embeddings, which can be solved efficiently using well-known sparse recovery algorithms. This novel formulation enables our method to deal with all propositional logic connectives (conjunction, disjunction, and negation), unlike previous techniques. We presented experimental results highlighting the ability of our miner to discover relevant and, most importantly, diverse rules.

One difficulty in applying our methods is that classical sparse recovery algorithm require the complete enumeration of the candidate rule bodies, which is exponential in rule length. In order to solve this issue, we plan to apply recent *online* recovery algorithms, like Online Search OMP (Weinstein & Wakin, 2012), which can explore the space of alternative bodies on-the-fly.

As the quality of relational embedding techniques improves, for instance thanks to path-based Gu et al. (2015); Neelakantan et al. (2015); García-Durán et al. (2015) and logic-based Rocktäschel et al. (2015) training techniques, we expect the reliability and performance of theory learning in feature space to substantially improve as well.

References

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *Dbpedia: A nucleus for a web of open data*. 2007.

A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*, 2011.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

S.S. Chen, David L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

Sašo Dzeroski and Nada Lavrac. *Inductive logic programming: Techniques and applications*. 1994.

Sašo Dzeroski and Nada Lavrac (eds.). *Relational Data Mining*, New York, NY, USA, 2000. Springer-Verlag New York, Inc.

R. Fang, A. Gallagher, T. Chen, and A. Loui. Kinship classification by modeling facial feature heredity. In *Proceedings of ICIP*, pp. 2983–2987, 2013.

L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal*, 24(6):707–730, 2015.

A. García-Durán, A. Bordes, and N. Usunier. Composing relationships with translations. In *Proceedings of EMNLP*, pp. 286–290, 2015.

K. Gu, J. Miller, and P. Liang. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*, 2015.

J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of AAAI*, volume 3, pp. 5, 2006.

S. Kok and P. Domingos. Learning markov logic network structure via hypergraph lifting. In *Proceedings of ICML*, pp. 505–512, 2009.

Denis Krompaß, Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Non-negative tensor factorization with rescal. In *Tensor Methods for Machine Learning, ECML workshop*, 2013.

N. Landwehr, A. Passerini, L. De Raedt, and P. Frasconi. kfoil: Learning simple relational kernels. In *AAAI*, volume 6, pp. 389–394, 2006.

B. London, T. Rekatsinas, B. Huang, and L. Getoor. Multi-relational learning using weighted tensor decomposition with modular loss. *arXiv preprint arXiv:1303.1733*, 2013.

D. Malioutov and K. Varshney. Exact rule learning via boolean compressed sensing. In *Proceedings of ICML*, pp. 765–773, 2013.

S. Muggleton. Inverse entailment and progol. *New generation computing*, 13(3-4):245–286, 1995.

Stephen Muggleton, Ramon Otero, and Alireza Tamaddoni-Nezhad. *Inductive logic programming*, volume 168. 1992.

A. Neelakantan, B. Roth, and A. McCallum. Compositional vector space models for knowledge base completion. *arXiv preprint arXiv:1504.06662*, 2015.

M. Nickel, V. Tresp, and H-P Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*, pp. 809–816, 2011.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of WWW*, pp. 271–280, 2012.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

Y. Chandra Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of Asilomar Conference on Signals, Systems and Computers, 1993*, pp. 40–44, 1993.

J. R. Quinlan. Learning logical definitions from relations. *Machine learning*, 5(3):239–266, 1990.

B.L. Richards and R.J. Mooney. Learning relations by pathfinding. In *Proceedings of AAAI*, pp. 50–55, 1992.

S. Riedel, L. Yao, A. McCallum, and B. M. Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pp. 74–84, 2013.

T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of NAACL HTL*, 2015.

R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*, pp. 926–934, 2013.

G. Su, D. Wei, K.R. Varshney, and D.M. Malioutov. Interpretable two-level boolean rule learning for classification. *arXiv preprint arXiv:1511.07361*, 2015.

A. J. Weinstein and M. B. Wakin. Online search orthogonal matching pursuit. In *Proceedings of IEEE SSP Workshop*, pp. 584–587, 2012.

B. Yang, W. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR*, 2015.

Appendix A: Error Derivation

Fix a target head relation h. Let E^h denote the RESCAL error matrix $E^h := Y^h - X^\top W^h X$, and $\tilde{E}^h$ denote the error matrix of W^h due to FRM, namely $\tilde{E}^h := W^h - \sum_{B \in T^h} \alpha^B W^B$ where $W^B = \prod_{b \in B} W^b$. Putting the two definitions together, we obtain:

$$E^h = Y^h - X^\top \left[\sum_{B \in T^h} W^B + \tilde{E}^h \right] X = Y^h - X^\top \left[\sum_{B \in T^h} W^B \right] X - X^\top \tilde{E}^h X$$

Then, the Frobenius norm of the reconstruction error of head h is:

$$\|Y^h - X^\top \sum_{B \in T^h} \alpha_B W^B X\| = \|X^\top \tilde{E}^h X + E^h\| \leq \|X^\top \tilde{E}^h X\| + \|E^h\| \leq \|X\|^2 \|\tilde{E}^h\| + \|E^h\|$$

where the last step follows from the sub-multiplicativity of the Frobenius norm. Now, FRM minimizes $\|\tilde{E}^h\|$, and therefore minimizes an upper bound of the misclassification error of T^h over Y^h.

We note in passing that the bound can be tightened by reducing the norm of the entity embeddings X, for instance by choosing the proper embedding method. The question of how to find an optimal choice, however, is left as future work.

Appendix B: Extended Results

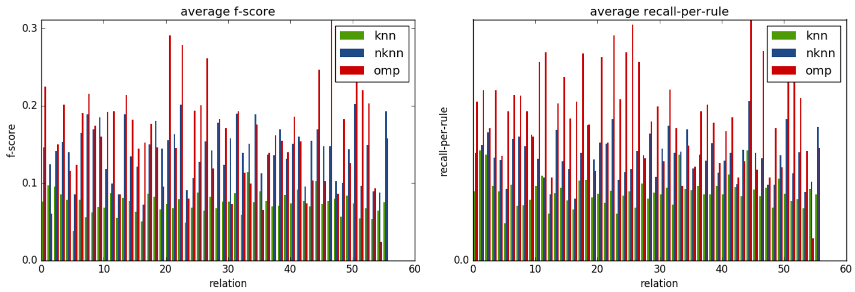

Figure 4: Detailed results for the nations KB with length 2 rules.

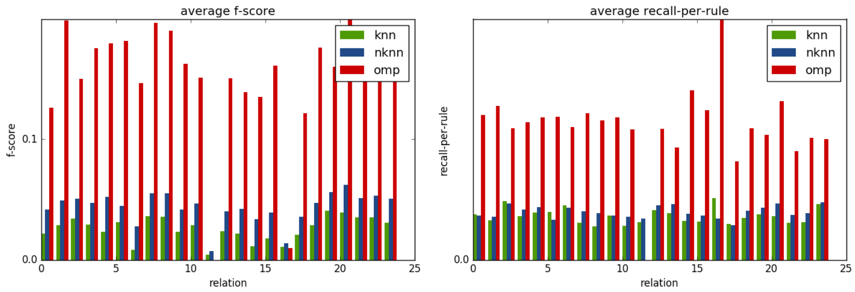

Figure 5: Detailed results for the kinship KB with length 2 rules.

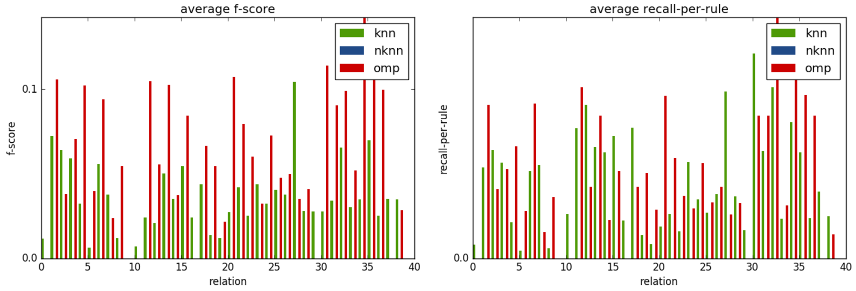

Figure 6: Detailed results for the UMLS KB with length 2 rules.

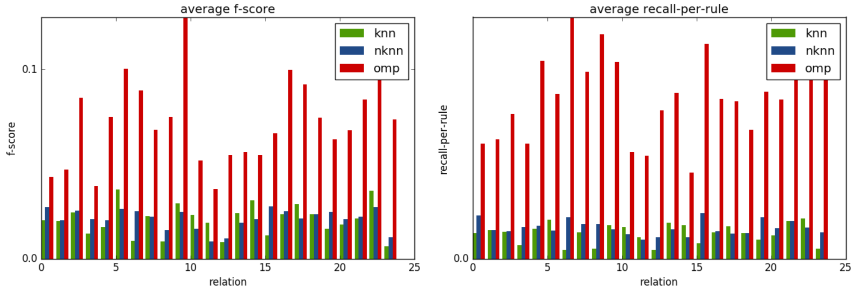

Figure 7: Detailed results for the family KB with length 2 rules.

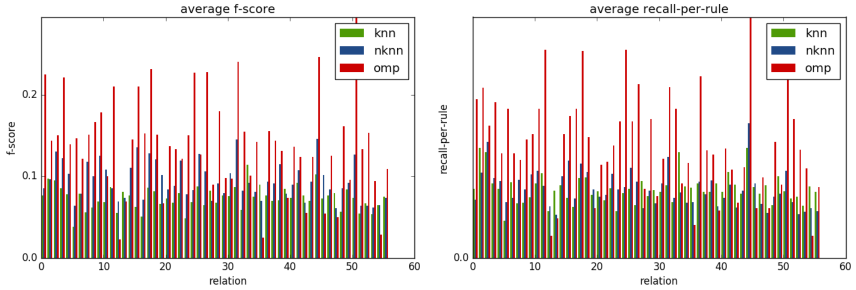

Figure 8: Detailed results for the nations KB with length 3 rules.

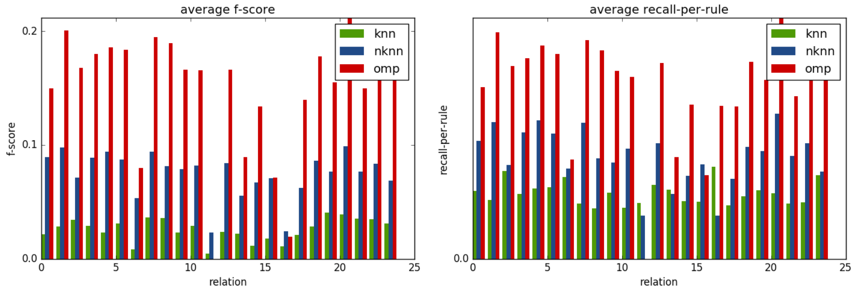

Figure 9: Detailed results for the kinship KB with length 3 rules.

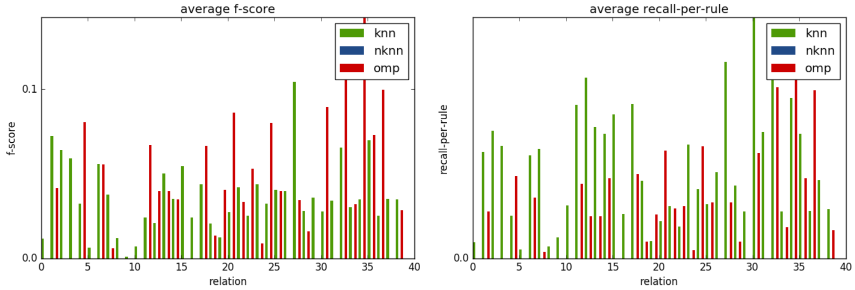

Figure 10: Detailed results for the UMLS KB with length 3 rules.

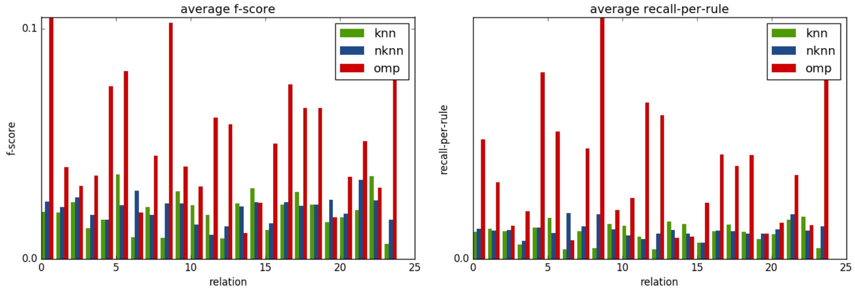

Figure 11: Detailed results for the family KB with length 3 rules.