Abducing Relations in Continuous Spaces

Taisuke Sato[1], **Katsumi Inoue**[2], **Chiaki Sakama**[3]
[1] AI research center AIST, Japan
[2] National Institute of Informatics, Japan
[3] Wakayama University, Japan
satou.taisuke@aist.go.jp, inoue@nii.ac.jp, sakama@sys.wakayama-u.ac.jp

Abstract

We propose a new approach to abduction, i.e., non-deductive inference to find a hypothesis H for an observation O such that $H, KB \vdash O$ where KB is background knowledge. We reformulate it linear algebraically in vector spaces to abduce *relations*, not logical formulas, to realize approximate but scalable abduction that can deal with web-scale knowledge bases. More specifically we consider the problem of abducing relations for Datalog programs with binary predicates. We treat two cases, the non-recursive case and the recursive case. In the non-recursive case, given $r_1(X,Y)$ and $r_3(X,Z)$, we abduce $r_2(Y,Z)$ so that $r_3(X,Z) \Leftrightarrow \exists Y r_1(X,Y) \wedge r_2(Y,Z)$ approximately holds, by computing a matrix $\mathbf{R}_2$ that approximately satisfies a matrix equation $\mathbf{R}_3 = \min_1(\mathbf{R}_1\mathbf{R}_2)$ containing a nonlinear function $\min_1(x)$. Here $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ encode as adjacency matrix $r_1(X,Y)$, $r_2(Y,Z)$ and $r_3(Y,Z)$ respectively. We apply this matrix-based abduction to rule discovery and relation discovery in a knowledge graph. The recursive case is mathematically more involved and computationally more difficult but solvable by deriving a recursive matrix equation and solving it. We illustrate concrete recursive cases including a transitive closure relation.

1 Introduction

Traditionally logical inference in AI has been conducted symbolically and low-level data processing such as image recognition is carried out non-symbolically using real numbers and vectors. However recent emergence of big knowledge graphs [Nickel *et al.*, 2016] such as Freebase [Bollacker *et al.*, 2008] and Knowledge Vault [Dong *et al.*, 2014], where a proposition $r(i,j)$ saying that a subject i and an object j stands in the relation r is represented as a triple (i,r,j), has spurred the development of scalable linear algebraic technology for relational inference that is applicable to logical inference as well. For example, given a domain of size n, we enumerate entities in the domain and represent the i-th entity $(1 \leq i \leq n)$ by $n \times 1$ one-hot vector $\mathbf{v}_i = (0, \cdots, 1, 0 \cdots)^T$ such that only the i-th entry is one. Also the relation r is encoded as an adjacency matrix $\mathbf{R}$ such that $\mathbf{R}_{ij} = 1$ if $r(i,j)$ is true and $\mathbf{R}_{ij} = 0$ otherwise. The truth value (1 if true, else 0) of $r(i,j)$ is computed by multiplying $\mathbf{v}_i$, $\mathbf{v}_j$ and $\mathbf{R}$ like $\mathbf{v}_i^T \mathbf{R} \mathbf{v}_j \in \{1,0\}$. We note that starting from this simple setting, a full-fledged formulation of first order logic is developed in tensor spaces where predicates and quantifications are represented by tensors [Sato, 2017b]. Also by applying such an algebraic formulation to deductive inference, i.e., Datalog evaluation, it is possible to achieve, for example, 10^1 times to 10^4 times faster computation than symbolic approaches for some class of Datalog programs [Sato, 2017a]. Recently it is reported that the well-known logic programming semantics such as the least model semantics and the stable model semantics are rigorously formalized in tensor spaces [Sakama *et al.*, 2017]. These results clearly demonstrate the potential of "doing logic in continuous spaces" and suggest further exploration of the linear algebraic approach in logical inference beyond deductive one.

In this paper, we propose a new approach to abduction, i.e., non-deductive inference to find a hypothesis H for an observation O such that $H, KB \vdash O$ where KB is background knowledge. We reformulate it linear algebraically in vector spaces as "*linear algebraic abduction*" to abduce *relations*, not logical formulas, to realize approximate but scalable abduction that can deal with web-scale knowledge bases. More specifically we consider the problem of abducing relations for Datalog programs with binary predicates. We treat two cases, the non-recursive case and the recursive case. In the non-recursive case, given $r_1(X,Y)$ and $r_3(X,Z)$[1], we abduce $r_2(Y,Z)$ so that $r_3(X,Z) \Leftrightarrow \exists Y r_1(X,Y) \wedge r_2(Y,Z)$ approximately holds, by computing $\mathbf{R}_2$ that approximately satisfies a matrix equation $\mathbf{R}_3 = \min_1(\mathbf{R}_1\mathbf{R}_2)$ containing a nonlinear function $\min_1(x)$. Here $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ encode as adjacency matrix $r_1(X,Y)$, $r_2(Y,Z)$ and $r_3(Y,Z)$ respectively. We apply this matrix-based abduction to rule discovery and relation discovery in a knowledge graph. The recursive case is mathematically more involved and computationally more difficult but solvable by deriving a recursive matrix equation and solving it. We illustrate concrete recursive cases including a transitive closure relation. Our contribution thus includes a proposal of innovative linear algebraic approach to abduction and the establishment of its technical basis.

[1] We follow Prolog convention. So strings beginning with a capital letter are variables and others are constants or predicate symbols.

2 Preliminaries

2.1 Abduction in Datalog

We first briefly explain abduction in Datalog [Kakas *et al.*, 1992; Eiter *et al.*, 1997; Gottlob *et al.*, 2010]. A Datalog program $KB = F \cup R$ is a finite set of clauses without function symbols where F is a set of ground atoms called facts and R a set of rules, i.e., definite clauses. The least Herbrand model $\mathbf{M}_{KB}$ is the set of ground atoms provable from KB. We assume KB has only binary predicates and each clause in R is of the form $r_0(X_0,X_n) \Leftarrow r_1(X_0,X_1) \wedge \cdots \wedge r_n(X_{n-1},X_n)$. We also assume for simplicity that recursive programs have only one recursive goal in their clause body.

Logically, deduction in Datalog can be done by SLD deduction. Think of three predicates, nationality(X,Z) (X's nationality is Z), live_in(X,Y) (X lives in Y) and located_in(Y,Z) (Y is located in Z) together with a program $KB_0 = F_0 \cup R_0$ where $F_0 = \{$live_in(spielberg,california), located_in(california,usa)$\}$ and $R_0 = \{$nationality$(X,Z) \Leftarrow$ live_in$(X,Y) \wedge$ located_in$(Y,Z)\}$. Datalog proves nationality(spielberg,usa) as a top-level query from KB_0 using SLD deduction.

Now consider abduction. Unlike deduction, abduction is invoked when a set F' of facts is not enough to deduce observations. Suppose we have a set of special atoms called *abducibles* as possible hypothesis. When we have found that $F',R \nvdash o$ for an observed ground atom o, we hypothesize, or *abduce*, a set $\{a_1,\ldots,a_m\}$ of abducibles and add them to F' so that $\{a_1,\ldots,a_m\} \cup F', R \vdash o$ holds. Suppose for example we have $F'_0 = \{$live_in(spielberg,california)$\}$ and observe $o =$ nationality(spielberg,usa). We note that $F'_0 \cup R_0 \nvdash o$. So we attempt to retain the conclusion by abducing a hypothesis $a =$ located_in(california,usa) so that we have $\{a\} \cup F'_0, R_0 \vdash o$. In Datalog, abduction can be realized by extending SLD deduction to return selected abducibles with which a top-level goal is proved. In our approach however, we abduce not some selected abducibles but the extension of an *abducible predicate* that represents a hypothesized/missing relation.

2.2 Matrix Compilation for Deductive Inference

Here we recap how a program KB is compiled to a set of matrix equations that compute the least model $\mathbf{M}_{KB}$ [Sato, 2017a]. Without loss of generality, it is assumed that the domain of size n consists of integers i ($1 \leq i \leq n$). For concreteness, suppose KB contains ground atoms about live_in/2 and located_in/2 and one clause nationality$(X,Z) \Leftarrow$ live_in$(X,Y) \wedge$ located_in(Y,Z).

In compilation, first ground atoms are compiled to matrices; ground atoms about live_in/2 and located_in/2 are respectively compiled to $n \times n$ adjacency matrices $\mathbf{L}_i$ and $\mathbf{L}_o$ such that $(\mathbf{L}_i)_{ij} = 1$ if live_in$(i,j) \in KB$ and $(\mathbf{L}_i)_{ij} = 0$ otherwise, etc. Then the definite clause nationality$(X,Z) \Leftarrow$ live_in$(X,Y) \wedge$ located_in(Y,Z) is compiled, in which a conjunction in the clause body is compiled to a matrix product. Thus the clause body live_in$(X,Y) \wedge$ located_in(Y,Z) is compiled to $\mathbf{v}_X^T \min_1(\mathbf{L}_i\mathbf{L}_o)\mathbf{v}_Z$ where $\mathbf{v}_X$ designates an indefinite one-hot vector corresponding to the variable X. Here $\min_1(x)$ denotes a nonlinear thresholding function defined by $\min_1(x) = 1$ if $x \geq 1$, and $\min_1(x) = x$ otherwise. $\min_1(\mathbf{A})$ for matrix $\mathbf{A}$ means a componentwise application of $\min_1$.

The clause head nationality(X,Z) is compiled to $\mathbf{v}_X^T \mathbf{N}_a \mathbf{v}_Z$ where $\mathbf{N}_a$ is the adjacency matrix representing nationality/2. Finally, recalling that nationality$(X,Z) \Leftrightarrow \exists Y$live_in$(X,Y) \wedge$ located_in(Y,Z) holds in $\mathbf{M}_{KB}$ for any X and Z, and hence so does $\mathbf{v}_X^T \mathbf{N}_a \mathbf{v}_Z = \mathbf{v}_X^T \min_1(\mathbf{L}_i\mathbf{L}_o)\mathbf{v}_Z$ for any one-hot vectors $\mathbf{v}_X$ and $\mathbf{v}_Z$, we reach the following matrix equation as compilation output:

$$\mathbf{N}_a \quad = \quad \min_1(\mathbf{L}_i\mathbf{L}_o). \tag{1}$$

This equation should hold for any $\mathbf{N}_a$, $\mathbf{L}_i$ and $\mathbf{L}_o$ encoding nationality/2, live_in/2 and located_in/2 in $\mathbf{M}_{KB}$. We can use it for bottom-up computation of $\mathbf{M}_{KB}$ in such a way that we evaluate the clause body first, obtaining $\mathbf{L}_i$ and $\mathbf{L}_o$, then compute $\mathbf{N}_a$ for the head predicate using (1).

2.3 Recursive Programs

Now consider recursive programs. For illustration, we take up the following program (2) which computes the transitive closure $r_2(X,Z)$ of a base relation $r_1(X,Y)$. We can compile this program to a matrix equation with recursion using the method of [Sato, 2017a] as follows.

$$\begin{aligned} r_2(X,Y) &\Leftarrow r_1(X,Y) \\ r_2(X,Z) &\Leftarrow r_1(X,Y) \wedge r_2(Y,Z) \end{aligned} \tag{2}$$

The compilation procedure compiles each clause body, then combining the output by sum and $\min_1$, produces the following nonlinear recursive equation:

$$\mathbf{R}_2 \quad = \quad \min_1(\mathbf{R}_1 + \mathbf{R}_1\mathbf{R}_2) \tag{3}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are $n \times n$ adjacency matrices respectively encoding $r_1(X,Z)$ and $r_2(X,Z)$ in the least model. Although the matrix equation (3) holds true in the least model and solving (3) for $\mathbf{R}_2$ assuming $\mathbf{R}_1$ gives the transitive closure $r_2(X,Y)$ as an adjacency matrix, the nonlinearity of $\min_1$ blocks the application of linear algebraic operations to solve it. We circumvent this difficulty by linearizing (3) to

$$\tilde{\mathbf{R}}_2 \quad = \quad \varepsilon(\mathbf{R}_1 + \mathbf{R}_1\tilde{\mathbf{R}}_2) \tag{4}$$

where ε is a small positive number such that $(\mathbf{I} - \varepsilon\mathbf{R}_1)^{-1}$ exists, for example $\varepsilon = \frac{1}{1+\|\mathbf{R}_1\|_\infty}$ [2]. This equation (4) is easy to solve and gives $\tilde{\mathbf{R}}_2 = (\mathbf{I} - \varepsilon\mathbf{R}_1)^{-1}\varepsilon\mathbf{R}_1$. It is proved that the thresholded matrix $(\tilde{\mathbf{R}}_2)_{>0}$ coincides with the (least) solution of equation (3), also denoted by $\mathbf{R}_2$ [Sato, 2017a].

$$\mathbf{R}_2 \quad = \quad (\tilde{\mathbf{R}}_2)_{>0} \tag{5}$$

Here $(\tilde{\mathbf{R}}_2)_{>0}$ is a matrix obtained by thresholding each entry of $\tilde{\mathbf{R}}_2$ at 0, i.e., replacing the entry $(\tilde{\mathbf{R}}_2)_{ij}$ by 1 if $(\tilde{\mathbf{R}}_2)_{ij} > 0$, else by 0. This way of computing transitive closure has time complexity $O(n^3)$ [3] and empirically outperforms conventional symbolic approaches to Datalog evaluation when $\mathbf{R}_1$ is non-sparse [Sato, 2017a].

Matrix compilation which produces matrix equations like (1) and (3) can be extended to general Datalog programs but we omit it. In what follows, we explain our new approach to abduction in detail. We first deal with the non-recursive case.

[2] $\|\mathbf{A}\|_\infty$ is a matrix norm defined by $\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}|$.

[3] Or less, $O(n^{2.376})$, if the Coppersmith-Winograd algorithm [Coppersmith and Winograd, 1990] is used.

3 Non-recursive Abduction

3.1 Abducing Adjacency Matrices

Our abduction approximately abduces relations as adjacency matrices by solving matrix equations. Suppose we have two relations nationality(X,Z) and live_in(X,Y) as $n \times n$ adjacency matrices $\mathbf{N}_a$ and $\mathbf{L}_i$ respectively together with a non-recursive clause nationality$(X,Z) \Leftarrow$ live_in$(X,Y) \wedge$ located_in(Y,Z), and seek to abduce located_in(Y,Z) such that nationality$(X,Z) \Leftrightarrow \exists Y$ live_in$(X,Y) \wedge$ located_in(Y,Z) as a missing relation. This abduction problem is equivalent to abducing an adjacency matrix $\mathbf{L}_o$ for located_in(Y,Z) that satisfies a matrix equation $\mathbf{N}_a = \min_1(\mathbf{L}_i\mathbf{L}_o)$.

$\mathbf{N}_a = \min_1(\mathbf{L}_i\mathbf{L}_o)$ would be exactly solvable for $\mathbf{L}_o$ by reformulating it to a SAT problem at the cost of seeking the truth value of n^2 Boolean variables that encode $\mathbf{L}_o$. However as we are thinking of applications to knowledge graphs where n easily goes up to 10^4 or higher, solving a SAT problem of having 10^8 Boolean variables seems not feasible as of now. Hence we approximately solve the problem in a vector space and obtain $\mathbf{L}_o$ such that $\mathbf{N}_a \approx \min_1(\mathbf{L}_i\mathbf{L}_o)$ but how? We first simply solve $\mathbf{N}_a = \mathbf{L}_i\mathbf{X}$ for $\mathbf{X}$ and put $\mathbf{L}_o = \mathbf{X}_{>\theta}$ for some appropriate θ where $\mathbf{X}_{>\theta}$ denotes thresholding at θ of each entry x in $\mathbf{X}$ such that x is replaced by 1 if $x > \theta$, else by 0. The difference between $\mathbf{N}_a$ and $\min_1(\mathbf{L}_i\mathbf{X}_{>\theta})$ will be minimized by adjusting θ. Formally we state

Problem definition:(relation abduction in vector spaces) Given adjacency matrices $\mathbf{R}_1$ and $\mathbf{R}_3$, our abduction problem is to find $\mathbf{R}_2$ such that $\mathbf{R}_3 \approx \min_1(\mathbf{R}_1\mathbf{R}_2)$.

We solve this problem by computing a matrix $\mathbf{X}$ that minimizes $\mathbf{J}$:

$$\mathbf{J} \;=\; \|\mathbf{R}_3 - \mathbf{R}_1\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_F^2 \qquad (6)$$

where $\lambda > 0$, $\|\cdot\|_F$ denotes Frobenius norm and $\lambda\|\mathbf{X}\|_F^2$ is a penalty term controlling the Frobenius norm of $\mathbf{X}$.

Theorem: The minimizer of $\mathbf{J}$ is given by

$$\mathbf{X} \;=\; (\lambda\mathbf{I} + \mathbf{R}_1^T\mathbf{R}_1)^{-1}(\mathbf{R}_1^T\mathbf{R}_3) \qquad (7)$$

where $\mathbf{I}$ is an identity matrix.
Sketch of proof: The partial derivatives of the objective function $\mathbf{J}$ w.r.t. $\mathbf{X}_{ij}$s set to zero give a matrix equation $(\lambda\mathbf{I} + \mathbf{R}_1^T\mathbf{R}_1)\mathbf{X} = \mathbf{R}_1^T\mathbf{R}_3$. Since $\lambda\mathbf{I} + \mathbf{R}_1^T\mathbf{R}_1$ is positive definite for $\lambda > 0$, it has an inverse thereby yielding (7).

The error is computed as $\delta = \mathbf{R}_3 - \mathbf{R}_1\mathbf{X} = \mathbf{P}\mathbf{R}_3$ where $\mathbf{P}$ approaches the orthogonal projection onto the orthogonal complement of *range*$(\mathbf{R}_1)$ when $\lambda \to 0$. Accordingly if $\mathbf{R}_3$ is factored into $\mathbf{R}_3 = \mathbf{R}_1\mathbf{Y}$ for some matrix $\mathbf{Y}$, $\delta = \mathbf{P}\mathbf{R}_3 = \mathbf{P}\mathbf{R}_1\mathbf{Y} = \mathbf{0}$ (all 0 matrix) when $\lambda \to 0$, and hence we have $\mathbf{R}_3 = \mathbf{R}_1\mathbf{X}$ (but unfortunately, $\mathbf{X}$ is not yet an adjacency matrix, so we need thresholding to obtain $\mathbf{R}_2$).

After computing $\mathbf{X}$ by (7), we abduce $\mathbf{R}_2$ as $\mathbf{R}_2 = \mathbf{X}_{>\theta}$ while adjusting the threshold θ so that $\mathbf{R}_3$ is best approximated by $\min_1(\mathbf{R}_1\mathbf{R}_2)$. We evaluate the proximity of $\min_1(\mathbf{R}_1\mathbf{R}_2)$ to $\mathbf{R}_3$ in terms of their F-measure[4] where we consider an adjacency matrix $\mathbf{R}$ as a set $\{(i,j) \mid \mathbf{R}_{ij} = 1\}$. The time complexity of this abduction, computing $\mathbf{R}_2$ and $\min_1(\mathbf{R}_1\mathbf{R}_2)$ from $\mathbf{R}_3$ and $\mathbf{R}_1$, is $O(n^3)$. We next test our linear algebraic approach using artificial data and real data[5].

3.2 Experiment with Random Graphs

Here we conduct an experiment for non-recursive abduction with artificial data. We use adjacency matrices expressing directed random graphs D(n,p_e) with n vertices such that an edge occurs independently with probability p_e. In the experiment, we set $n = 10^4$, create two $n \times n$ random adjacency matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ encoding two D(n,p_e)s and construct $\mathbf{R}_3 = \min_1(\mathbf{R}_1\mathbf{R}_2)$. Note that $\mathbf{R}_3$ is not a pure random matrix representing some D(n,p_e) but one with internal structure in the form of matrix production. The purpose of this experiment is to identify this internal structure by abduction; we compute $\mathbf{X}$ by (7) and put $\mathbf{R}_{2\_\mathrm{abd}} = \mathbf{X}_{>\theta}$, and examine to what extent $\mathbf{R}_3 = \min_1(\mathbf{R}_1\mathbf{R}_{2\_\mathrm{abd}})$ holds. Or logically speaking, given $r_3(X,Z)$ and $r_1(X,Y)$, we examine whether $r_3(X,Z)$ can be written as an existential conjunction $\exists Y r_1(X,Y) \wedge r_2(Y,Z)$ or not.

θ is adjusted as follows; we equally divide the space between the largest and smallest values of $\mathbf{X}$ into 50 discrete steps, test each step as a thresholding value and choose one achieving the best F-measure for $\mathbf{R}_3$ and $\min_1(\mathbf{R}_1\mathbf{R}_{2\_\mathrm{abd}})$. We evaluate the quality of abduction in two ways: one by the absolute error Δ_{error}, i.e., the number of different entries between $\mathbf{R}_3$ and $\min_1(\mathbf{R}_1\mathbf{R}_{2\_\mathrm{abd}})$[6] and the other their F-measure. We repeat this process for $\lambda = 1$ and $p_e \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. The results are summarized in Table 1 where $|\mathbf{R}_3|$ denotes the number of 1s in $\mathbf{R}_3$ etc and "time" means execution time. We can make several observations. First of all, recall that random matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ encoding D(n,p_e) where $n = 10^4$ have an entry 1 with probability p_e. It follows by calculation that an entry in $\min_1(\mathbf{R}_1\mathbf{R}_2)$ is 1 with probability $1-(1-p_e^2)^n$ and $\mathbf{R}_3$ is expected to have $n^2(1-(1-p_e^2)^n)$ 1s on average. The column $|\mathbf{R}_3|$ of Table 1 supports this estimation well.

Second, F-measure is high for all values of p_e, which means if the observed relation $r_3(X,Z)$ is describable as $\exists Y r_1(X,Y) \wedge r_2(Y,Z)$, we may have a good chance of reconstructing it if $r_1(X,Z)$ is given. Also note that the absolute error Δ_{error} is small or even zero for sparse matrices where $p_e \leq 10^{-3}$. Besides, $|\mathbf{R}_{2\_\mathrm{abd}}| < |\mathbf{R}_2|$ holds for all p_es and the difference widens as $\mathbf{R}_3$ becomes sparse. In other words, $\mathbf{R}_3$, if sparse, is approximated well using $\mathbf{R}_{2\_\mathrm{abd}}$ which is smaller than the original $\mathbf{R}_2$. This is particularly important when applying to knowledge graphs as real relations tend to be very sparse.

Third, computation is done in reasonable time despite the fact that we are dealing with $10^4 \times 10^4$ sized matrices. This

[4]The F-measure for two sets A and B is defined as $F(A,B) = 2|A \cap B|/(|A|+|B|)$ where $|A|$ denotes the cardinality of A. $0 \leq F(A,B) \leq 1$ and $F(A,B) = 1$ if-and-only-if $A = B$. Hence, the higher, the better.

[5]All experiments in this paper are carried out by GNU Octave 4.0.0 on a PC with Intel(R) Core(TM) i7-3770@3.40GHz CPU, 28GB memory.

[6]Δ_{error} is the number of unequal entries of $\mathbf{R}_3$ and $\min_1(\mathbf{R}_1\mathbf{R}_{2\_\mathrm{abd}})$.

p_e	$\lvert\mathbf{R}_3\rvert$	$\lvert\mathbf{R}_1\rvert$	$\lvert\mathbf{R}_2\rvert$	$\lvert\mathbf{R}_{2\_\text{abd}}\rvert$	Δ_{error}	F-measure	time(sec)
10^{-2}	63,209,013	999,390	1,000,216	998,936	3,434,829	0.947	1597.2
10^{-3}	996,254	99,924	100,190	99,923	3	1.000	1629.8
10^{-4}	9,772	10,003	9,818	7,501	342	0.966	1690.8
10^{-5}	104	1,035	1,041	107	0	1.000	1673.6

Table 1: Abduction with random adjacency matrices

may be due to multi-threaded matrix operations on a multi-core processor. Also notably, computation time remains almost constant irrespective of p_e. We therefore may say that our linear algebraic abduction is applicable to considerably large relations.

3.3 Rule Discovery and Relation Discovery for Knowledge Graph

Now we deal with real data. We use FB15k, a standard knowledge graph comprised of triples (subject, relation, object) in RDF format for 1,345 binary relations and 14,951 entities [Bordes *et al.*, 2013]. Let $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}_3$ be matrices encoding relations $r_1(X,Y)$, $r_2(Y,Z)$ and $r_3(X,Z)$ in FB15k respectively. For later use, we define F-measure of $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_2(Y,Z)$ as the F-measure for $\min_1(\mathbf{R}_1\mathbf{R}_2)$ and $\mathbf{R}_3$.

We tackle two problems at once. One is rule discovery that finds a rule of the form $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_2(Y,Z)$ with a high F-measure. The other is relation discovery that seeks for an *unknown relation* $r_{\text{unknown}}(Y,Z)$, not existent in the original database but connecting known relations $r_3(X,Z)$ and $r_1(X,Y)$ by a rule

$$r_3(X,Z) \quad \Leftarrow \quad r_1(X,Y) \wedge r_{\text{unknown}}(Y,Z). \tag{8}$$

A naive approach to the first problem, rule discovery, would be to enumerate all possible triples $(r_1(X,Y), r_2(Y,Z), r_3(X,Z))$ of relations in FB15k and check if $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_2(Y,Z)$ holds well. However since FB15k contains 1,345 relations, it is unrealistic to test their $1,345^3 \approx 2.4 \times 10^9$ combinations. We therefore focus on relations on a specific domain such as person and film. Furthermore we choose specifically top-ten largest relations (in the number of triplets) in the domain and search for $(r_1(X,Y), r_2(Y,Z), r_3(X,Z))$ where $r_1(X,Y)$ and $r_3(X,Z)$ are two of the top-ten relations but $r_2(Y,Z)$ comes from the entire FB15k relations, hoping that we may be able to find interesting rules that reveal unknown connections among existing relations in the domain.

For the second problem, relation discovery, we simply abduce $r_{2\_\text{abd}}(Y,Z)$ as an unknown relation so that a rule $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_{2\_\text{abd}}(Y,Z)$ achieves a high F-measure. Fortunately by first abducing $r_{2\_\text{abd}}(Y,Z)$ and reusing it for rule discovery, we can find rules and new relations simultaneously over the specified domain.

More concretely, we do the following for all pairs of matrices $(\mathbf{R}_3, \mathbf{R}_1)$, each $14,951 \times 14,951$ matrix, for the top-ten relations in the domain[7]. First abduce $\mathbf{R}_{2\_\text{abd}}$ such that $\mathbf{R}_3 \approx \min_1(\mathbf{R}_1\mathbf{R}_{2\_\text{abd}})$. If F-measure of $\mathbf{R}_3$ and $\min_1(\mathbf{R}_1\mathbf{R}_{2\_\text{abd}})$ is less than 0.5, do nothing. Otherwise, search for $\mathbf{R}_{2\_\text{sim}}$ from the entire 1,345 relations in FB15k which is most similar to $\mathbf{R}_{2\_\text{abd}}$ in terms of cosine similarity[8]. Return $r_{2\_\text{abd}}(Y,Z)$, the binary relation represented by $\mathbf{R}_{2\_\text{abd}}$, as an abduced relation and $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_{2\_\text{sim}}(Y,Z)$ as a discovered rule for the pair $(\mathbf{R}_3, \mathbf{R}_1)$. Since additional work to take cosine similarity is $O(n^2)$ for $n \times n$ matrices, the time complexity to compute $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_{2\_\text{sim}}(Y,Z)$ remains $O(n^3)$.

We list two of the rules and new relations we discovered[9]. Given nationality(X,Z) and live_in(X,Y) in a domain concerning people in FB15k, we discovered

$$\text{nationality}(X,Z) \quad \Leftarrow \quad \text{live\_in}(X,Y) \wedge \text{located\_in}(Y,Z).$$

A typical instance of this rule is nationality(spielberg, usa) $\Leftarrow$ live_in(spielberg, cincinnati) $\wedge$ located_in(cincinnati, usa). Here all three relations are existent in FB15k. The quality of the rule, the F-measure for $\mathbf{N}_a$ and $\min_1(\mathbf{L}_i\mathbf{L}_o)$ is 0.415 where $\mathbf{N}_a$, $\mathbf{L}_i$ and $\mathbf{L}_o$ respectively encode nationality(X,Z), live_in(X,Y) and located_in(Y,Z) in the knowledge graph. Comparing to the F-measure 0.007 for $\mathbf{N}_a$ and $\mathbf{L}_i$, this value is much higher, which means nationality(X,Z) is much better predicted from live_in$(X,Y) \wedge$ located_in(Y,Z) than from live_in(X,Y) alone. Although we have discovered other interesting rules with high F-measures, they are omitted for space limitations.

Concerning relation discovery, *genre_lang*(Y,Z) in the clause below is an example of the abduced relations in a domain related to film.

$$\text{language}(X,Z) \quad \Leftarrow \quad \text{genre}(X,Y) \wedge \textit{genre\_lang}(Y,Z)$$

Here language(X,Z) and genre(X,Y) are existing relations but *genre_lang*(Y,Z) is an abduced one, not existent in FB15k. A plausible instance by known entities of this rule is language(the_lord_of_the_rings, english) $\Leftarrow$ genre(the_lord_of_the_rings, adventure) $\wedge$ *genre_lang*(adventure, english). The abduced relation *genre_lang*(Y,Z) teaches us how genre(X,Y) is possibly

[7] The triples comprising a relation in FB15k are divided into training, validation and test sets [Bordes *et al.*, 2013]. We use training sets for rule and relation discovery.

[8] The cosine similarity of matrices $\mathbf{A}$ and $\mathbf{B}$ is defined as $(\sum_{ij}\mathbf{A}_{ij}\mathbf{B}_{ij})(\|\mathbf{A}\|_F\|\mathbf{B}\|_F)^{-1}$.

[9] We assign familiar names to relations in FB15k for intuitiveness.

connected to language(X,Z). Actually, while the F-measure of a rule language$(X,Y) \Leftarrow$ genre(X,Y) that predicts language(X,Y) from genre(X,Y) alone is zero, the F-measure of the above rule is 0.655, a reasonably high value, which exemplifies the effectiveness of combining abduced relations and existing relations to predict target relations.

4 Recursive Abduction for Transitive Closure

So far we have only been dealing with the non-recursive case where we abduce $r_2(Y,Z)$ for a non-recursive clause $r_3(X,Z) \Leftarrow r_1(X,Y) \wedge r_2(Y,Z)$. However, our approach is applicable to the recursive case as well. Look at the equation $\mathbf{R}_2 = \min_1(\mathbf{R}_1 + \mathbf{R}_1\mathbf{R}_2)$ (3) for transitive closure compiled from a recursive Datalog program (2) that computes the transitive closure $r_2(X,Y)$ of a base relation $r_1(X,Y)$, where matrices $\mathbf{R}_1$ and $\mathbf{R}_2$ encodes respectively $r_2(X,Y)$ and $r_1(X,Y)$. Then our abduction problem is stated as follows.

Problem definition:(recursive abduction in vector spaces) Given $\mathbf{R}_2$, find a base relation $\mathbf{R}_1$ such that $\mathbf{R}_2 \approx \min_1(\mathbf{R}_1 + \mathbf{R}_1\mathbf{R}_2)$.

We abduce the base relation $\mathbf{R}_1$ from $\mathbf{R}_2$ in two steps. First we solve (9)

$$\mathbf{R}_2 \quad = \quad \mathbf{X} + \mathbf{X}\mathbf{R}_2 \qquad (9)$$

for $\mathbf{X}$ and obtain $\mathbf{X} = \mathbf{R}_2(\mathbf{I} + \mathbf{R}_2)^{-1}$. Then we threshold $\mathbf{X}$ at some θ and return $\mathbf{R}_{1\_\mathrm{abd}} = \mathbf{X}_{>\theta}$ as an abduced base relation such that $\mathbf{R}_2 \approx \min_1(\mathbf{R}_{1\_\mathrm{abd}} + \mathbf{R}_{1\_\mathrm{abd}}\mathbf{R}_2)$.

We conduct an experiment with artificial data to test the above approach. We use directed random graphs D(n,p_e) similarly to the previous experiment. We set $n = 10^4$ and $\theta = 10^{-4}$. Given p_e, we generate an $n \times n$ random matrix $\mathbf{R}_1$ encoding D(n,p_e) and compute its transitive closure $\mathbf{R}_2$. Then we abduce $\mathbf{R}_{1\_\mathrm{abd}}$ such that $\mathbf{R}_2 \approx \min_1(\mathbf{R}_{1\_\mathrm{abd}} + \mathbf{R}_{1\_\mathrm{abd}}\mathbf{R}_2)$ and evaluate the quality of this abduction in terms of Δ_{error} = the number of different entries between $\mathbf{R}_2$ and $\min_1(\mathbf{R}_{1\_\mathrm{abd}} + \mathbf{R}_{1\_\mathrm{abd}}\mathbf{R}_2)$. For each $p_e \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, we repeat the above process five times and take the average of $|\mathbf{R}_2|$, $|\mathbf{R}_1|$, $|\mathbf{R}_{1\_\mathrm{abd}}|$, Δ_{error} and computation time. The results are listed in Table 2.

Seeing Table 2, we know $\Delta_{error} = 0$, i.e. $\mathbf{R}_2 = \min_1(\mathbf{R}_{1\_\mathrm{abd}} + \mathbf{R}_{1\_\mathrm{abd}}\mathbf{R}_2)$ holds exactly for all cases, so we may say our abduction is exact under the setting of this experiment. In particular for $p_e = 10^{-5}$, 10^{-6} when matrices are sparse, during the experiment for all runs, we observed that $\mathbf{R}_{1\_\mathrm{abd}} = \mathbf{R}_1$ (hence $|\mathbf{R}_{1\_\mathrm{abd}}| = |\mathbf{R}_1|$ in the table) and $\mathbf{R}_{1\_\mathrm{abd}}$ is a transitive reduction of $\mathbf{R}_1$[10]. I.e., in addition to $\mathbf{R}_3$, the base relation $\mathbf{R}_1$ is also correctly recovered by the linear algebraic abduction for $p_e = 10^{-5}$, 10^{-6}. This is a bit surprising considering the simplicity of our approach. However when p_e increases to 10^{-4}, the gap between $\mathbf{R}_{1\_\mathrm{abd}}$ and $\mathbf{R}_1$ appears and at $p_e = 10^{-3}$, abduction apparently fails as $|\mathbf{R}_{1\_\mathrm{abd}}| = |\mathbf{R}_2|$ strongly suggests we abduced $\mathbf{R}_2$ itself for all runs[11]. The lesson we can draw from this experiment is that the linear algebraic abduction is likely to work well if matrices are sparse even though there is recursion.

We then conduct a similar experiment with real data. We choose five network graphs from the Koblenz Network Collection [Kunegis, 2013] and apply the linear algebraic abduction. For each network $\mathbf{R}_1$ (as $n \times n$ adjacency matrix), we compute its transitive closure $\mathbf{R}_2$ and abduce $\mathbf{R}_{1\_\mathrm{abd}}$ so that $\mathbf{R}_2 \approx \min_1(\mathbf{R}_{1\_\mathrm{abd}} + \mathbf{R}_{1\_\mathrm{abd}}\mathbf{R}_2)$ holds. Table 3 contains the results from the experiment.

We observe that networks vary in size and some networks are large ($n > 20{,}000$) but abduction is done in reasonable time. Also the transitive closure relations $\mathbf{R}_2$ are well-approximated by the transitive closure of $\mathbf{R}_{1\_\mathrm{abd}}$[12]. Furthermore $|\mathbf{R}_{1\_\mathrm{abd}}| < |\mathbf{R}_1|$ holds for the reactome network, which means $\mathbf{R}_{1\_\mathrm{abd}}$ gives the same transitive closure but more compactly. We thus remark that the linear algebraic abduction can be a useful tool for approximately finding base relations for transitive closure relations from real data.

5 Related Work

There are not many papers concerning logical inference in vector spaces. Ceri and Tanca formulated relational equations in relational algebra to compute the least model semantics of Datalog and analyzed bottom-up and top-down execution methods. However relations are not expressed nor manipulated as matrices [Ceri *et al.*, 1989]. Ioannidis and Wong gave an algebraic formalization of recursive Horn programs using the theory of closed semiring. They proved that processing recursive clauses are reduced to solving recursive operator equations [Ioannidis and Wong, 1991]. Lin formulated resolution linear algebraically and studied a linear algebraic treatment of SAT problems [Lin, 2013]. Grefenstette reconstructed quantifier free first-order logic in tensor spaces. Quantification is also considered but nested one is excluded [Grefenstette, 2013]. Sato proposed tensorization of full first-order logic with arbitrary quantification [Sato, 2017b]. Sakama et al. formalized logic programming semantics of various types and their computation using tensors [Sakama *et al.*, 2017]. Sato described a linear algebraic formulation of Datalog and demonstrated that faster computation can be achieved by compiling programs to matrix equations [Sato, 2017a]. However, it considers only deduction, and realizing abduction in matrix computation has been left open.

In the field of knowledge graph, linear algebraic techniques in vector spaces such as low-dimensional embedding to process a huge number of triplets in a knowledge graph have been developed [Bordes *et al.*, 2013; Yang *et al.*, 2015; Nickel *et al.*, 2016; Rocktäschel *et al.*, 2015; García-Durán *et al.*, 2016; Trouillon *et al.*, 2016]. Entities are compactly represented as low-dimensional vectors and binary relations are represented either by vectors or by matrices. Their learning from real data has been a central issue while logical inference is attempted to assist extracting new knowledge [Rocktäschel *et al.*, 2015] or simulated in the form of link traversal in

[10]For adjacency matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A}$ is said to be a transitive reduction of $\mathbf{B}$ if both have the same transitive closure and $\mathbf{A}$ is minimal as a graph.

[11]$\mathbf{R}_2 = \min_1(\mathbf{R}_2\mathbf{R}_2)$ always holds as $\mathbf{R}_2$ is a transitive closure.

[12]The absolute error Δ_{error} for subelj-cora looks large but the relative error is $6618944/23166^2$ = 1.2%.

p_e	$\lvert\mathbf{R}_2\rvert$	$\lvert\mathbf{R}_1\rvert$	$\lvert\mathbf{R}_{1\_\mathrm{abd}}\rvert$	Δ_{error}	time(sec)
10^{-3}	99,992,000.0	100,110.0	99,992,000.0	0	40.3
10^{-4}	313,433.4	9,972.8	11,814.0	0	41.2
10^{-5}	1,092.2	982.2	982.2	0	43.4
10^{-6}	100.4	99.8	99.8	0	43.9

(all figures are averages over five runs)

Table 2: Abduction through transitive closure of directed random graphs

Network	n	$\lvert\mathbf{R}_2\rvert$	$\lvert\mathbf{R}_1\rvert$	$\lvert\mathbf{R}_{1\_\mathrm{abd}}\rvert$	Δ_{error}	F-measure	time(sec)
ego-twitter	23,370	353,882	33,101	172,870	0	1.000	559.7
subelj-cora	23,166	93,584,386	91,500	43,521,487	6,618,944	0.929	847.4
dblp-cite	12,591	7,583,575	49,743	1,439,337	0	1.000	83.7
reactome	6,327	4,744,333	147,547	99,912	65,380	0.986	11.2
moreno-blogs	1,224	982,061	19,025	962,118	0	1.000	0.1

Table 3: Abduction with network graphs from Koblenz Network Collection

the graph. Yang et al. attempted rule discovery in the low-dimensional embedding space. They basically enumerate all combinations of relations existent in the knowledge graph as the clause body unlike our approach that combines relations using relation discovery [Yang *et al.*, 2015]. Widdows and Cohen studied the vector representation of propositions and apply it to analogical reasoning. They represent entities and binary relations in terms of sparse vectors of various types (binary, real, complex) and show how to compose and decompose propositions by specifically designed linear algebraic operations called binding and release [Widdows and Cohen, 2015]. These works however pay attention primarily to deduction and neither abduction nor relation discovery is treated.

Abduction has been extensively studied in logic programming [Kakas *et al.*, 1992; Eiter *et al.*, 1997; Gottlob *et al.*, 2010]. Compared to the conventional approach, our abduction differs vastly in that first it is computed in vector spaces, second it abduces not formulas but relations in the form of adjacency matrices, and third relations are obtained approximately by solving matrix equations.

These features bring several advantages. Since vector spaces are continuous and so is inference, our abduction, though approximate, is expected to be robust to impreciseness and noise, complementary to exact but brittle symbolic abduction. Also a broad range of linear algebraic operations are available in vector spaces which are difficult or impossible to perform in symbolic approaches. They include inner product, projection, a various types of decomposition and low-dimensional embedding to name a few. Even recursion is implemented using matrix inversion. Furthermore their computation is highly scalable with modern computer technology such as multi-core CPUs.

6 Conclusion

We proposed an innovative approach, linear algebraic abduction, to abductive inference in Datalog, which approximately abduces relations in the form of adjacency matrices. Our formulation is based on the derivation of a matrix equation from a program. The equation reflects the least model of the program and abduction is performed by solving it w.r.t. an unknown matrix that represents the adjacency matrix for the missing relation to be abduced. The equation is non-recursive or recursive depending on whether the original program is non-recursive or recursive. We empirically showed that the linear algebraic abduction applied to the non-recursive case works well for sparse matrices representing directed random graphs. We also applied it to a knowledge graph and simultaneously discovered rules and new relations, which seems unprecedented.

For the recursive case, we conducted several experiments with the transitive closure relation where base relations are abduced for these relations as adjacency matrices. We observed that our approach can derive base relations which give relations identical to or close to the original relations.

In this paper, we outlined a new form of abduction in vector spaces but there remains a lot to be done. In particular, we need to expand the class of abducible programs to those with non-binary predicates by generalizing matrices to tensors for example based on the theoretical framework laid out by [Sato, 2017b] where a ternary predicated is represented by a 3rd order 0-1 tensor $\{\mathbf{R}_{ijk}\}$. Also we need to find a way to abduce not just one relation but many alternatives, which might be achieved by seeking multiple solutions when solving a matrix equation.

Acknowledgments

This research is supported in part by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO) and also by NII Collaborative Research.

References

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of data*, pages 1247–1250, 2008.

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2787–2795. 2013.

[Ceri *et al.*, 1989] Stefano Ceri, Georg Gottlob, and Letizia Tanca. What you always wanted to know about datalog (and never dared to ask). *IEEE Transactions on Knowledge and Data Engineering*, 1(1):146–166, 1989.

[Coppersmith and Winograd, 1990] Don Coppersmith and Shmuel Winograd. Matrix Multiplication via Arithmetic Progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[Dong *et al.*, 2014] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD2014*, pages 601–610, 2014.

[Eiter *et al.*, 1997] Thomas Eiter, Georg Gottlob, and Nicola Leone. Abduction from Logic Programs: Semantics and Complexity. *Theoretical Computer Science*, 189(1-2):129–177, 1997.

[García-Durán *et al.*, 2016] Alberto García-Durán, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research (JAIR)*, 55:715–742, 2016.

[Gottlob *et al.*, 2010] Georg Gottlob, Reinhard Pichler, and Fang Wei. Tractable Database Design and Datalog Abduction Through Bounded Treewidth. *Information Systems*, 35(3):278–298, 2010.

[Grefenstette, 2013] Edward Grefenstette. Towards a Formal Distributional Semantics: Simulating Logical Calculi with Tensors. *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, pages 1–10, 2013.

[Ioannidis and Wong, 1991] Yannis E. Ioannidis and Eugene Wong. Towards an algebraic theory of recursion. *J. ACM*, 38(2):329–381, April 1991.

[Kakas *et al.*, 1992] Antonis C. Kakas, Robert Kowalski, and Francesca Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.

[Kunegis, 2013] Jérôme Kunegis. KONECT - The Koblenz Network Collection. In *Proceedings of International Web Observatory Workshop*, pages 1343–1350, 2013.

[Lin, 2013] Fangzhen Lin. From satisfiability to linear algebra. Technical report, Hong Kong University of Science and Technology, 2013.

[Nickel *et al.*, 2016] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs: From Multi-Relational Link Prediction to Automated Knowledge Graph Construction. *Proceedings of the IEEE*, 104(1):11–33, 2016.

[Rocktäschel *et al.*, 2015] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015.

[Sakama *et al.*, 2017] Chiaki Sakama, Katsumi Inoue, and Taisuke Sato. Linear Algebraic Characterization of Logic Programs. In *Proceedings of the 10th International Conference on Knowledge Science, Engineering and Management (KSEM2017), LNAI 10412, Springer-Verlag*, pages 520–533, 2017.

[Sato, 2017a] Taisuke Sato. A linear algebraic approach to Datalog evaluation. *Theory and Practice of Logic Programming (TPLP)*, 17(3):244–265, 2017.

[Sato, 2017b] Taisuke Sato. Embedding Tarskian semantics in vector spaces. In *AAAI-17 Workshop on Symbolic Inference and Optimization (SymInfOpt-17)*, 2017.

[Trouillon *et al.*, 2016] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *Proceedings of the 33rd International Conference on Machine Learning (ICML' 2016)*, pages 2071–2080, 2016.

[Widdows and Cohen, 2015] D. Widdows and T. Cohen. Reasoning with Vectors: A Continuous Model for Fast Robust Inference. *Logic journal of the IGPL / Interest Group in Pure and Applied Logics*, 23(2):141–173, 2015.

[Yang *et al.*, 2015] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.