Relationalization: a framework for reconstructing structures from propositional descriptions

Filip Železný

Czech Technical University in Prague
Technická 2, 166 27, Prague 6, Czech Republic
zelezny@fel.cvut.cz

Abstract. I redefine propositionalization as an optimization problem where one seeks to select a prescribed number of relational features allowing for the most accurate reverse construction of structures ('relationalization') from their flattened representations, with respect to some similarity measure on first-order objects. This framework is independent of the particular kind of propositionalization technique one employs and of classification labels possibly cast on the data. The descriptive emphasis of the selected features makes them akin to *principle components* known from linear spaces, except that they maintain their instant interpretability. Unlike in linear principle component analysis, I am not able to identify the optimal solution in a closed form, but can obtain a suboptimal one through a simple gradient descent algorithm. To demonstrate my approach, I show a simple experiment with the East West Trains dataset using the 'extended transformation' technique of propositionalization.

1 Introduction

Traditional approaches to propositionalization through relational feature construction are biased towards the objective of predictive classification of the propositionalized data. As a consequence, the constructed features are required to capture some important, however isolated, properties whose verification contributes well to discriminatory power of classifiers.

Here I am trying to propose what appears to be a more general account of the problem. My goal is to generate a feature set allowing the flattened data form to simply be a good *representation* of the originally structured data, ie. to carry as much as possible descriptive information regardless of any data categorization.

Having some suitable propositionalization algorithm, the goal above in turn requires to introduce two further concepts:

- a *relationalization* procedure to 'reinvent' structures from their propositionalized descriptions.
- a structural *similarity measure* to evaluate the accuracy of the reverse construction and, in turn, the descriptive-representation quality of the employed feature set.

Informally, the technical goal is to select a feature set making relationalization ideally the inverse operator to propositionalization. However, the relationalization procedure is in a way an "inductive" task: it is basically search for a structure complying to given propositions (usually about the truth values of some relational assertions), while many such structures may exist. In logic terms, the main difference from the usual ILP task is that one searches here for a first-order interpretation rather than a theory. As for the structural similarity measure, I follow here on the relatively large body of research on first-order metrics conducted in ILP.

In a bird's eye view, the feature selection task entailed by the approach outlined above is similar to principal component analysis known from statistics (given the descriptive accuracy optimization goal). The fundamental difference lies in that one gets no closed-form solution to the involved optimization problem in the discrete-structure domain and exhaustive search for the optimal feature set (from some predefined pool of features) is intractable. A simple experiment however reveals that even a suboptimal solution obtained through a greedy feature selection algorithm brings significant benefits: given a required average similarity between the original and reconstructed data instances, one needs to employ about 30 times fewer greedily selected features than it would be needed if features were drawn randomly.

In the next section I outline a general framework for relationalization and optimal feature selection. Section 3 instantiates the framework using a specific propositionalization technique known as the extended transformation approach [4]. In Section 4 I show an experiment with the East-West Trains data set and give a URL reference to the Prolog implementation of the method. In Section 5 I discuss the main open issues and then conclude.

2 A General Framework

Definition 1. *Let S denote a set called* structures*, any $E \subseteq S$ will be called an* example set*. Let D denote the set called the* attribute domain*, $\mathcal{F}$ be a set called the* feature pool *whenever a unique function $f : S \rightarrow D$ is assigned to each element of $\mathcal{F}$. Each element of $\mathcal{F}$ is called a* feature*. By F I will denote some space of vectors with n ($n \in \mathcal{N}$) components from $\mathcal{F}$, ie. $F \subseteq \mathcal{F}^n$. Any $\boldsymbol{f} \in F$ is called a* feature vector*.*

I will use the same character for a feature and for the function assigned to it. The components of any vector $\boldsymbol{v}$ will be denoted as v_1, v_2 etc.

Definition 2. *The vector function $\boldsymbol{\pi} : F \times S \rightarrow D^n$ such that $\boldsymbol{\pi}(\boldsymbol{f}, s) = f_1(s), f_2(s), \ldots f_n(s)$ is called* propositionalization*. Any function $\rho : F \times D^n \rightarrow S$ such that $\boldsymbol{\pi}[\boldsymbol{f}, \rho(\boldsymbol{f}, \boldsymbol{d})] = \boldsymbol{d}$ for any $\boldsymbol{f} \in F, \boldsymbol{d} \in D$ is called* relationalization*. If S is countable (uncountable, respectively), any probability distribution (probability density function, respectively) R on S acquiring non-zero values only on the set*

$$C = \{s \in S | \boldsymbol{\pi}(\boldsymbol{f}, s) = \boldsymbol{d}\} \tag{1}$$

is called stochastic relationalization.

Although it is hard to imagine any useful uncountable spaces of structures, they are covered in the definition for sakes of generality.

Let me now discuss some elementary approaches through which relationalization could be achieved. For deterministic relationalization, one has to select a single $\rho(\boldsymbol{f}, \boldsymbol{d}) := s^*$ from a 'candidate' set coinciding with C in Eq. 1. Assuming there is some complexity measure on structures

$$size : S \rightarrow \mathcal{N} \tag{2}$$

a straightforward "Occam razor driven" choice is

$$s^* = \arg\min_{s \in C} size(s) \tag{3}$$

For a stochastic relationalization, one could either sample uniformly from C (if it is finite), or again install some bias, eg. towards simpler structures. In the general case the selection might be governed by the density

$$R(s) = \frac{e^{-size(s)}}{\int_{x \in C} e^{-size(x)}} \tag{4}$$

Of course the above recipes make practical sense only if the selection rules can be algorithmically implemented for a given structure set S and feature pool $\mathcal{F}$. To this end, I turn attention to a specific case in the next section.

The following definition captures a property one would naturally expect from a relationalization function: the order in which features (with their respective truth values) are presented to it should not be important.

Definition 3. *A relationalization ρ is said to be* order-invariant *if $\rho(\boldsymbol{f}, \boldsymbol{d}) = \rho[\mathcal{P}(\boldsymbol{f}), \mathcal{P}(\boldsymbol{d})]$ for any $\boldsymbol{f} \in F, \boldsymbol{d} \in D$ and an arbitrary permutation $\mathcal{P}$ on vector components.*

Although it seems straightforward that there actually exists some relationalization that is not order-invariant, it remains open to prove this.[1]

I will now shift focus onto properties of *features* in the context of relationalization. For sake of formal completeness, the following definition describes a 'golden-standard' feature vector whose utilization would allow to exactly reconstruct all propositionalized structures.

Definition 4. *For an example set E and a relationalization ρ, $\boldsymbol{f}$ is called a* perfect *feature vector if $\rho\left[\boldsymbol{f}, \boldsymbol{\pi}(\boldsymbol{f}, e)\right] = e$ for each $e \in E$.*

[1] In particular, it appears that for any structure set S and feature pool $\mathcal{F}$, one could always find a feature vector $\boldsymbol{f}$ making a non-order-invariant relationalization ρ violate the condition $\boldsymbol{\pi}[\boldsymbol{f}, \rho(\boldsymbol{f}, \boldsymbol{d})] = \boldsymbol{d}$ required by Def. 2.

```
$SelectFeatures(n, \rho, \mathcal{F})$ :
    1. for $i := 1$ to $n$
    2. Choose $f_i \in \mathcal{F}$ such that $f_i = \arg\min Err_{\rho,E}(f_1, \ldots f_{i-1}, f_i))$
    3. next $i$
    4. output $\boldsymbol{f} = f_1, f_2, \ldots f_n$
```

Fig. 1. A greedy algorithm for finding a sub-optimal feature vector $\boldsymbol{f}$ with n components, given a feature pool $\mathcal{F}$ and a relationalization function ρ. The function *Err* is defined in Eq. 6.

Of course, a perfect feature vector would in practice have to be very large (as the following lemma easily formalizes). As such it would fail an important goal of the approach, aiming at extracting a small set of 'principal component' features. Besides, if empoyed in predictive classification applications, it would likely contribute to overfitting.

Lemma 1. *There is no perfect n-component feature vector for an example set E such that $|E| > |D|^n$.*

Proof. Given Def. 2, $\boldsymbol{\pi}(\boldsymbol{f}, e)$ may acquire at most $|D|^n$ distinct values, so if $|E| > |D|^n$, there are some two examples $e_1 \neq e_2$ in E such that

$$\boldsymbol{\pi}(\boldsymbol{f}, e_1) = \boldsymbol{\pi}(\boldsymbol{f}, e_2) \equiv \boldsymbol{d} \tag{5}$$

If $\boldsymbol{f}$ is perfect, it must hold $\rho(\boldsymbol{f}, \boldsymbol{d}) = e_1$ but then it is not perfect as $\rho(\boldsymbol{f}, \boldsymbol{d}) \neq e_2$. □

A requirement clearly more reasonable than perfectness will be based on the notion of similarity between the original and the reconstructed structures. This can be formalized using a distance function (here coinciding with a *metric*) on structures. I accept the traditional requirements below.

Definition 5. *Let $\Delta : S \times S \to \mathcal{R}^+ \cup \{0\}$ be called a* distance *whenever*

1. *$\Delta(s, s) = 0$ for each $s \in S$*
2. *$\Delta(s_1, s_2) = \Delta(s_2, s_1)$ for each $s_1, s_2 \in S$ and*
3. *$\Delta(s_1, s_3) \leq \Delta(s_1, s_2) + \Delta(s_2, s_3)$ for each $s_1, s_2, s_3 \in S$*

A considerable effort has been given in relational learning research to define distances between structured objects. In the next section I will adopt what appears to be the most advanced result in this respect [5].

Finally I am in the position to formalize the task of optimal feature selection for relationalization, assuming the number n of resulting features is prescribed rather than arbitrary.

Definition 6. *Given $n \in \mathcal{N}$, a relationalization ρ and a distance Δ, the* optimal feature selection *task is to find a vector $\boldsymbol{f} = f_1, f_2, \ldots f_n$ ($f_i \in \mathcal{F}$) minimizing the error*

$$Err_{\rho,E}(\boldsymbol{f}) = \sum_{e \in E} \Delta\{e, \rho\left[\boldsymbol{f}, \boldsymbol{\pi}(\boldsymbol{f}, e)\right]\} \tag{6}$$

$Relationalize\_no\_unification(\boldsymbol{f}, \boldsymbol{d})$: Given a vector of features $\boldsymbol{f}$ and truth values $\boldsymbol{d}$ of the features, output a structure s for which $\boldsymbol{f}$ have truth values $\boldsymbol{d}$.

```
1. $s := \{\}$
2. for $i := 1$ to $n$
3. if $d_i = 1$ and $Atms(f_i)\theta \notin s\theta$ then
       $f_i^* := Ground\_Vars\_Injectively\_to\_New\_Consts(f_i, s)$
       $s := s \cup Atms(f_i^*)$
4. next $i$
5. output $s$
```

Fig. 2. A relationalization procedure. The function *Ground_Vars_Injectively_to_New_Consts* maps all variables in a feature f_i onto distinctive constants not appearing in the partially constructed structure s.

In general there is of course no closed solution to the problem and the evaluation of all n-element subsets of $\mathcal{F}$ (or, even worse, all n-element series for a non-order-invariant relationalization) is intractable. A straightforward approach to find a sub-optimal solution relies on a greedy algorithm in Fig. 1. Obviously, any traditional heuristic search strategies could be employed here as well, whether deterministic (such as A^* or beam search) or stochastic (such as GSAT).

3 Application in the Extended Transformation Approach

Here I adhere to the class of features forming a constant-free, function-free conjunction of first-order atoms. These conjunctions have to further comply to some user declarations on syntax; I refer the reader to the paper [2] in the main-track proceedings for details.

The function represented by a feature f is evaluated on an example e as follows

$$f(e) = \begin{cases} 1, \text{ if } Atms(f)\theta \subseteq e \text{ for some substitution } \theta; \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

The distance metric I use is a special case of one proposed by [5]. For two structures s_1, s_2, which are first-order interpretations, it calculates

$$\varDelta(s_1, s_2) = \min_{\theta \in \varTheta} \frac{|s_1\theta \setminus s_2\theta| + |s_2\theta \setminus s_1\theta|}{|s_1 \cup s_2|} \tag{8}$$

where $\varTheta$ are all injective mappings (identity preserving renamings) of γ onto γ, while γ is the set of constants in $s_1 \cup s_2$. Simply put, the metric measures the normalized size of the symmetric difference between the interpretations (atoms sets) s_1 and s_2 with the 'most-matching', identity-preserving renaming of constants.

Lastly, the relationalization function is here implemented either by the algorithm in Fig. 2, or the one in Fig. 3. The former is rather naive: it extends the partially constructed structure s by adding all atoms in $f\theta$ where f is the currently processed feature and θ is a standardized (w.r.t. constants in s) grounding injective (identity preserving) substitution, whenever the atom set of f is not already contained in s. Thus the constructed structure may grow rather quickly as more features are being taken into account. On the contrary, the latter algorithm tries to match the feature's variables with constants in s in a way producing the most conservative extension of s (see Fig. 3 for details).

Lemma 2. *Let $\rho(\boldsymbol{f}, \boldsymbol{d})$ be computed by the algorithm*

- *Relationalize_no_unification$(\boldsymbol{f}, \boldsymbol{d})$ in Fig. 2, or*
- *Relationalize_unification$(\boldsymbol{f}, \boldsymbol{d})$ in Fig. 3.*

Then for any f_i, $Atms(f_i)\theta \subseteq s$ if and only if $d_i = 1$.

I omit the formal proof in this short paper.

Corollary 1. *Both the mentioned algorithms implement relationalization as defined in Def. 2.*

The last remark is in order regarding the meaning of a 'perfect feature vector' in this instantiation of the method. A perfect feature vector in the extended transformation approach (given $|E| \leq |D|^n$, that is $|E| \leq 2^n$ in the ETA) would be one where each feature would be the conjunction of all ground literals in the description of a unique example, thus trivially carrying its complete structural description. A slight modification from the usual approach would have to be made though in the way feature truth values would be determined. Rather than verifying θ-subsumption as in Eq. 7, which would result in the non-invertibility of $\boldsymbol{\pi}$ (Eq. 5) whenever $e_1 \subset e_2$ for some $e_1, e_2 \in E$, I would stipulate

$$f_i(e) = 1 \Leftrightarrow Lits(f_i) = e \tag{9}$$

In other words, I would construct $|E|$ *example-indicator* features, ie. $\boldsymbol{f} = f_1, f_2, \ldots f_{|E|}$ where each $f_i(e_j) = 1$ if $i = j$ and $f_i(e_j) = 0$ otherwise. Then $\rho(\boldsymbol{f}, \boldsymbol{d})$ would be the set of literals in the conjunction f_i iff $d_i = 1$.

4 Implementation and an Experiment

The code for SWI Prolog, implementing the described method can be downloaded from `http://labe.felk.cvut.cz/~zelezny/rela.pl`. It also includes the experimental East West data set along with a pool of 1000 features (from which selection is being made) randomly sampled from the space of declaration-satisfying features using the method [2].

The experiment uses the two relationalization procedures (unifying/non-unifying) as well as the distance metric described in the previous section. The goal is to measure the achieved accuracy of reverse construction of data instances for a range of n (the number of employed features), comparing two strategies:

$Relationalize\_unification(\boldsymbol{f}, \boldsymbol{d}, \Delta)$: Given a vector of features $\boldsymbol{f}$, their truth values $\boldsymbol{d}$ and a distance function Δ, output a structure s where $\boldsymbol{f}$ have truth values $\boldsymbol{d}$.

```
$s := \{\}$
**for** $i := 1$ **to** $n$
if $d_i = 1$ **and** $Atms(f_i)\theta \notin s\theta$ **then**
    $f_i^* := Ground\_Vars\_Admissibly\_Injectively\_with\_Min\_Distance(f_i, s, \Delta, \boldsymbol{d})$
    $s := s \cup Atms(f_i^*)$
**next** $i$
**output** $s$
```

Fig. 3. A relationalization procedure. The function *Ground_Vars_Admissibly_Injectively_with_Min_Dist* maps all variables in a feature f_i onto distinctive constants possibly including those appearing appearing in the partially constructed structure s, in such a way that structural dissimilarity (distance) between s and $Atms(f_i^*)$ measured by the function Δ is minimized, which effectively minimizes the number of atoms by which s is extended in the next step [5]. The minimization is subject to the constraint of $s \cup Atms(f_i^*)$ not making true any feature holding the false value in $\boldsymbol{d}$.

- n features are sampled randomly from the feature pool.
- n feature are selected from the pool by the greedy algorithm from Fig. 1.

The reconstruction accuracy is measured as the average distance $\Delta(e, e')$ between an original structured example e and its reconstructed counterpart e', among all 20 examples in the data set.

Fig. 4 plots the results diagrammatically, with n on a logarithmic scale. The main observations are as follows

- Regarding the two strategies based on random selection of features, varying in using either the non-unifying (Fig. 2) or the unifying (Fig. 3) relationalization algorithm: Both versions perform roughly equally in terms of accuracy. There are no results for the non-unifying version for $n > 200$. Due to the absence of unification, the structure generated by this algorithm grows in size with n and for $n \approx 200$, and the involved computations become intractable. For this reason I abandoned non-unifying relationalization for purposes of the subsequent greedy feature selection.
- For the same reconstruction accuracy achieved, the greedy feature selection algorithm needs about 30 times fewer features than the random feature selection.

5 Discussion, Conclusion

An obvious drawback of the experiment presented above is that features were evaluated (in terms of reconstruction accuracy) on the data set used for their selection. An experiment based on a train/test split is needed.

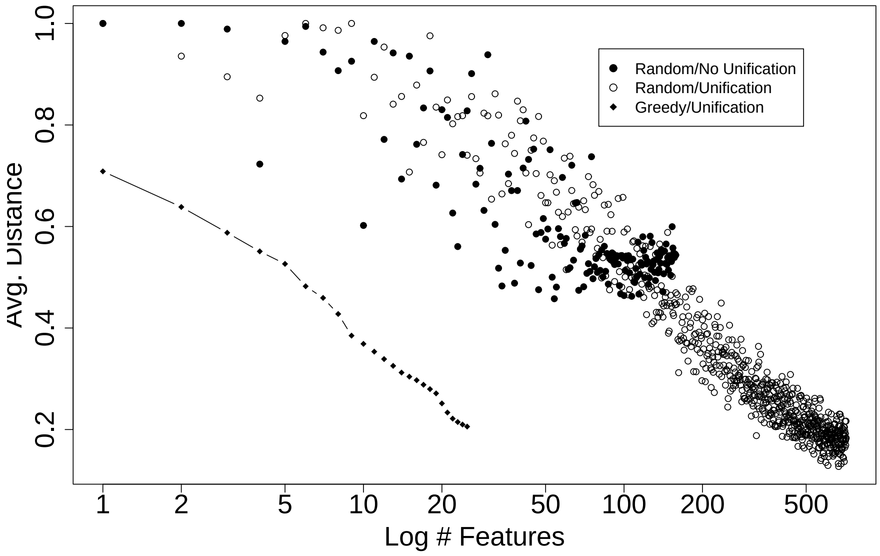

Fig. 4. Average reverse construction accuracy for a growing number of features employed in the propositional representation, for two feature selection methods (Random / Greedy).

It would be interesting to apply the presented general framework to different classes of features than that used by the extended transformation approach. Besides the evident candidates represented by aggregative features [3] or the existential-aggregative hybrids [6], there are some yet completely unexplored possibilities. Consider for example *clauses* (not necessarily Horn), rather than conjunctions, as features. They would be evaluated followingly

$$f_i(s) = \begin{cases} 1, \; s \models f_i; \\ 0, \; \text{otherwise.} \end{cases} \tag{10}$$

(where the structure s is again an interpretation). Applying relationalization with such features would correspond to a deductive process, where one would look for models of a clausal theory, likely accepting some suitable prior probability distribution on the model class (perhaps in terms of their size).

Regarding possible applications, I believe that the presented feature selection approach may contribute to predictive tasks in relational learning based on propositionalization. The reason for the hope is that it has the potential to select a (i) small set of features carrying maximum descriptive information about the original structures, while completely (ii) disregarding any classification cast on the data. These are clearly two strong anti-overfitting factors. Obviously, more experiments are needed to validate this hypothesis.

A straightforward application may of course be also for purposes of lossy compression of structured data.

Acknowledgement

I am supported by the grant agency of the Czech Academy of Sciences through the project KJB201210501. Thanks go to Jan Ramon (KU Leuven) for a consultation on first-order distance functions.

References

1. L. Deraedt and L. Dehaspe. Clausal discovery. *Machine Learning*, 26:99–146, 1997.
2. Zelezny F. Efficient sampling in relational feature spaces. In *Proceedings of the 15th International Conference on Inductive Logic Programming*. Springer-Verlag, 2005.
3. M-A. Krogel, S. Rawles, F. Železný, S. Wrobel, P. Flach, and N. Lavrac. Comparative evaluation of approaches to propositionalization. In *Proceedings of the 13th International Conference on Inductive Logic Programming*. Springer-Verlag, 2003.
4. Nada Lavrac and Peter A. Flach. An extended transformation approach to inductive logic programming. *ACM Trans. Comput. Logic*, 2(4):458–494, 2001.
5. J. Ramon and M. Bruynooghe. A framework for defining distances between first-order logic objects. volume 1446, pages 271–280, 1998.
6. Blockeel H. Dzeroski S. Vens C., Van Assche A. First order random forests with complex aggregates. In *Proceedings of the 14th International Conference on Inductive Logic Programming*. Springer-Verlag, 2004.