Evolutionary propositionalization of multi-relational data

Valentin Kassarnig
Institute of Software Technology
Graz University of Technology
Graz, Austria
kassarnig@ist.tugraz.at

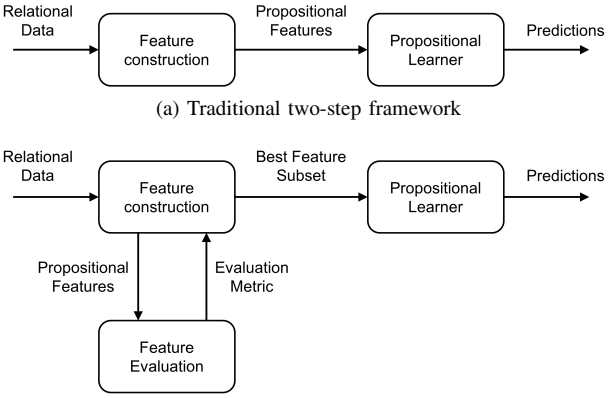
Franz Wotawa
Institute of Software Technology
Graz University of Technology
Graz, Austria
wotawa@ist.tugraz.at

***Abstract*—Propositionalization has been proven to be a very effective solution for multi-relational data mining tasks. Traditional propositionalization approaches follow a two-step principle: transforming the relational data into a single, flat table and applying a propositional learning algorithm. During the transformation the target table gets expanded by adding many new features summarizing the information of the non-target tables. Based on the used feature construction strategy, this leads to a table of very high dimensionality with a lot of irrelevant and/or redundant features that has a negative effect on the predictive performance. In this paper, we propose an alternative propositionalization approach that evaluates the features already during the construction phase and reports only a subset of highly predictive features to the propositional learner. We present an implementation of this approach that adapts a state-of-the-art propositionalization technique and combines it with a genetic algorithm to search for an optimal feature subset. Our experiments on a number of benchmark datasets reveal superior predictive performance of the approach compared to traditional two-step methods making it a considerable extension for any propositionalization algorithm.**

I. Introduction

The rapid advance of data mining techniques during the last decades has lead to countless real-world applications, such as forecasting stock prices [1]–[3], predicting customer behavior [4], [5], or detecting credit card fraud [6]–[8]. However, mining relational data is still problematic since conventional data mining algorithms can be only applied to propositional data. A common approach to solve this problem is *Propositionalization* which typically follows a two-step principle: First, transform the relational data into a single, flat table and second, apply a propositional learning algorithm on the transformed data. This principle is also referred to as *Polka* (named after the two-step dance) and is illustrated in Fig. 1a. Such two-step propositionalization methods have been successfully applied on numerous ILP benchmark tasks as well as real-world applications such as Kaggle competitions [9], [10].

The separation of the two steps in Polka has the downside that the feature construction process is completely isolated from the learning task. Due to the lack of any evaluation of the feature construction process all possible features need to be constructed. Consequently, this results in a table of unnecessarily high dimensionality with a lot of irrelevant (and often redundant) features. Furthermore, in the case of very complex databases and sophisticated feature construction strategies, the exhaustive feature construction may be even intractable [11].



(a) Traditional two-step framework

(b) Proposed framework with feedback-loop

Fig. 1: Traditional and proposed propositionalization framework. The proposed framework has a feature evaluation step with a feedback loop to the feature construction that allows evaluating partial solutions and adapt the feature construction accordingly.

We propose to adapt the traditional approach by adding a feature evaluation step with a feedback loop to the feature construction, as illustrated in Fig. 1b. This change allows us to evaluate the predictive power of feature subsets and adapt the feature construction accordingly. So, in addition to the actual transformation, propositionalization performs a feature subset selection. This class of problems is proven to be NP-complete [12] because only the exhaustive evaluation of all possible subsets would guarantee an optimal solution. In order to tackle this problem we utilize a genetic algorithm (GA) which has a very high rate of convergence to find a near-optimal solution [13]. The GA searches through the space of all possible feature subsets and evaluates the predictive power of the candidate solutions. By only constructing a constant number of features per generation we are capable of propositionalizing complex databases where exhaustive feature construction would be intractable.

DOI reference number: 10.18293/SEKE2018-136

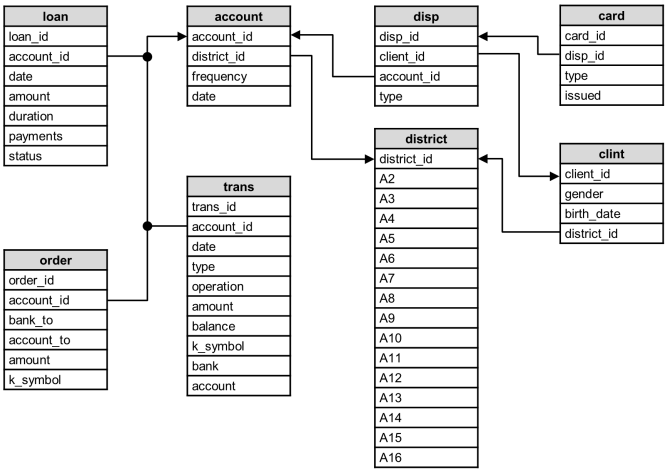


Fig. 2: Data model of the financial database from the PKDD 1999 Discovery Challenge.

In this paper, we use the financial database from the PKDD 1999 Discovery Challenge [14] as a running example to explain different concepts. This database captures information about a bank offering services to private clients. The goal is to find out, what clients need to be watched carefully to minimize the bank losses. Fig. 2 shows the corresponding data model with the table *loan* containing the target attribute *status* that indicates whether there were any repayment problems with a given loan or not.

Our main contributions through this paper are as follows:

- We propose an extension of the traditional two-step propositionalization framework
- We introduce a genetic-based algorithm to propositionalize relational data for multi-relational classification tasks
- We conduct an experimental evaluation of our method on a number of well-known benchmark tasks and compare its performance with those of state-of-the-art methods

II. Related work

The problem of mining relational data has been extensively studied in the past. The two main approaches for this problem are Inductive Logic Programming (ILP) and Propositionalization [15]. Although there are cases where ILP mehthods have been successfully applied [16], propositionalization approaches generally outperform them in terms of speed [17], scalability [17]–[20], and predictive performance [20]–[22]. Furthermore, ILP-based systems perform poorly on a noisy domain compared to numerical propositionalization [15], [17], [21].

Recent works have successfully used aggregation-based propositionalization approaches to automatically mine big databases (up to 100 GB of raw data) [9], [10]. Their experiments showed that with exhaustive feature construction methods an enormous amount of computing power is required to process such massive databases. Even when the workload was distributed among 60 CPUs it took their best scaling method nearly 13 hours to process a database of about one GB raw data [23].

Different approaches have attempted to overcome such shortcomings of traditional two-step propositionalization. The aggregation-based algorithm PRORED [24] avoids exhaustive feature construction by using stochastic optimization. Based on heuristically determined probabilities only a subset of attributes and aggregate functions is chosen to construct features. The used heuristic function makes attributes of tables further away from the target table less likely to be chosen. However, their gain in scalability comes at the cost of reduced accuracy.

Genetic Algorithms (GA) have been demonstrated to be useful tools for propositionalization. Braud and Vrain [25] propose a logic-oriented propositionalization approach with a GA-based feature construction. Their GA optimizes individual features, represented as Horn clause-patterns, through the operations union, intersection, variable isolation, variable move, split, and merge. Similarly, Alfred [26] utilizes a GA as extension of his propositionalization framework *DARA* [27] to find the most predictive patterns of combined attributes. Results indicate that this extension improves the efficiency as well as the performance. Furthermore, in the field of propositional data mining, GAs have been found to be robust and powerful means to find near-optimal subsets of features [28]–[31]. While the mentioned GA-based propositionalization approaches utilize GAs to optimize the predictive power of individual features, our framework optimizes the predictive power of the final feature subset. That is, a feature is either selected or not but it is not changed in any way.

III. Method

A. Genetic algorithm

Our proposed approach adapts the traditional two-step framework by adding a feature evaluation step that evaluates partial solutions (see Fig. 1). For this purpose we utilize a standard Genetic Algorithm (GA) [32] with a rank-based selection strategy. Individuals are encoded as binary strings of length N where N is defined by the total number of possible features under the current feature construction strategy. Every position in the binary string indicates either the presence or absence of a particular feature. A feature can be either an attribute of the target table or a construct based on the attributes of other tables. See Subsection III-C for more details about the feature construction process.

We will refer to this implementation as *GenPro* (GENetic PROpositionalization) throughout the remaining paper. In our experiments we parameterized the GA with the following values:

- Population size: 20
- Max. Number of generations: 150
- Probability of initial selection: 0.01
- Probability of crossover: 0.85
- Probability of mutation: $\frac{1}{N}$

Every population consists of 20 individuals where each individual represents a candidate solution encoded as a binary

string. Over the course of 150 generations those individuals are evaluated, selected, combined and mutated in order to maximize their fitness. The probability of initial selection determines how likely a feature gets chosen to be part of an individual in the initial population. When creating an offspring for the next generation, the probability of crossover defines whether the offspring is derived by combining two individuals or just mutating one. The probability of mutation specifies how likely a bit in the binary string is flipped during the mutation operation. Because N corresponds to the length of the binary string, on average only one feature per individual is either added or removed.

B. Fitness function

The goal of the fitness function is to evaluate the predictive power of a given feature subset. For this we perform a stratified 10-fold cross-validation with a Classification and Regression Tree (CART) [33] and determine the predictive accuracy, defined as

$$accuracy(x) = \frac{\text{Correct predictions}}{\text{Total number of examples}} \tag{1}$$

where x is a bit-string encoded individual representing a subset of features.

As fitness measure for the GA we consider two different metrics. For the first one, Fit_1, we use simply the resulting accuracy from the cross-validation:

$$Fit_1(x) = accuracy(x) \tag{2}$$

The second metric, Fit_2, takes additionally the cost of creating the feature subset into account, as suggested by Yang et al. [29]. In our case, this corresponds to the number of features in the subset. That is, the fitness function favors smaller subsets in order to improve generalizability [34] and reduce computational costs. We define this fitness metric for an individual x as

$$Fit_2(x) = \frac{accuracy(x)}{|x| + \sum_{i=1}^{|x|} x_i} \tag{3}$$

where x_i is the feature at position i. The sum of x corresponds to the number of active features in the subset and $|x|$ is the total number of possible features.

C. Feature construction

The feature construction strategy of a propositionalization algorithm determines the total number of possible features. While exhaustive approaches create all of them up front, we construct them on-the-fly as needed. For our GenPro implementation we adopt the RELAGGS algorithm [35] as feature construction strategy. Note that we could have used here any propositionalization technique that is based on the Polka scheme.

We reimplemented the basic RELAGGS version as presented in its original paper [35]. RELAGGS propagates the identifiers of the target instances to the non-target tables and summarizes then their attributes through numeric aggregation. While for numeric attributes the standard SQL aggregate functions *MIN*, *MAX*, *SUM*, and *AVG* are used, nominal attributes are summarized by counting the occurrences of every distinct value. Additionally, a feature representing the group size of a summarized table is created for every summarized table. The RELAGGS paper gives no indication of how to treat *Date* attributes. Thus, we have extracted from every date its year, month, week number, day of the year, and weekday and treat each of them as a numeric attribute. As described in [36] we set in our experiments the *maximum cardinality* parameter for nominal attributes to 100. It is not specified what upper limit of literals was used in the experiments and we just presume a value of 6 which gives us a sufficiently large number of possible features. In contrast to the original version we allow tables to appear twice in a clause in order to capture additional information about the past [37].

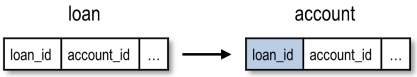
The following example illustrates the basic principle of the feature construction. At first, the target identifier *loan_id* is propagated to the associated table *account* as illustrated in Fig. 3a. This association has a *N:1* multiplicity and thus, every target instance can be directly linked to a particular *account* instance. Consequently there is no summarization needed and the attributes can be simply added to the target table. Next, the target identifiers are further propagated to the table *trans*. This association has a multiplicity of *1:N* which implies that every target instance can belong to multiple *trans* instances. Therefore, the table needs to be summarized so that every target instance corresponds to exactly one row. This is done by applying the previously discussed aggregate functions to each attribute according to its data type. The example in Fig. 3b shows how the numeric attribute *amount* is summarized by applying four different aggregate functions. Each of those four new attributes represents a feature that is eventually added to the target table. While this example illustrates the general idea of the feature construction, in GenPro we do not summarize entire tables but only the attributes needed to create the active features (the ones with a 1 in the binary string) of the current individual.

IV. Evaluation

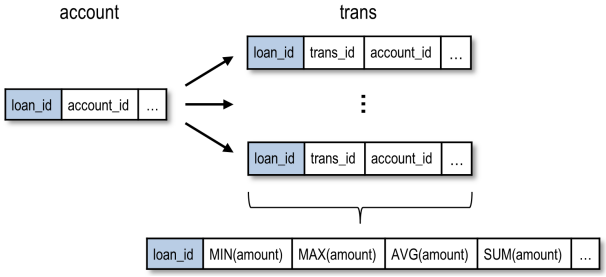
A. Setup

In order to find out how propositionalization benefits from the proposed framework we applied our method GenPro, including both fitness functions, on a number of benchmark tasks. For direct comparison, we performed the same tasks with our implementation of the RELAGGS algorithm, which uses the same feature construction strategy as GenPro but creates all features exhaustively. In addition to the basic version, we also tested RELAGGS with a follow-up feature selection (FS) and dimensionality reduction (DR) step. For FS we used the top 10 % features based on an ANOVA F-test. DR was performed through a Principal Component Analysis (PCA) where the feature space was reduced to 10 % of its original size.

Since the actual performance depends very much on the used learner we used three different models, namely CART,



(a) ID propagation along a *N:1*-association.



(b) ID propagation along a *1:N*-association with subsequent summarization through numeric aggregation.

Fig. 3: Simplified illustration of the RELAGGS feature construction strategy. The colored attribute fields indicate the propagated target identifiers.

Random Forest (RF), and SVM. We used their respective Scikit-learn [38] implementations with 100 estimators for the RF and default parameters other than that. All reported results are based on ten independent and stratified 10-fold cross-validations. For the sake of a fair comparison we stopped the GA in all cases only after 150 generations regardless of whether the fitness score has already converged or not.

We performed the experiments on a PC with Windows 10 Professional, an Intel Core i7 CPU with 2x 1.70 GHz, and 8 GB RAM. The code was written in Python 2.7 an no optimizations techniques, such as parallel or GPU computing, were used. However, constructed features were cached and reused when needed in order to avoid redundant computations.

B. Datasets

We used the financial database [14] from the PKDD 1999 discovery challenge as primary benchmark task to evaluate different GenPro variants. The dataset consists of eight tables (see Fig. 2) and more than a million records. The goal is to predict for a given loan whether there will be any repayment problems. The target table *loan* has 682 instances of which 606 did not cause any problems and only 76 had repayment issues. As suggested by Frank et al. [39] we only used transactions dated before the loan was granted in order to avoid peeking at retrospective data.

Further experiments were performed on the Mutagenesis database [40], Medical database [41], Hepatitis database [39], and the two Musk datasets [42] to cover a wide spectrum of different problem types.

Table I provides an overview of the used datasets and their properties. The last column *# features* describes the total number of features that can be constructed using our RELAGGS implementation.

Dataset	# tables	# target rows	# attributes	# features
Financial	8	682	55	675
Hepatitis	7	500	26	57
Mutagenesis	3	188	14	43
Medical	3	806	64	232
Musk large	2	102	170	665
Musk small	2	92	170	665

TABLE I: Overview of the benchmark datasets

Method	CART	RF	SVM
RELAGGS	0.904	0.918	0.889
RELAGGS + DR	0.819	0.896	0.889
RELAGGS + FS	0.906	0.932	0.889
GenPro with Fit_1	0.964	0.950	0.885
GenPro with Fit_2	**0.971**	0.959	0.891

TABLE II: Predictive accuracies on the financial task. Evaluation of different RELAGGS and GenPro variants for propositionalization in combination with Classification and Regression Trees (CART), Random Forests (RF) and Support Vector Machines (SVM) as predictive models.

V. Results

A. Financial task

This section discusses the results on the financial task. It is divided into three parts: The first paragraph compares the predictive performance of GenPro and RELAGGS, and discusses the impact of different machine learning models. The second paragraph discusses the differences between the two GenPro variants before the last paragraph presents the results of a meta-comparison with other propositionalization methods and multi-view approaches.

Predictive performance: Table II shows the achieved accuracies of different GenPro and RELAGGS variants on the financial task. With an accuracy of 97.1 % the best result was achieved by GenPro with the Fit_2 fitness function and the CART model. This configuration outperformed the best RELAGGS variant by +4 %. Moreover, both GenPro variants with either CART or RF models achieved superior results over any RELAGGS configuration. We can also observe that RELAGGS benefits from feature selection (FS) while dimensionality reduction (DR) has rather a negative effect. Furthermore, the RF classifier seems to better handle the data's high dimensionality leading to better results for RELAGGS. On the other side, GenPro works best with the CART model which was also used to determine the fitness scores. SVM performs poorly on this task and seems completely inapplicable here. Note that both RELAGGS and GenPro produce the same features but in contrast to RELAGGS, GenPro uses only a subset of it for the learning task.

GenPro's superior performance over RELAGGS comes at the cost of decreased computational efficiency. In our experiment a single 10-fold cross-validation with RELAGGS took nearly one minute. On the other side, it took GenPro about three minutes to complete the same task. However, note that GenPro was stopped after 150 generations but converged

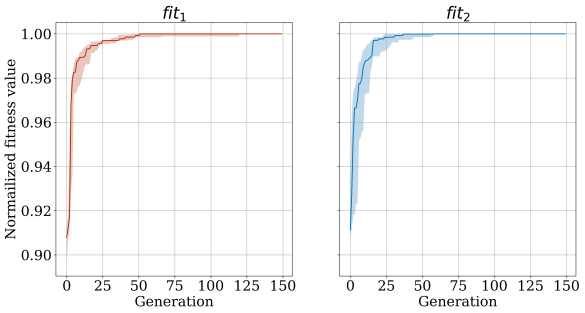


Fig. 4: GenPro's convergence behavior of the fitness value with the two fitness functions fit_1 and fit_2.

already way earlier (see Fig. 4).

GenPro variants: Fig. 4 illustrates GenPro's convergence behavior of the fitness value with either of the two fitness functions. The solid lines indicate the median of the normalized fitness values of ten test runs while the transparent area around it is bound by the respective 25 % and 75 % percentile. We can see that both fitness functions result in a very similar convergence behavior and moreover, in most cases they converge within 75 generations.

In terms of selected feature subset, the two fitness functions produce very different results. On the financial task, fit_1 and fit_2 lead to average subset sizes of 34.3 and 7.3 features, respectively. This indicates that for this task only very few features are necessary to achieve outstanding results.

Meta-comparison: In a follow-up meta-comparison we compared our achieved results on the financial task with those of some prominent techniques for multi-relational classification problems. We considered here the propositionalization methods DARA [27], RELAGGS [35], and CrossMine [19], as well as the multi-view approaches MVC [43] and MRC [20]. Table III shows the predictive accuracies of each representative. With an accuracy of 97.1 % GenPro achieved the highest score; 2 % better than the second best approach DARA. Note that the here reported result of the RELAGGS algorithm origins from the original paper and is slightly better than the ones of our implementation (c.f. Table II). This is because not sufficient information about the actual implementation or experimental setup was available to us in order to fully reproduce their results. Note that the entire comparison here underlies the limitations of meta-analysis approaches and thus, the results should be interpreted with caution [44]. Nevertheless, the outstanding performance of our approach still indicates a high potential.

B. Further benchmark tasks

Table IV shows the empirical results on further benchmark tasks. For reasons of clarity we only report the results of the best RELAGGS and the best GenPro variant, respectively. On four out of five tasks GenPro clearly outperformed RELAGGS.

Method	Accuracy	Source
GenPro	*0.971*	
DARA	0.951	[45]
RELAGGS	0.941	[35]
MVC	0.941	[43]
MRC	0.934	[20]
CrossMine	0.895	[19]

TABLE III: Meta-comparison of different approaches on the financial task.

Dataset	RELAGGS	GenPro	Δ
Musk small	0.823	0.923	+10.0 %
Musk large	0.794	0.846	+5.2 %
Hepatitis	0.857	0.906	+4.9 %
Mutagenesis	0.900	0.919	+1.9 %
Medical	0.903	0.903	±0.0 %

TABLE IV: Predictive accuracies on further benchmark tasks. Results indicate the highest accuracies achieved by any RELAGGS or GenPro variant, respectively. The Δ column indicates the difference between the results of the two methods.

It achieved predictive accuracies of up to 10 % higher than RELAGGS. In the one remaining case, on the Medical dataset, both methods achieved the same result. Surprisingly, even at the very small Hepatitis and Mutagenesis datasets (<60 features), GenPro could score a respectable improvement over RELAGGS which suggests a high degree of versatility of the underlying framework.

VI. Discussion and Conclusion

In this paper, we have presented a modified propositionalization approach, that overcomes disadvantages of the traditional two-step propositionalization framework. By combining feature construction and feature evaluation we are capable of avoiding exhaustive feature construction and produce only a subset of highly predictive features. Furthermore, we have demonstrated how to use a Genetic Algorithm (GA) to implement the proposed approach. Our empirical results suggest competitive performance as well as great versatility of our approach. In a direct comparison, it outperformed a state-of-the-art propositionalization method on numerous benchmark tasks. Furthermore, it achieved superior results in a meta-comparison with other propositionalization methods and multi-view approaches. Thus, we conclude that this approach represents a considerable extension for any propositionalization technique to improve the predictive performance.

Despite the promising results, our approach has also some limitations which are discussed hereafter. First, we have tested our approach only with a single feature construction strategy. Although it achieved outstanding results, the framework's performance should be also evaluated with other strategies in order to strengthen the conclusions. Furthermore, all our experiments were performed on relatively small databases making it precarious to make assumptions about the scaling behavior. Compared to exhaustive feature construction, GenPro's properties as anytime-algorithm support undeniably the ability to handle bigger and more complex databases. Also, the increased computational cost can be reduced to a minimum

through parallel computation and further optimizations [46]. However, to fully evaluate the scalability of our approach experiments on big real-world datasets need to be performed and is therefore, topic of future research.

REFERENCES

[1] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with Applications*, vol. 19, no. 2, pp. 125–132, 2000.

[2] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," *Expert Systems with applications*, vol. 29, no. 4, pp. 927–940, 2005.

[3] P.-F. Pai and C.-S. Lin, "A hybrid arima and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[4] C. Apte, B. Liu, E. P. Pednault, and P. Smyth, "Business applications of data mining," *Communications of the ACM*, vol. 45, no. 8, pp. 49–53, 2002.

[5] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007. New York, NY, USA, 2007, p. 35.

[6] E. Aleskerov, B. Freisleben, and B. Rao, "Cardwatch: A neural network based database mining system for credit card fraud detection," in *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*. IEEE, 1997, pp. 220–226.

[7] P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo, "Distributed data mining in credit card fraud detection," *IEEE Intelligent Systems and Their Applications*, vol. 14, no. 6, pp. 67–74, 1999.

[8] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[9] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*. IEEE, 2015, pp. 1–10.

[10] H. T. Lam, J.-M. Thiebaut, M. Sinn, B. Chen, T. Mai, and O. Alkan, "One button machine for automating feature engineering in relational databases," *arXiv preprint arXiv:1706.00327*, 2017.

[11] C. A. Ferreira, J. Gama, and V. S. Costa, "Exploring multi-relational temporal databases with a propositional sequence miner," *Progress in Artificial Intelligence*, vol. 4, no. 1-2, pp. 11–20, 2015.

[12] A. A. Albrecht, "Stochastic local search for the feature set problem, with applications to microarray data," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1148–1164, 2006.

[13] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern recognition*, vol. 43, no. 1, pp. 5–13, 2010.

[14] P. Berka *et al.*, "Guide to the financial data set," *PKDD2000 discovery challenge*, 2000.

[15] S. Kramer, "Relational learning vs. propositionalization: Investigations in inductive logic programming and propositional machine learning," *AI communications*, vol. 13, no. 4, pp. 275–276, 2000.

[16] S. Džroski, "Relational data mining," in *Relational Data Mining*, S. Dèzeroski, Ed. New York, NY, USA: Springer-Verlag New York, Inc., 2000, ch. Relational Data Mining Applications: An Overview, pp. 339–360. [Online]. Available: http://dl.acm.org/citation.cfm?id=567222.567240

[17] R. Alfred and D. Kazakov, "Pattern-based transformation approach to relational domain learning using dynamic aggregation for relational attributes." in *DMIN*, 2006, pp. 118–124.

[18] H. Blockeel and M. Sebag, "Scalability and efficiency in multi-relational data mining," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, pp. 17–30, 2003.

[19] X. Yin, J. Han, J. Yang, and S. Y. Philip, "Crossmine: Efficient classification across multiple database relations," in *Constraint-Based mining and inductive databases*. Springer, 2006, pp. 172–195.

[20] A. Thakkar and Y. Kosta, "Survey of multi relational classification (mrc) approaches & current research challenges in the field of mrc based on multi-view learning," *International Journal of Soft Computing and Engineering (1)*, vol. 247, p. 252, 2012.

[21] C. Perlich and F. Provost, "Distribution-based aggregation for relational learning with identifier attributes," *Machine Learning*, vol. 62, no. 1-2, pp. 65–105, 2006.

[22] A. J. Knobbe, M. De Haas, and A. Siebes, "Propositionalisation and aggregates," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2001, pp. 277–288.

[23] H. T. Lam, T. N. Minh, M. Sinn, B. Buesser, and M. Wistuba, "Learning features for relational data," *arXiv preprint arXiv:1801.05372*, 2018.

[24] V. Gjorgjioski and S. Dzeroski, "Stochastic propositionalization of relational data using aggregates," 2008.

[25] A. Braud and C. Vrain, "A genetic algorithm for propositionalization," in *International Conference on Inductive Logic Programming*. Springer, 2001, pp. 27–40.

[26] R. Alfred, "Feature transformation: A genetic-based feature construction method for data summarization," *Computational Intelligence*, vol. 26, no. 3, pp. 337–357, 2010.

[27] R. Alfred and D. Kazakov, "Data summarization approach to relational domain learning based on frequent pattern to support the development of decision making," in *International Conference on Advanced Data Mining and Applications*. Springer, 2006, pp. 889–898.

[28] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," in *Handbook Of Pattern Recognition And Computer Vision*. World Scientific, 1993, pp. 88–107.

[29] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature extraction, construction and selection*. Springer, 1998, pp. 117–136.

[30] H. Vafaie and K. De Jong, "Robust feature selection algorithms," in *Tools with Artificial Intelligence, 1993. TAI'93. Proceedings., Fifth International Conference on*. IEEE, 1993, pp. 356–363.

[31] F. Z. Brill, D. E. Brown, and W. N. Martin, "Fast generic selection of features for neural network classifiers," *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 324–328, 1992.

[32] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[33] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[34] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International Work-Conference on Artificial Neural Networks*. Springer, 2005, pp. 758–770.

[35] M.-A. Krogel and S. Wrobel, "Transformation-based learning using multirelational aggregation," in *International Conference on Inductive Logic Programming*. Springer, 2001, pp. 142–155.

[36] M.-A. Krogel, "On propositionalization for knowledge discovery in relational databases," Ph.D. dissertation, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek, 2005.

[37] M. Samorani, F. Ahmed, and O. R. Zaïane, "Automatic generation of relational attributes: An application to product returns," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1454–1463.

[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[39] R. Frank, F. Moser, and M. Ester, "A method for multi-relational classification using single and multi-feature aggregation functions," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2007, pp. 430–437.

[40] A. K. Debnath, d. C. R. Lopez, G. Debnath, A. J. Shusterman, and C. Hansch, "Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity." *Journal of medicinal chemistry*, vol. 34, no. 2, pp. 786–797, 1991.

[41] M. Jan, S. Tsumoto, and K. Takabayashi, "Medical thrombosis data description."

[42] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1, pp. 31–71, 1997.

[43] S. Modi, "Relational classification using multiple view approach with voting," *International Journal of Computer Applications*, vol. 70, no. 16, 2013.

[44] T. D. Spector and S. G. Thompson, "The potential and limitations of meta-analysis." *Journal of Epidemiology and Community Health*, vol. 45, no. 2, p. 89, 1991.

[45] R. Alfred, *A Data Summarisation Approach to Knowledge Discovery*. Citeseer, 2008.

[46] E. Cantú-Paz, "A survey of parallel genetic algorithms," *Calculateurs paralleles, reseaux et systems repartis*, vol. 10, no. 2, pp. 141–171, 1998.