ILP, the Blind, and the Elephant: Euclidean Embedding of Co-Proven Queries

Hannes Schulz[1] and Kristian Kersting[2] and Andreas Karwath[1]

[1] Institut für Informatik, Albert-Ludwigs Universität
Georges-Köhler-Allee 79, 79110 Freiburg, Germany
{schulzha,karwath}@informatik.uni-freiburg.de
[2] Dept. of Knowledge Discovery, Fraunhofer IAIS
Schloss Birlinghoven, 53754 St Augustin, Germany
kristian.kersting@iais.fraunhofer.de

Abstract. Relational data is complex. This complexity makes one of the basic steps of ILP difficult: understanding the data and results. If the user cannot easily understand it, he draws incomplete conclusions. The situation is very much as in the parable of the blind men and the elephant that appears in many cultures. In this tale the blind work independently and with quite different pieces of information, thereby drawing very different conclusions about the nature of the beast. In contrast, visual representations make it easy to shift from one perspective to another while exploring and analyzing data. This paper describes a method for embedding interpretations and queries into a single, common Euclidean space based on their co-proven statistics. We demonstrate our method on real-world datasets showing that ILP results can indeed be captured at a glance.

1 Introduction

Once upon a time, there lived six blind men in a village. One day the villagers told them, “Hey, there is an elephant in the village today.” They had no idea what an elephant is. They decided, “Even though we would not be able to see it, let us go and feel it anyway.” All of them went where the elephant was and touched the elephant. Each man encountered a different aspect of the elephant and drew a different inference as to its essential nature. One walked into its side, concluding that an elephant is like a wall. Another, prodded by the tusk, declared that an elephant is like a spear. The chap hanging onto the tail was convinced that he had found a sort of rope. The essential nature of the elephant remained undiscovered.

The tale is that of “The Blind and the Elephant”, which appears in many cultures. It illustrates the problem many ILP users face, to make sense of relational data and models, the elephants, before applying their algorithms or while interpreting the results. Due to the complexity of the data and the models, the user can only touch small parts of them, like specific queries. Hence, he often gets only a narrow and fragmented understanding of their meaning.

In contrast, visual representations make it easy to shift from one perspective to another while exploring and analyzing data. How to visually explore relational data and queries jointly, however, has not received a lot of attention within the ILP community. This can be explained by the fact that relational data involves objects of several very different types without a natural measure of similarity. Syntactic similarity measures typically used in ILP (c. f. [8]) cannot be used for two reasons: First, they cannot relate interpretations and queries, and second, syntactically different queries might actually have identical semantics w.r.t. the data, resulting in hard to optimize and hard to interpret embeddings.

Our paper addresses this problem of creating embeddings which visualize interpretations and queries. In our embeddings, query-query and query-interpretation distances are determined by their respective co-occurrence statistics, i. e., queries are placed close to queries which were often co-proven and close to interpretations in which they are true. Properly colored and labeled, our embeddings provide a generic visualization and interactive exploration of relational databases as well as the working of query-generating ILP-algorithms.

2 Euclidean Embedding of Co-proven Queries

Given a set of interpretations $\mathcal{I}$ and queries $\mathcal{Q}$, we assign positions in $\mathbb{R}^d$ to all $i \in \mathcal{I}$ and $q \in \mathcal{Q}$ such that the distances in $\mathbb{R}^d$ reflect the co-occurrence statistics. Computing joint embeddings of $\mathcal{I}$ and $\mathcal{Q}$ essentially requires three steps: (1) collecting embeddable queries, (2) embedding queries and interpretations into a single Euclidean space, and – as an optional postprocessing step – (3) labelling the representation by extracting local representatives. We make the C++ implementation of our method available on our website.[3]

Step 1 – Queries: Given a finite set $\mathcal{I}$ of observed interpretations, any ILP algorithm can be used to preselect embeddable queries $\mathcal{Q}$ for $\mathcal{I}$. Although in this work we concentrate on feature miners, our embeddings can also be used to for example embed features extracted for classification. In this paper, we use *Molfea* [5] and *C-armr*[4] [2] to mine databases of molecules. Both systems are inspired by the Agrawal's *Apriori* algorithm [1]: they construct general queries and specialize them only if they are frequent. Only queries more frequent than some threshold are retained and further expanded, i.e., specialized. While *Molfea* constructs linear fragments only (atom, bond, atom, . . .), *C-armr* constructs general queries and can take background knowledge into account as well. In addition to the queries, we also store the interpretations in which they were true. This will prove useful for the next step and can efficiently be represented in a binary matrix $\mathcal{C} \in \{0, 1\}^{|\mathcal{Q}| \times |\mathcal{I}|}$, where $|\mathcal{Q}|$ is the number of queries and $|\mathcal{I}|$ is the number of observed interpretations.

Step 2 – Embedding: We wish to assign positions to elements in $\mathcal{I}$ and $\mathcal{Q}$ through mappings $\phi : \mathcal{I} \mapsto \mathbb{R}^d$ and $\psi : \mathcal{Q} \mapsto \mathbb{R}^d$ for a given dimensionality d.

[3] `http://www.ais.uni-bonn.de/~schulz/`

[4] in the CLASSIC'CL implementation [10]

These mappings should reflect the dependence between $\mathcal{I}$ and $\mathcal{Q}$ such that the co-occurrence $\mathcal{C}_{qi}$ (c. f. Step 1) of some $i \in \mathcal{I}$ and $q \in \mathcal{Q}$ determines the distance between $\phi(i)$ and $\psi(q)$.

For this purpose, we employ Globerson *et al.*'s *CODE* algorithm [4]. *CODE* is a generic scheme for co-occurrence-based embedding algorithms. Its main idea is to represent the empirical joint distribution $\bar{p}(A, B)$ of two random variables A and B in a low dimensional space, such that items with high co-occurrence probability are placed close to each other. In *CODE*, this idea is realized in the assumption that

$$p(i, q) = Z^{-1} \cdot \exp(- \|\phi(i) - \psi(q)\|^2), \tag{1}$$

where $Z = \sum_{i,q} \exp(- \|\phi(i) - \psi(q)\|^2)$ is a normalization constant. Starting from a random position assignment in ϕ and ψ, *CODE* then minimizes the Kullback-Leibler divergence between $\bar{p}(i, q) \propto \mathcal{C}_{qi}$ and $p(i, q)$ by maximizing the log-likelihood $l(\phi, \psi) = \sum_{i,q} \bar{p}(i, q) \log p(i, q)$. The maximization is performed by gradient ascent on the gradient of the log-likelihood function derived with respect to the axis of the embedding space for each $i \in \mathcal{I}$ and $q \in \mathcal{Q}$. The same problem can also be stated as a convex semi-definite programming problem which is guaranteed to yield the optimal solution, see [4] for details.

Our situation, however, is slightly more complicated. Following suggestions by Globerson *et.al.*, we extend the simple model by marginalization and within-variable co-occurrence measures.

(i) First, we observe that the marginal probability of an interpretation is a quantity artificially created by the number of queries which are true in it. An embedding influenced by this margin is not very intuitive to read. We therefore make the embedding insensitive to $\bar{p}(i)$ by adding it as a factor to (1), yielding

$$p(i, q) = \frac{1}{Z} \cdot \bar{p}(i) \cdot \exp(-\|\phi(i) - \psi(q)\|^2). \tag{2}$$

(ii) However, the results will still be unsatisfactory: using $p(\mathcal{I}, \mathcal{Q})$ only, *CODE* tends to map the interpretations to a circle. This results from the fact that $\mathcal{C}$ is binary and all distances are therefore enforced to be of the same length. To generate more expressive embeddings, we use the interpretations and queries to generate a non-binary query-query co-occurence matrix $\mathcal{D} = \mathcal{C}\mathcal{C}^T$ and set $\bar{p}(q, q') \propto \mathcal{D}_{qq'}$. This co-proven statistics of queries q and q' should be represented by distances in the embedding (similar to (1)) as

$$p\left(q, q'\right) = \frac{1}{Z} \cdot \exp\left(-\|\psi(q) - \psi(q')\|^2\right) \ . \tag{3}$$

As described above, we assign initial positions randomly but now adapt them so that they maximize the "log-likelihood" of the combined and weighted models in (2) and (3):

$$l(\phi, \psi) = \sum\nolimits_{i,q} \bar{p}(i, q) \log p(i, q) + |\mathcal{I}|/|\mathcal{Q}| \cdot \sum\nolimits_{q,q'} \bar{p}\left(q, q'\right) \log p\left(q, q'\right) \ .$$

Thus, our embeddings reflect the relations between co-proven queries as well as interpretations and queries.

Step 3 – Condensation: Literally thousands of queries and instances can be embedded into a single Euclidean space and can – as a whole – provide useful insights into the structure of the data. However, we would also like to get a grasp on what queries in certain regions focus on. To do so, we propose to single out queries q in an iterated fashion. Specifically, we assign to each query q the weight $\mathrm{w}(q) = F(q)/\operatorname{length}(q)$, where $F(q)$ is q's F_1 or F_2-measure, see e.g. [7], and $\operatorname{length}(q)$ is its description length. We now locally remove queries with a low weight in a two-step process. First, we build the k-nearest neighbour graph of the embedded queries. From the weight, we subtract the weight of its graph neighbours, thereby removing large-scale differences. Second, we increase weights of queries with lower weighted neighbours and decrease weights which have higher weighted neighbours. The last step is repeated until the number of queries q with a positive weight is not changing anymore. In other words, we prefer short queries with high F-measures on a local neighbourhood.

3 Interpretation of Generated Embeddings

When feature miners are used to generate queries, embeddings have a typical shape which looks like a tree as seen from above.

(i) Very common queries are generally placed in the center. For databases where different classes can be distinguished, we note that those central queries typically are not very discriminative. For example, in a molecular database containing organic molecules, we would expect chains of carbon atoms to be in the center of the embedding.

(ii) Queries which hold only on a subset of the interpretations are placed further away from the center. Those queries can therefore be seen as "specializations" of the queries in the center or as a cluster of interpretations. The specializations can describe discriminative or non-discriminative properties. In the database of organic molecules, certain aromatic rings might be common in one group of the molecules, but not in the other.

(iii) The tree-like branching repeats recursively. Neighboring branches most likely represent a meta-cluster of interpretations; far away branches represent interpretations which are very dissimilar w.r.t their queries.

Please note that the embedding reflects two of the most important axis in ILP learning tasks: Learning a theory is always a tradeoff between coverage and simplicity. In our embeddings, it is easy to judge how specific a query is, how many of the total interpretations it covers and how it relates to other, similar queries. Furthermore, the embeddings abstract from the syntax of the queries. They are solely determined by the interaction of the queries with each other and the interpretations, which is essential if the complexity of queries necessary to distinguish interpretations varies over the database.

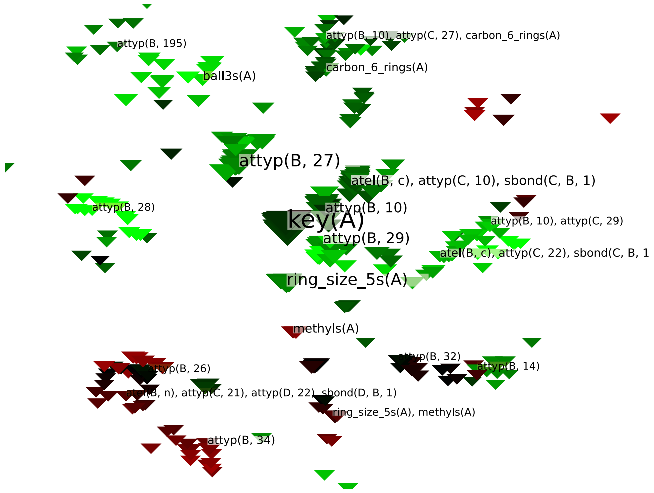

Fig. 1. Mutagenesis embedding. We show all frequent queries (triangles) distinct w.r.t. the interpretations. Black/colored queries have low/high precision, small/large queries have low/high recall. Red/green queries indicate negative/positive class. The queries with textual descriptions were automatically selected, the trivial `key` attribute was omitted in all but the central queries.

4 Showcases

We tested our approach on several real-world datasets for the two-dimensional case. To provide a qualitative assessment of our method, we apply it to datasets where some structures or models have already been discovered and point out the properties described in Section 3.

4.1 Mutagenesis

On **Mutagenesis** [9], the problem is to predict the mutagenicity of a set of compounds. In our experiments, we use the atom and bond structure information only (including the predefined predicate like *ball3s*, *ring_size_5s*, and others). The dataset consists of 230 compounds (138 positives, 92 negatives). The 2D Euclidean embedding is shown and in Fig. 1.

As discussed in Section 1, the most common query, here `key(A)`, is placed in the center. Also notice specializations branching from the center, for example variants of `attyp(B,10)`. The embedding reflects rules we could induce employing Srinivasan's ALEPH on the same dataset such as `active(A) :- attyp(A,B,29), ring_size_5s(A)` or `active(A) :- ball3s(A)`. Queries which primarily hold on positive/negative interpretations are spatially well separated and form clusters which indicate similar coverage of interpretations.

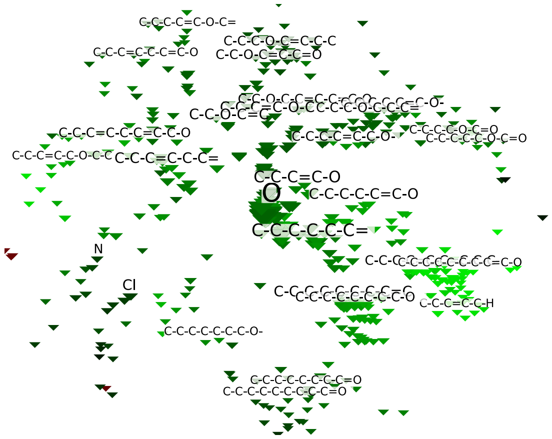

Fig. 2. Estrogen embedding. The coding is as in Fig. 1; only black/colored queries indicate now low/high mean activity of the respective interpretations.

4.2 Estrogen Database

The **Estrogen** database was extracted from the EPA's DSSTox NCTRER Database[5]. The original dataset was published by Fang *et al.* [3], and is specially designed to evaluate QSAR approaches. The NCTRER database provides activity classifications for a total of 232 chemical compounds, which have been tested regarding their binding activities for the estrogen receptor. The database contains a diverse set of natural, synthetic, and environmental estrogens, and is considered to cover most known estrogenic classes spanning a wide range of biological activity [3]. Here, "activity" is an empirically measured value between 0 and 100, which we averaged and used as a query's color. The 2D Euclidean embedding is shown and discussed in Fig. 2.

The embedding shows oxygen (O) as a primary component of organic molecules and a typical branching structure, starting from the center and on a smaller level for example from nitrogen (N) and chlorine (Cl) -based compounds. The activity varies smoothly in the embedding, as extending a chain with certain atoms increases or decreases the chance of higher activity. Even for clearly active molecules, different subtypes can be distinguished.

In their original publication Fang *et al.* have identified that a phenolic ring connected by one to three atoms to another benzene ring is one of the key features that have to be present regarding the likelihood of a compound being an ER ligand. A phenolic ring is a 6-carbon benzene ring with an attached hydroxyl (OH) group. In the embedding, it can be seen that this is reflected in features like `C-C-C=C-O`, which indicates that there is a path of one carbon atom to (a part of) a ring structure (`C-C=C`) connected to an oxygen.

[5] http://www.epa.gov/ncct/dsstox/sdf_nctrer.html

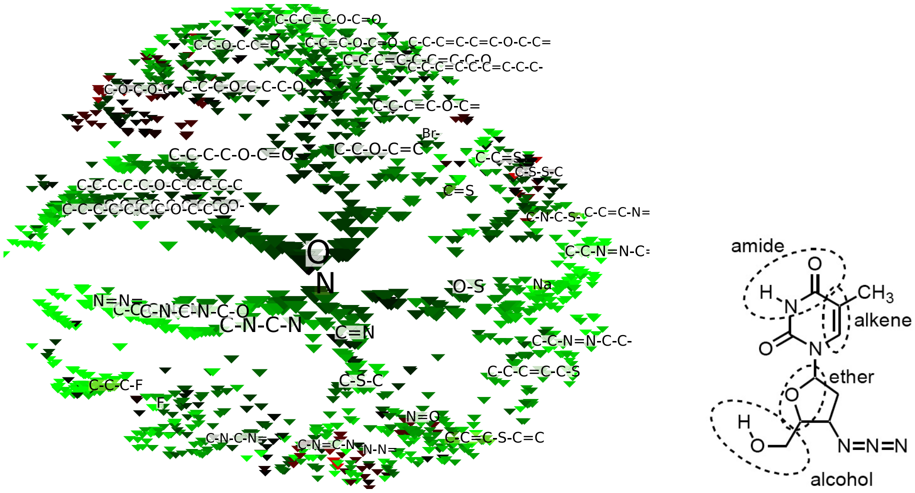

Fig. 3. Left: **AIDS** database embedding. The coding is as in Fig. 1. Right: Azidothymidine (AZT), a component known to be very active. AZT and derivatives are represented by the left central branch in the embedding.

4.3 AIDS Database

The DTP **AIDS** Antiviral Screening Database originating from the NCI's development therapeutics program NCI/NIH[6] consists of SMILES representations of 41,768 chemical compounds [6]. Each data entry is classified as either active, moderately active, or inactive. A total of 417 compounds are classified as active, 1,069 as moderately active, and 40,282 as inactive. We have converted this dataset into SDF format using the OpenBabel toolkit and randomly sampled 400 active and 400 moderate/inactive compounds. The 2D Euclidean embedding is shown and discussed in Fig. 3.

In this database we can clearly see a natural distinction of queries: While oxygen (O) and nitrogen (N) are both common, the lower half of the embedding focuses on nitrogen-based molecules. Within this half, further distinctions (left: flourine (F), right: sulfur (S) -based compounds) can be made.

The embedding clearly indicates compounds that are derivatives of Azidothymidine (AZT), a potent inhibitor of HIV-1 replication. In the left central branch, the embedding clearly indicates prominent features of AZT, such as the nitrogen-group `N=N=N` and the `C-N-C-N-C-O` chain connecting both rings.

5 Concluding Remarks

In our opinion, to unveil its full power, ILP must incorporate visual analysis methods. With the work presented here, we have made a step in this direction.

[6] http://dtp.nci.nih.gov/

We have presented the first method for embedding interpretations and queries into the same Euclidean space based on their co-occurrence statistics. As our experiments demonstrate, the spatial relationships in the resulting embedding are intuitive and can indeed reveal useful and important insights at a glance. Aside from their value for visual analysis, embeddings are also an important tool in unsupervised learning and as a preprocessing step for supervised learning algorithms. In future research, we will explore this direction.

Acknowledgments: The authors would like to thank Amir Globerson for providing the *CODE* code. The research was partially supported by the EU seventh framework program under contract no. Health-F5-2008-200787 (*OpenTox*), and by the Fraunhofer ATTRACT fellowship STREAM.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann, San Francisco, CA, USA, 1994.
2. L. De Raedt and J. Ramon. Condensed representations for inductive logic programming. In *Proceedings of 9th International Conference on the Principles of Knowledge Representation and Reasoning*, pages 438–446, 2004.
3. H. Fang, W. Tong, L.M. Shi, R. Blair, R. Perkins, W. Branham, B.S. Hass, Q. Xie, S.L. Dial, C.L. Moland, , and D.M. Sheehan. Structure-activity relationships for a large diverse set of natural, synthetic, and environmental estrogens. *Chem. Res. Tox*, 14:280–294, 2001.
4. A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean Embedding of Co-occurrence Data. *The Journal of Machine Learning Research*, 8:2265–2295, 2007.
5. C. Helma, S. Kramer, and L. De Raedt. The molecular feature miner MolFea. In *Proceedings of the Beilstein-Institut Workshop*, 2002.
6. S. Kramer, L. De Raedt, and C. Helma. Molecular feature mining in HIV data. In Foster Provost and Ramakrishnan Srikant, editors, *Proc. KDD-01*, pages 136–143, New York, NY, USA, August 26-29 2001. ACM Press.
7. D.D. Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Int. ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254, 1995.
8. J. Ramon. *Clustering and instance based learning in first order logic*. PhD thesis, CS Dept., K.U. Leuven, 2002.
9. A. Srinivasan, S. H. Muggleton, R. D. King, and M. J. E. Sternberg. Theories for Mutagenicity: A Study of First-Order and Feature -based Induction. *Artificial Intelligence Journal*, 85:277–299, 1996.
10. C. Stolle, A. Karwath, and L. De Raedt. CLASSIC'CL: An Integrated ILP System. In A. G. Hoffmann, H. Motoda, and T. Scheffer, editors, *Discovery Science*, volume 3735 of *Lecture Notes in Computer Science*, pages 354–362. Springer, 2005.