Towards Extracting Faithful and Descriptive Representations of Latent Variable Models

Iván Sánchez **Tim Rocktäschel** **Sebastian Riedel** **Sameer Singh**
Department of Computer Science
University College London
{i.sanchezcarmona,t.rocktaschel,s.riedel}@cs.ucl.ac.uk

Computer Science & Engineering
University of Washington
sameer@cs.washington.edu

Abstract

Methods that use latent representations of data, such as matrix and tensor factorization or deep neural methods, are becoming increasingly popular for applications such as knowledge base population and recommendation systems. These approaches have been shown to be very robust and scalable but, in contrast to more symbolic approaches, lack interpretability. This makes debugging such models difficult, and might result in users not trusting the predictions of such systems. To overcome this issue we propose to extract an interpretable proxy model from a predictive latent variable model. We use a so-called pedagogical method, where we query our predictive model to obtain observations needed for learning a descriptive model. We describe two families of (presumably more) descriptive models, simple logic rules and Bayesian networks, and show how members of these families provide descriptive representations of matrix factorization models. Preliminary experiments on knowledge extraction from text indicate that even though Bayesian networks may be more faithful to a matrix factorization model than the logic rules, the latter are possibly more useful for interpretation and debugging.

1 Introduction

In many successful machine learning models, a set of latent vectors is learned by means of minimizing an error function with respect to training data. These include models of matrix and tensor factorization and neural architectures. One advantage of these models is their scalability: they can be trained on massive amounts of data and achieve high accuracy, making them attractive for many large-scale, real-world tasks such as Recommender Systems (Koren, Bell, and Volinsky 2009) and Information Extraction (Riedel et al. 2013). An increasingly desirable property for machine learning systems is human interpretability, both to (a) explain the model and its predictions, and (b) to debug possible mistakes. However, latent representations that use such dense, real-valued vectors are notoriously difficult to interpret.

One way to address this lack of interpretability is to employ hybrid symbolic and probabilistic frameworks such as Markov Logic (Richardson and Domingos 2006) or Bayesian Logic Programs (Milch et al. 2005), but in practice these often suffer from limited scalability. Constraints such as those imposed in nonnegative matrix factorization make considerable steps towards addressing this concern, however they still lack the interpretability of purely symbolic models in which, for example, predictions are derived from explicitly stated rules or from a few features.

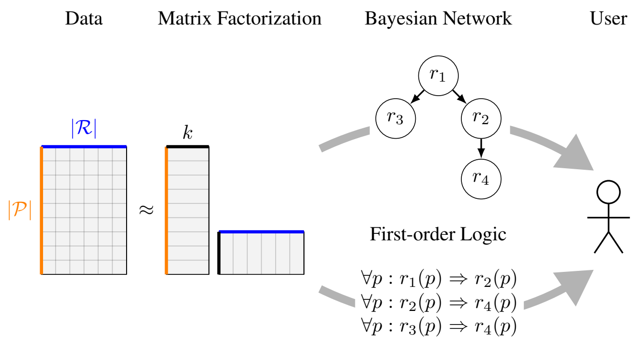

Figure 1: Since the internal representation of latent variable models (e.g. latent vectors of a matrix factorization model) are not easy to comprehend by the end-user, we investigate two simpler but more interpretable proxy models: Bayesian networks and first-order logic formulae.

In this paper, we still use scalable latent variable models for prediction, but additionally learn simpler, human-interpretable *descriptive* models that can be used to explain the behavior of these more complex latent models. More concretely, given a latent variable model, we seek to find a more interpretable proxy model that approximately behaves like the original model. This problem has been addressed before in the context of Artificial Neural Networks (Craven and Shavlik 1996), (Thrun 1995), where a set of logic rules is extracted from a neural network by either, training on predictions from the neural network, analyzing its neurons, or both. The objective of the set of rules learned is to mimic the behavior of the neural network in order to act as a descriptive model and provide confidence in the predictions made by the neural network. To the best of our knowledge, however, there is no work in knowledge extraction from tensor-based models; given the wide impact and acceptance of this models, we consider of big importance to have a descriptive proxy model that can provide a human-readable description of their predictions. In order to find an interpretable version of latent representation models, we need to define the family of models to search over. Models in this family should be easy to interpret by the end-user, yet powerful enough to be as faithful to the original predictive model as possible. Clearly there is a trade-off here: one can achieve perfect reproduction by choosing the original model class itself (say matrix factorization) but then obviously gain no interpretability; on the other hand, a set of symbolic rules may be quite interpretable, but do not handle noise or provide confidence measures that makes the original model useful. One can use probabilistic logic representations such as Markov Logic to find a sweet-spot, but inference is often intractable in these frameworks, which makes evaluating their faithfulness to the original model difficult.

We present our ideas in context of interpreting a matrix factorization (MF) model for information extraction (Riedel et al. 2013), and investigate two types of descriptive model classes, (1) (limited) first-order logic formulae, and (2) a Bayesian network (BN) tree (see Figure 1). The former is at the "most interpretive" end of the spectrum, but accurately reproduction of the MF model predictions is difficult. The latter sacrifices interpretability (and explicit coverage of all correlations the factorization model captures), but provides a model that is a better fit to the original model, while still allowing efficient learning and inference. We use predictions of the matrix factorization model as training data for the descriptive models and evaluate the fidelity of the descriptive models by comparing their predictions to those by matrix factorization. We observe that the rule-based model is indeed a poorer fit than Bayesian network, however it is able to identify errors in the factorization model that are only captured indirectly in the Bayesian network. This raises several interesting research questions for the research community to investigate.

2 Background

We will first give a brief description of the matrix factorization model. Then we will discuss two possible model families that can serve as more interpretable proxy models.

2.1 Matrix Factorization

We work with a variant of the matrix factorization model for knowledge base completion (Riedel et al. 2013). Given a sparse high-dimensional matrix where rows $\mathcal{P}$ correspond to entity-pairs and columns $\mathcal{R}$ to OpenIE-like surface patterns and structured Freebase relations, the aim is to learn k-dimensional distributed vector representations for entity-pairs and relations that approximate given factual knowledge and generalize to unobserved facts (see left half of Figure 1). Once we have learned vector representations of entity-pairs and relations, any fact can be predicted efficiently by applying the sigmoid function to the dot product of the corresponding vector representations. However, explaining the behavior of a matrix factorization model by direct inspection of these vector representations is difficult and such a model, albeit yielding state-of-the-art performance for knowledge base population, does not provide a user with an explanation why a certain fact has been predicted. In the following sections we introduce two human-interpretable proxy models for matrix factorization, and contrast them with each other.

2.2 First-Order Logic

As the simplest proxy to a latent variable model, we consider simple first-order implication formulae of the form $\forall x, y : r_s(x, y) \Rightarrow r_t(x, y)$, where the predicates are defined over the columns of the matrix. For example, $\forall x, y : \textit{professorAt}(x, y) \Rightarrow \textit{employeeAt}(x, y)$ states that every professor is also an employee at his or her university. Such formulae are intuitive, making them a good candidate for explaining complex latent representation models. On the other hand, their deterministic nature leads to a brittle and poor reproduction of the probability distribution over words defined by the matrix factorization model, and thus restricts their faithfulness to the original model.

2.3 Bayesian Networks (BNs)

A Bayesian network is a graphical representation of the set of conditional dependencies that hold for a joint probability distribution over a set of random variables, $P(X_1, ..., X_n)$. A local influence of a variable X_i over a variable X_j is represented as $X_i \rightarrow X_j$, and is associated with a conditional probability distribution (CPD), $P(X_j|X_i)$. Moreover, the joint probability distribution of the complete Bayesian network can be factorized as the product of local CPDs: $P(X_1, ..., X_n) = \prod_i P_i(X_i|\textit{Parents}(X_i))$ where $\textit{Parents}(X_i)$ is the set of parents of the variable X_i. For matrix factorization models, we use the columns of the matrix (the relations) as the set of random variables to discover dependencies between. This representation defines which local influences are independent of each other, making Bayesian networks easy to read. However, conditional probability tables are not intuitive to interpret for non-experts, restricting their interpretability.

2.4 Related Work

Previous work considered knowledge extraction from Artificial Neural Networks, for example (Lehmann, Bader, and Hitzler 2005; Thrun 1995; Kim et al. 2006). More recent approaches have tackled the same problem for Support Vector Machines (Barakat and Bradley 2010). In both scenarios knowledge is extracted in the form of either first-order logic or propositional formulae with the objective of behavior interpretation. One example of symbolic knowledge extraction from neural networks using a pedagogical approach is the method by (Craven and Shavlik 1996), where the neural network is used as an *oracle* in order to obtain training instances for learning a decision tree. Structure learning is often used to extract causal, interpretable structures directly from data (Sangüesa and Cortés 1997), and we employ such techniques to find a proxy model in Section 3.2.

3 Finding Interpretable Models

Three main techniques have been applied in existing work to extract knowledge from complex models: *pedagogical*, *decompositional*, and *eclectic*. In the pedagogical approach, the model is treated as a black-box that is queried with a subset of observations (such as relational facts). The objective is to model the behavior of the system without any knowledge of its internals (Craven and Shavlik 1996). In a decompositional approach the aim is to "strip" the complex model and analyze each of its parts. In the case of Artificial Neural Networks, for example, the model components consist of both the neurons (activation thresholds) and the weights (Thrun 1995). An eclectic approach is a combination of both, pedagogical and decompositional.

In this work, for simplicity and universality, we focus on the pedagogical approach, leaving decompositional and hybrid methods for subsequent investigation. We first use the matrix factorization model to predict confidence values for all the cells of the matrix. These values are then thresholded at 0.5 to yield a set of true and false facts to learn from, resulting in a "training set" D_{MF} of generated facts that is used to identify the proxy model. The obtained training data consists of 4111 random variables and 39864 true facts. We now describe how this data is used to identify appropriate models for each of the model families from Section 2.

3.1 Extracting First-Order Formulae

We learn a set of first-order formulae from the predictions of the matrix factorization model. Each formula is of the form $\forall x, y : b(x, y) \Rightarrow h(x, y)$ where the predicates/relations b and h are the body and the head of the formula, respectively. Thus, we restrict the formulae to implication rules and require arguments of the body to also be arguments of the head. To select rules we measure point-wise mutual information between body and head predicates in the training set D_{MF} generated by the MF model, and select all rules with scores above a certain threshold. Clearly other metrics are conceivable, such as probability of formula being true in the data, or scoring functions used in Inductive Logic Programming. Here we opt for the simplest method, and will investigate alternative metrics in future work.

To measure the fidelity of this model we need to evaluate its predictions under a set of observed facts. We define the set of predictions to be the transitive closure of the rules applied to the observed facts. That is, inference is done by simply propagating truth assignments of the body b_i of each rule R_i to the head h_i until no new facts can be predicted.

3.2 Extracting Bayesian Networks

Learning the structure of a Bayesian Network (BN) is NP-hard in general (Chickering 1996). This means that we can either resort to approximate algorithms, or restrict the set of possible structures. In preliminary experiments we found it extremely difficult to learn useful, interpretable BNs with approximate learning schemes, primarily due to the scale of the data (we have more than 4000 variables).

In this work we constrain the structure of the BN to be a tree. In this case, learning the structure reduces to finding a maximum spanning tree with respect to mutual information between variables (i.e. relations) in the training set D_{MF} generated by the MF model. This problem that can be solved optimally in $O(N^2)$ using Prim's algorithm, where N is the number of variables. Once we learn the structure, we use a smoothed maximum likelihood estimate for the BN parameters (the conditional probability distributions).

come_from $\Rightarrow$ *move_from*
play $\Rightarrow$ *win_against*
play $\Rightarrow$ *lose_to*
win_against $\Rightarrow$ *victory_over*
professor_at $\Rightarrow$ *expert_at*
move_to $\Rightarrow$`containedby`

Table 1: Subset of rules learned from matrix factorization

A:parent→ B:child	$p(B = 1\|A = 1)$	$p(B = 0\|A = 0)$
trip_to → *visit_to*	*0.7432*	*0.9977*
negotiate_with → *agree_with*	*0.6713*	*0.9925*
official → *chief_in*	*0.5765*	*0.9959*

Table 2: A subset of local conditional probability distributions from the Bayesian network tree learned from the matrix factorization model.

Besides being easy to learn, a BN tree is also easy to interpret in that each variable can have at most one parent, and the complete model is described using only N edges. In addition, inference in BN trees is linear in N, which makes it easy to evaluate the fidelity of the model by computing its predictions on D_{MF}. To the best of our knowledge, there is no existing work on extracting a Bayesian network representation from predictive models. We believe this may be in part due to the complexity of learning and inference in such models.

4 Experiments

To evaluate the two proxy models, we train them on predictions of a matrix factorization model for information extraction and measure mean average precision on the test set. Since the data matrix is quite sparse, we use a low threshold of 0.1 on the mutual information score for accepting a formula for the logic based proxy model, obtaining 4086 such formulae. For learning the BN tree we use mutual information as scoring function, followed by maximum likelihood estimation of local conditional probability distributions. Each surface pattern and relation appears as a node in the tree.

To illustrate the obtained proxy models, we list a subset of the extracted rules in Table 1, and show some of the local conditional probability distributions of the BN in Table 2. Figure 2 shows the precision-recall curves of each descriptive model with respect to the MF model as a measure of fidelity. Precisely, we treat the predictions on the test set from the MF model as true facts, and measure accuracy of the descriptive models in respect to them.

5 Discussion

The two main objectives in generating a proxy model from a MF model are interpretability and fidelity. These factors are often at odds with each other: More powerful model families often use complex representations that are difficult to intuit, while more interpretable models are quite restrictive.

First-order rules, as we can see from Table 1, are clearly easy to read and understand. Along with being useful for the model to explain its predictions, these rules can also be used

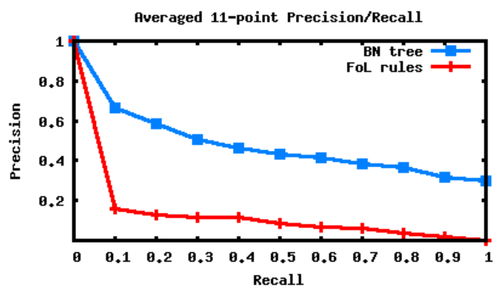

Figure 2: Fidelity of the descriptive models to the MF model.

to identify errors in the model. For example, the rules indicate the model learns an incorrect entailment of the Freebase `containedby` relation when *move_to* is observed as a surface pattern (a connection not found by the BN proxy model). On the other hand, it is not clear how much these explanations apply to the original model, since Figure 2 indicates that the rules are not necessarily an accurate representation of the original model. As we mention before, this is possibly caused by the deterministic, brittle nature of simple rules, and we would like to explore more powerful but still interpretable alternatives such as probabilistic logic in the future.

Bayesian networks, instead, learn a probabilistic, graphical representation of the original model. As we can see in Figure 2, this more expressive formulation results in a much superior predictive performance than that of the rule-based system. For interpretation, Bayesian networks provide a useful overview of the influences that the relations have over each other. On the other hand, due to their use of many potentially large conditional probability tables, Bayesian networks are harder to interpret at fine-grained level.

6 Conclusions

In this position paper we highlighted the problem of finding interpretable proxies for high-performance latent variable models. While this problem has been discussed before, we believe it is time for the community to revisit it. The reasons are both the recent successes of latent variable models, and the increasing complexity of the tasks they address. In particular, in this work we looked at matrix factorization models for knowledge base population, a more complex task than the classification problems considered in existing literature.

As the starting point we proposed two proxy representations, simple rule-based systems and Bayesian Network trees. Both provide complementary advantages, and we have only begun to investigate and contrast these. More importantly, in this work several open questions emerged. For example:

- What are good proxy representations for factorization models? BNs and rules have benefits, but there are several other options (such as MLNs, Sum-Product Networks). How does this choice of the proxy family depend on the problem domain, the original model family, expertise of the users, and their needs (debugging versus explanability)?
- How do we evaluate the *utility* of a proxy representation? Is there a natural "downstream evaluation"? How can we measure whether a proxy is useful for debugging models?
- Should a proxy representation be coupled with a means to change the predictive model according to user feedback?

In future work we will investigate these questions, and aim to address them with respect to several predictive models (such as tensor factorizations or neural tensor networks) and datasets from different domains.

7 Acknowledgments

The first author is sponsored by CONAYCT. This work was supported in part by Microsoft Research through its PhD Scholarship Programme and in part by the TerraSwarm Research Center, one of six centers supported by the STARnet phase of the Focus Center Research Program (FCRP) a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

Barakat, N., and Bradley, A. P. 2010. Rule extraction from support vector machines: a review. *Neurocomputing* 74(1):178–190.

Chickering, D. M. 1996. Learning bayesian networks is np-complete. In *Learning from data*. Springer. 121–130.

Craven, M. W., and Shavlik, J. W. 1996. Extracting tree-structured representations of trained networks. *Advances in Neural Information Processing Systems (NIPS-8)* 24–30.

Kim, H.; Yoon, T.-S.; Zhang, Y.; Dikshit, A.; and Chen, S.-S. 2006. Predictability of rules in hiv-1 protease cleavage site analysis. In *Computational Science (ICCS)*. 830–837.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42:30–37.

Lehmann, J.; Bader, S.; and Hitzler, P. 2005. Extracting reduced logic programs from artificial neural networks. In *IJCAI Workshop on Neural-Symbolic Learning and Reasoning*.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2005. Blog: Probabilistic models with unknown objects.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62:107–136.

Riedel, S.; Yao, L.; Marlin, B. M.; and McCallum, A. 2013. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Sangüesa, R., and Cortés, U. 1997. Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks. *AI Communications* 10(1):31–61.

Thrun, S. 1995. Extracting rules from artificial neural networks with distributed representations. *Advances in neural information processing systems* 505–512.