Extracting Interpretable Models from Matrix Factorization Models

Ivan Sanchez Carmona
Department of Computer Science
University College London
i.sanchezcarmona@cs.ucl.ac.uk

Sebastian Riedel
Department of Computer Science
University College London
s.riedel@cs.ucl.ac.uk

Abstract

Matrix factorization models have been successfully used in many real-world tasks, such as knowledge base completion and recommendation systems. However, explaining the causes that elicit a particular prediction by a manual inspection of its latent representations is a difficult task. In this paper we try to overcome this problem by exploring descriptive model classes in their ability to faithfully approximate the behavior of a pre-trained matrix factorization model. Crucially, our choice of descriptive model will allow us to provide an interpretable structured proof for each prediction of the original model. We compare the descriptive models in these two scopes: Fidelity and interpretability. We find that Bayesian network trees, a class of models that has not been considered for this purpose before, capture the matrix factorization model faithfully while providing multi-step explanations of predictions.

1 Introduction

A highly desirable property of a predictive system is the ability to provide an interpretable explanation of a particular prediction. Matrix factorization (MF) models, due to their advantages (high accuracy and scalability), have been used in real-world tasks such as recommendation systems [1], and knowledge base population [2]. These models are a type of latent variable model (LVM) where a set of latent vectors is learned from a matrix of relational data. However, they are deficient in providing an explanation of the relations among observed variables that elicit a particular prediction. This lack of interpretability prevents them from both providing a support for the predictions and from analysing errors. These two properties would not only benefit NLP tasks, but also decision-support tasks, such as credit-risk evaluation [3], and medical diagnosis.

One solution to this problem is to focus on designing more interpretable models from the onset such as [4], but this often means trading off accuracy and scalability. Another option is to use visualization methods [5], where high-dimensional vectors are projected into a two-dimensional space. Such methods are useful for model inspection, but it is unclear how they can provide fine-grained multi-step explanations of a prediction in the way that, say, rule-based systems can. We believe this granularity is important to spot higher level problems in the model that go beyond "one vector is too close to another."

Here we investigate an alternative option: Learn an interpretable descriptive model that *mimics* the behavior of the original LVM model as close as possible. While we still use the original LVM to make predictions, we use the descriptive model to explain these predictions. Our goal then is to find descriptive models for (so-called) *donor* LVMs that *faithfully* capture the idiosyncrasy of the donor LVM (they respond similarly to same inputs), while their anatomy remains *interpretable*.

Our starting point is work by [6] on extracting decision trees (DT) from neural networks. We aim to apply a variant of their method to the MF model of [2] to get DTs that can be used to explain

extracted relations. We identify two core problems with this approach: *1*) we can capture some of the decision boundaries of the MF model, but we cannot capture its *ranking* behaviour which is most relevant for applying the model in practice; *2*) the [2] model is a full *joint* model over a *universal schema* of both surface form relations such as *X-a-professor-at-Y* and Freebase relations such as *employee(X,Y)*; representing each of these thousands of relations with an own decision tree both makes the model harder to interpret, and less faithful as far as model structure is concerned.

To overcome the problems above we propose to use a class of descriptive models that have not been used before for this purpose: tree-structured Bayesian networks (BN) [7]. Such BNs are interpretable, as their sparse connectivity explains the most important correlations. As joint probabilistic models they are able to capture both the joint nature of the donor MF model, and its ranking behaviour. But in contrast to more complex probabilistic models, they are much easier to train.

We *quantitatively* compare the learned BN trees to two baselines: A variant of [6] and a method to extract logical rules from the MF model. We compare *fidelity* of the descriptive models by measuring how well their rankings match the ranking of the original MF model on test data. Fidelity is a crucial metric for a descriptive model: It may be very interpretable, but without high fidelity it explains the wrong behaviour.

We *qualitatively* compare our approach to the baselines by contrasting the explanations of two wrong predictions of the MF model. In the BN tree these explanations are represented as subgraphs that span from observed nodes to the node predicted (Figure 2).

2 Background

We describe the predictive model (MF) and the descriptive models (BN tree, logic rules, decision trees).

2.1 Matrix Factorization

Objectives in a MF formulation are a) to learn a low-rank matrix $X_{m\times n} = UV'$ that reconstructs a given matrix of relational data $Y_{m\times n}$, where $U_{m\times k}$ and $V_{n\times k}$ are latent factors, and b) to predict confidence values for unobserved cells (matrix completion). We use a variant of the MF model in [2], where each row of the matrix Y corresponds to a pair of entities (e.g. $(London, England)$), and each column corresponds to either a surface form or a Freebase relation (e.g. $capitalOf$). The matrix X is learned by minimizing a logistic loss function over data. Each latent vector U_p is a distributed representation of a pair of entities. Similarly, each vector V'_r represents a relation. The latent vectors of pair p and relation r define the prediction $x_{r,p} \in [0, 1]$ of the fact that relation r holds for pair p as $x_{r,p} = sigmoid(U_p V'_r)$.

An alternative view of a matrix factorization model is as a one-hidden-layer neural network [8], where the activation function is linear in the hidden neurons and non-linear (sigmoid) in the output layer. The latent factors U, V correspond to the parameters of the network, i.e., the weights in the hidden and output layers. The model can be re-written as $\mathbf{y} = sigmoid(UV'\mathbf{x})$, where $\mathbf{x}$ is a one-hot input vector and $sigmoid$ is a component-wise function.

2.2 Logic Rules

We choose implication rules as a baseline due to their comprehensibility and to the maturity of rule extraction algorithms [9]. The rules learned are of the form $\forall x, y : A(x, y) \Rightarrow B(x, y)$, where *A, B* are predicates. All rules are range restricted (arguments occur in both body and head predicates). A rule, as the simplest building block in an explanation, accounts for the cause of predicting the realization of a fact. For example, the rule *professorAt*$(x, y) \Rightarrow$ *employeeAt*(x, y) would explain the cause of predicting the fact that x is an *employeeAt* y by observing the realization of the body of the rule as a true fact.

2.3 Decision Trees

A decision tree [10] is a hierarchical classification model. Each internal node corresponds to an input variable. The root node is the highest in the hierarchy (it splits first the input space). Leaf

nodes represent the class decision. We choose DTs as a baseline due to their main properties: a) interpretability (a path in the tree can be seen as a conjuctive logic rule), and b) suitability for estimating class probabilities (maximum likelihood estimation in each leaf).

2.4 Bayesian Networks

Decision trees can capture, to some extent, the probabilistic nature of the MF model, but they do not define a joint model across the complete relational schema. A set of logical rules can provide a fuller picture, and a notion of a proof that covers reasoning in various parts of the schema, but they cannot simulate the ranking behaviour of the MF model well.

We think that Bayesian networks can overcome both issues. A BN is the set of conditional independencies holding for a set of random variables in the form of a directed acyclic graph. A node corresponds to a random variable. A directed edge between two nodes, $x_i \rightarrow x_j$, represents a local influence which defines a conditional probability distribution (CPD): $p(x_j|x_i)$ (a parameter of the BN). A BN encodes a factorized probability distribution over the variables of the form $p(x_1, ..., x_n) = \prod_i p_i(x_i|\textit{Parents}(x_i))$, where *Parents*$(x_i)$ is the set of nodes that have an outgoing edge to x_i, (e.g. Figure 2).

3 Related Work

Learning descriptive models from complex donor models such as neural networks [11, 12, 13, 14] and support vector machines [15] has been previously considered. They extract both a set of logic rules and decision trees with the objective of behavior explanation. Nevertheless, the methods used are not suitable for the donor model we propose due to structural constraints. Moreover, to the best of our knowledge, this is the first time a joint probability model (Bayesian network) is compared with a classifier (decision trees) with the purpose of prediction explanation for a matrix factorization model.

4 Learning Interpretable Models

We treat the MF model as both a joint probability model and a classifier due to the nature of the descriptive models. To learn a BN tree and a set of logic rules we take each row of the MF model as a training instance and each colum as a variable. To learn decision trees we take input vectors from the training data used for training the original MF model and attach them with class labels from the MF predictions, i.e., the MF model re-labels input instances.

4.1 Extracting Logic Rules

We used mutual information as a support measure for rule acceptance. A rule is accepted if the strength of dependency between two predicates (columns in the MF model) surpasses a threshold m (we manually selected m=0.1 based on predictive performance). The directionality of each rule is then determined by its confidence, *if* $p(B|A) > p(A|B)$*then* $A \Rightarrow B$ *else* $B \Rightarrow A$. We opted for an information-theoretic based measure due to its property of assigning monotonically increasing values to more statistically dependent variables. We refrained from using an inductive logic programming approach due to its requirement of negative examples [16]. In order to obtain a test prediction we applied a transitive closure over a set of observed facts (i.e., we apply modus ponens).

4.2 Extracting Decision Trees

We used the R package *rpart* [17] in order to extract decision trees. Learning a decision tree corresponds to recursively adding nodes to its hierarchy, partitioning the input space. Node selection is performed by minimizing a cost function that measures the number of instances correctly classified after partitioning. In each leaf of the tree maximum likelihood estimation is performed in order to compute the probability of the class variable given the input variables in the path from the root node to the leaf. Once the tree has been learned, classifying a test instance corresponds to traversing the tree along the path that matches the observed facts until a leaf node is reached.

4.3 Extracting Bayesian Networks

Learning the structure of a BN is NP-hard [18]. This means that we can either resort to approximate algorithms, or restrict the model class. 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. Therefore, we constrained the structure of the BN to be a tree.

Learning the structure of such BNs reduces to finding a maximum spanning tree with respect to mutual information between variables (columns in the MF model). This problem can be solved optimally in $O(N^2)$ using Prim's algorithm, where N is the number of variables. Parameter estimation is performed by smoothed maximum likelihood estimation (we share parameters accross training instances).

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 test data.

5 Experiments

We use the predictions from the MF model as training data for learning the descriptive models: We predict a confidence value for each cell in the MF model and threshold them at t=0.5 (we tried for different t in [0,1]). Training datasets are of size 4111 variables by 39864 instances. We obtained 4572 logic rules, a BN tree with 4111 nodes, and 19 decision trees (we chose 19 target variables) with average depth of 5 nodes. We evaluated the descriptive models on the test set of [2]. We performed two types of model analyses: Fidelity and interpretability.

Fidelity Figure 1a shows the 11-point average precision curves of the descriptive models with respect to the predictions of the MF donor model. We see that the logic rules learned are not a faithful representation of the predictive model: The ranked list of facts produced by the logic rules poorly matches the predictions of the MF model. This might be due to their deterministic nature: Since their response is in a binary domain they are not able to provide a confidence value in [0,1]. This makes it difficult to capture the ranking behavior of the MF model.

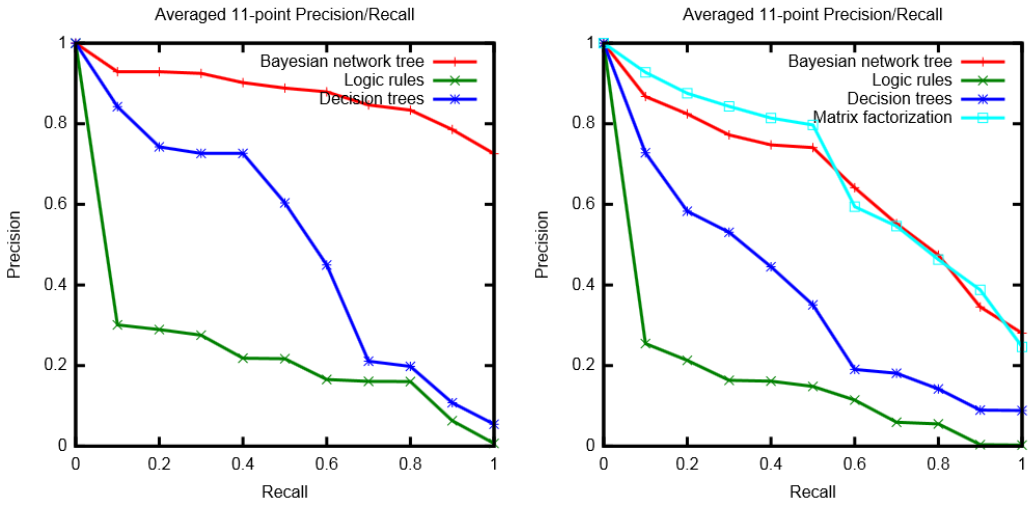
The decision trees provide more sensible confidence scores and hence rankings. This is reflected in better average precision curves. The BN model outperforms the other models substantially. We believe that this is a consequence of its probabilistic formulation, and the ability to capture the joint nature of the MF model better.

We also compute generalization performance of the descriptive models on a gold test sample and compare against generalization performance of the MF donor model. Figure 1b shows how low fidelity models (logic rules, decision trees) generalize poorly whereas high fidelity models (Bayesian network tree) have a generalization performance comparable to the original donor model. This confirms that the descriptive models approximates the MF model well.

Interpretability We show two examples of *explanations* for wrong predictions as produced by the descriptive models. Figure 2a shows possible causes for the MF model predicting the wrong fact $arenaStadium(PhiladelphiaEagles, Canton)$ as a true fact with confidence of 0.885: The observed node, $playAt$, influences the next nodes in the trajectory towards the target node $arenaStadium$. A clear error of the MF model is indicated by the connection in the BN: $beatAt \rightarrow arenaStadium$ (a team x beating another team at arena y does not necessarily entails that y is its home stadium).

Following the above example, the explanation from the decision tree learned for the relation $arenaStadium$ is the rule: *if* $playAt = 1$ *then* $p(arenaStadium = 1) = 1.0$. We think that the interpretability of the DT and the BN are quite comparable in this case. By contrast, the logical system does not even predict this fact as it did not recover the $playAt(x, y) \Rightarrow arenaStadium(x, y)$ implication.

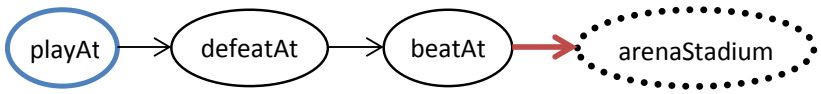
Figure 2b shows the explanation for the MF model wrongly predicting as a true fact: $reviewMovie(DanielKahneman, Nobel)$ (Kahneman is a Nobel price winner, not a reviewer for



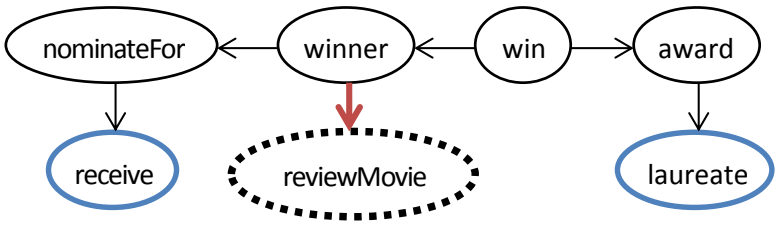
(a) Fidelity of the descriptive models to the matrix factorization model.

(b) Generalization of the descriptive models and the MF model on gold test data.

Figure 1: Fidelity and generalization of the descriptive models.



(a) variable predicted: $arenaStadium$



(b) variable predicted: $reviewMovie$

Figure 2: Two snippets from the BN tree: Influences from observed to predicted variables as an explanation of the causes that elicited a wrong prediction by the MF model. Bold arrows: wrong influences, bold nodes: observed variables.

a movie called $Nobel$). Given that $reviewMovie$ is not one of the 19 target variables no decision tree was learned for it, so no explanation from this model can be sought. On the other hand, in the set of logic rules none of the observed variables appeared, meaning that their statistical dependence with respect to other variables was low.

6 Conclusion

The problem of finding interpretable descriptive models for latent variable models has been discussed before. But we believe it is time to revisit it due to their recent successes and the increasing complexity of the tasks they address. In this work we looked at matrix factorization models for knowledge base population, a more complex task than the classification problems considered in ex-

Table 1: CPDs for Figure 2a.

A:parent→ B:child	$p(B=1\|A=1)$	$p(B=0\|A=0)$
$playAt \rightarrow defeatAt$	0.8651	0.9978
$defeatAt \rightarrow beatAt$	0.8435	0.9999
$beatAt \rightarrow arenaStadium$	0.8186	0.9989

isting literature. We proposed Bayesian network trees as a descriptive model and compared to two baselines: logic rules and decision trees. We found that BN trees provide a very competitive combination of fidelity and interpretability outperforming the baselines. We believe this model is prone to be used for analysing the latent variable model model by spotting wrong edges in its structure. In the future we like to investigate further representations, develop better ways to quantitatively evaluate the *utility* of a descriptive model, and apply the approach to other LVM models in NLP (such as as the matrix factorization formulation of the word2vec model).

Acknowledgments

The first author is sponsored by CONACYT. The second author is sponsored in part by the Paul Allen Foundation through an Allen Distinguished Investigator grant and in part by a Marie Curie Career Integration Fellowship. Thanks to Ivan Meza Ruiz for helpful discussions and to the anonymous reviewers who provided insightful comments.

References

[1] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009.

[2] Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 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)*, June 2013.

[3] Bart Baesens, Rudy Setiono, Christophe Mues, and Jan Vanthienen. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science*, 49(3):312–329, 2003.

[4] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. 2013.

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *The Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[6] Mark W Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. *Advances in Neural Information Processing Systems (NIPS-8)*, pages 24–30, 1996.

[7] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[8] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *Proc. 3rd ICLR*, 2015.

[9] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: Concepts and techniques, 3rd edition. 2011.

[10] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[11] Jens Lehmann, Sebastian Bader, and Pascal Hitzler. Extracting reduced logic programs from artificial neural networks. In *IJCAI Workshop on Neural-Symbolic Learning and Reasoning*, 2005.

[12] Sebastian Thrun. Extracting rules from artificial neural networks with distributed representations. *Advances in neural information processing systems*, pages 505–512, 1995.

[13] Hyeoncheol Kim, Tae-Sun Yoon, Yiying Zhang, Anupam Dikshit, and Su-Shing Chen. Predictability of rules in hiv-1 protease cleavage site analysis. In *Computational Science (ICCS)*, pages 830–837. 2006.

[14] AS d'Avila Garcez, Krysia Broda, and Dov M Gabbay. Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125(1):155–207, 2001.

[15] Nahla Barakat and Andrew P Bradley. Rule extraction from support vector machines: a review. *Neurocomputing*, 74(1):178–190, 2010.

[16] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422. International World Wide Web Conferences Steering Committee, 2013.

[17] Terry M Therneau, Beth Atkinson, Brian Ripley, et al. rpart: Recursive partitioning. *R package version*, 3:1–46, 2010.

[18] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from data*, pages 121–130. Springer, 1996.