Identifiability of Model Properties in Over-Parameterized Model Classes

Manfred Jaeger

Aalborg University, Denmark
jaeger@cs.aau.dk

Abstract. Classical learning theory is based on a tight linkage between hypothesis space (a class of function on a domain X), data space (function-value examples $(x, f(x))$), and the space of queries for the learned model (predicting function values for new examples x). However, in many learning scenarios the 3-way association between hypotheses, data, and queries can really be much looser. Model classes can be over-parameterized, i.e., different hypotheses may be equivalent with respect to the data observations. Queries may relate to model properties that do not directly correspond to the observations in the data. In this paper we make some initial steps to extend and adapt basic concepts of computational learnability and statistical identifiability to provide a foundation for investigating learnability in such broader contexts. We exemplify the use of the framework in three different applications: the identification of temporal logic properties of probabilistic automata learned from sequence data, the identification of causal dependencies in probabilistic graphical models, and the transfer of probabilistic relational models to new domains.

1 Introduction

This paper is originally motivated by ongoing research in learning probabilistic automata models for applications in model-based design in verification [16, 10]. In model-based design, various forms of finite probabilistic automata models are used to model hard- or software systems. Relevant properties of the system are expressed using a formal, logical representation language, such as probabilistic linear time logic (PLTL), or probabilistic computation tree logic (PCTL) [2]. Efficient algorithms exist to check whether such a property is satisfied by a given automaton model, i.e., whether the design or actual system represented by the model satisfies a certain specification. Traditionally, the formal models used in this process are constructed manually. This, however, can be a very time-consuming and error-prone process. We are therefore interested in the possibility of automatically learning finite automata models from data consisting of observations of visible system behaviors. Adapting standard automata learning algorithms [3, 4] we obtain learning methods for our application that come with certain convergence guarantees for the large-sample limit. However, these convergence guarantees do not directly imply what is of ultimate interest, namely,

that in the large-sample limit the learned model will agree with the actual observed system on properties representable in the formal representation language we use in model-checking. Concretely, building on the given convergence guarantees, one can show that the learned model will in the limit agree with the actual system on PLTL properties [10]. However, the same does not appear to be true for PCTL properties.

Abstracting from this specific learning problem, we are faced with the more general question: what classes of properties that we will want to query our learned model for, can, in principle, be learned from the observations that are represented by our data? This question is closely connected to *learnability* in the sense of computational learning theory, as well as to *identifiability* in the sense of statistics. However, it seems that neither these two existing theoretical frameworks are quite sufficient to analyze the scenario we are here considering.

Computational learning theory uses at its conceptual foundation a very close linkage between *hypothesis space*, *data space*, and what one may call *query space*: the hypothesis space is taken to consist of a set $\mathcal{F}$ of functions, data consists of a set of observed pairs $(x, f(x))$ of arguments and function values, and a learned function $f \in \mathcal{F}$ will be queried for its function values $f(x)$. This setup does not incorporate the possibility of an over-parameterized hypothesis space, i.e., the existence of two distinct hypotheses h, h' that define the same function, and that therefore would lead to equivalent data observations. Under the assumption that a learned hypothesis will be queried for its function values, this possibility may also be safely neglected, since it would make no difference whether hypothesis h or h' is learned. This radically changes, however, when also the close linkage between data space and query space is lost, and the learned hypothesis will be queried for properties that may not exactly match the type of observations found in the data. This is precisely the situation we find ourselves in when learning probabilistic automata for model-checking purposes: two distinct automata can induce the same data-distribution of observable behaviors, but differ with respect to some properties in our formal query languages. On the other hand, some relaxation of the three-way linkage between data, hypotheses and queries does not necessarily preclude learnability: in our positive results about learning PLTL properties we have data consisting of finite strings, hypotheses consisting of finite automata, and queries consisting of logical formulas.

The issue of over-parameterized model spaces containing distinct hypotheses that generate indistinguishable data is exactly the subject of the statistical notion of identifiability. However, statistical identifiability theory does not relate hypothesis and data space to classes of queries over the learned model. It is implicitly assumed that the purpose of learning, which here comes down to parameter estimation, is to identify the true parameter. In contrast, we may be satisfied with learning a hypothesis that is distinct from the "true" one, as long as it is equivalent with regard to a certain class of query properties.

The problem of learnability of certain query properties in an over-parameterized model class is a quite common one, and certainly not limited to, or first encountered in, our problem of identifying logical properties of a finite automa-

ton. Another motivating example of the same problem which we shall consider in this paper is *causal discovery* from observational data: a directed graphical model (Bayesian network) is sometimes regarded as a causal model, where directed connections between random variables represent causal dependencies. However, it is well known that a Bayesian network learned from statistical data only allows a limited interpretation as a causal model, since networks with different directed edge structures can induce the same data distribution, and hence be indistinguishable based on observational data. The possibilities and inherent limitations of using Bayesian network learning for causal discovery are now well understood [9]. However, the sometimes controversial debate into this issue has not been fully phrased within a formal theory of learnability or identifiability, which, one could imagine, sometimes might have helped to elucidate matters more clearly [15, 6].

With an increasing ambition of learning models in more and more expressive model classes, for example probabilistic programming languages [12, 7, 5], one also encounters more and more complex relationships between the wide range of model properties that could be queried, and the empirical content of the original data. Broadening existing theories of learnability to enable a systematic and principled analysis of these relationships and dependencies, thus, could be useful in a variety of contexts.

In this paper we are going to propose a formal framework combining elements of computational learnability and statistical identifiability that enables a systematic study of what model properties can, or can not, be identified in a certain model class, given a particular type of training data. To this end we first introduce a very general setup of learning as maximization of a score function (Section 2). We then propose a definition of *identifiability* that makes no assumptions on structural correspondences between data-, model-, and query space. Based on these definitions, we obtain a first theorem about non-identifiability. We demonstrate the applicability of our framework by deriving non-identifiability results for PCTL properties of probabilistic automata, directed edge relationships in Bayesian networks, and probabilistic queries on varying domains in statistical relational models. Finally, in Section 4 we adapt the initial very general framework for the special case of statistical learning, and discuss its relationship with PAC learnability [18].

2 Learning as Score Maximization

We characterize learning as the maximization of a score function σ that defines for each M in a model (or hypothesis) space $\mathcal{M}$, and each dataset D in a dataspace $\mathcal{D}$ a score $\sigma(M, D)$. This, on the one hand, is the most natural description of a wide range of learning methods that in fact operate by using heuristic or stochastic search to maximize a given score function. On the other hand, it also accommodates in a trivial way any other algorithmic learning procedure not based on an explicit score function by representing an algorithm l that on input

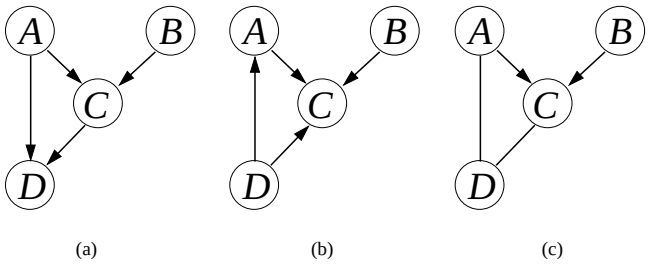

Fig. 1. Two Bayesian Networks and their Essential Graph

D outputs a hypothesis $l(D) \in \mathcal{M}$ via a score function σ_l with $\sigma_l(M, D) = 1$ iff $M = l(D)$, and $\sigma(M, D) = 0$ otherwise.

One restriction we impose on the scoring function σ is that for all $D \in \mathcal{D}$ the supremum of $\sigma(\cdot, D)$ is attained for some $M \in \mathcal{M}$. We write

$$\mathcal{M}(\sigma, D) := \{M \in \mathcal{M} \mid \sigma(M, D) = \max_{\tilde{M} \in \mathcal{M}} \sigma(\tilde{M}, D)\}$$

for the set of all M maximizing $\sigma(\cdot, D)$. Thus, $\mathcal{M}(\sigma, D)$ is the set of models learned from D using σ. $\mathcal{M}(\sigma, D)$ may contain more than one element, but is assumed to be nonempty.

In the following we give two examples of learning algorithms not usually seen from the perspective of score-based learning. We here develop a more meaningful characterization of these learning methods in terms of score maximization than simply by the trivial σ_l representation mentioned above.

Example 1. Bayesian Networks are probabilistic models for the joint distribution of a set of random variables [13]. They consist of a directed acyclic graph over the random variables, and, for each variable, the specification of the conditional probability distribution of that variable given its parents in the graph. Figure 1 (a),(b) shows two different Bayesian Networks for random variables A, B, C, D. The graph structure of a Bayesian network encodes conditional independence relations among the variables.

Probabilistic automata represent probability distributions over infinite sequences of symbols from a given alphabet Σ. Figure 2 shows three different probabilistic automata defining probability distributions on $\{a, b\}^\infty$. In the figure, states marked with * are the initial states of the automaton. Solid edges denote state transitions taken with probability 1, and dashed edges transitions with probability 1/2. All automata in Figure 2 define the same probability distribution P_M, which is the uniform distribution on $aa\{a, b\}^\infty$.

M_a and M_b are deterministic automata: for every state in the automaton, and every $s \in \Sigma$ there exists at most one possible successor state labeled with s. M_c, on the other hand, is non-deterministic, since the initial state has two different a-successors.

A classic algorithm for learning Bayesian networks is the PC-algorithm [17], and a standard algorithm for learning deterministic probabilistic automata is the Alergia algorithm and its variants [3, 4, 8, 10].

Even though quite different with respect to the learning tasks they solve, and the algorithmic details, the PC algorithm and Alergia share some common features which can be expressed in a common structure of a high-level score function representation of these approaches. Both PC and Alergia identify the graphical structure of the model based on statistical tests performed on the data. In the PC algorithm, these are conditional independence tests for subsets of the random variables. In Alergia, one performs tests for the identity of the conditional distributions on (infinite) suffixes $\boldsymbol{s} \in \Sigma^\infty$ given different (finite) prefixes from Σ^*.

In Alergia, the outcome of the statistical tests determines the structure of the model uniquely. In the PC-algorithm, the tests determine the structure only up to membership in a class of network structures encoding the same conditional independence constraints. This equivalence class can be represented by an *essential graph*, which is a mixed graph with directed and undirected edges. Figure 1 (c) shows the essential graph representing the equivalence class consisting of the networks (a) and (b). In order to obtain a final directed graph, one may employ any method to select a representative directed graph from the equivalence class.

Once the model structure is fixed, maximum likelihood parameters are fitted. The overall learning process can thus be described as maximizing a score function of the form

$$\sigma(M, D) = R(M) \cdot \mathit{Tests}(M, D) \cdot \mathit{Lhood}(M, D) \tag{1}$$

where $\mathit{Tests}(M, D)$ is a 0/1-valued function that evaluates to 1 iff the structure of M is consistent with the outcome of all relevant statistical tests performed on D, and $\mathit{Lhood}(M, D) = P_M(D)$ is the likelihood function (also depending on the parameters of M). $R(M)$ is a factor only needed for capturing the PC algorithm. It is a another 0/1-valued function that evaluates to 1 iff the structure of M is the representative directed graph for its underlying essential graph. For Alergia, we may just assume that $R(M)$ is constant equal to 1.

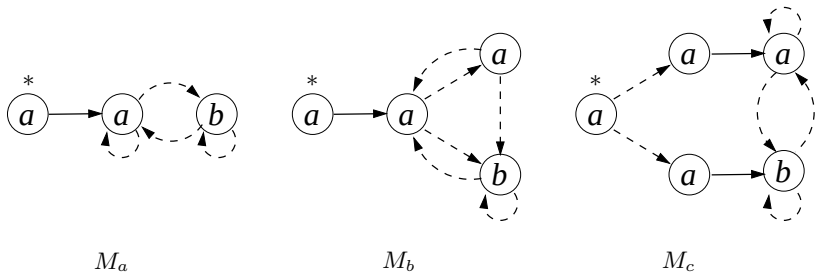

Fig. 2. Probabilistic Automata

A second example looks at a more standard form of score-based learning.

Example 2. (Penalized likelihood score (PLS)) Standard model scores like Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), or Minimum Description Length (MDL) are used for a wide range of probabilistic model classes. They are penalized likelihood scores of the form

$$\sigma_{PLS}(M, D) := c \cdot \log(Lhood(M, D)) - g(|\, D\,|)h(|\, M\,|), \tag{2}$$

where c is a constant, $g(|\, D\,|)$ is a function of the size of the data D, and $h(|\, M\,|)$ is a function of the size of M, where g and h are non-negative, and h is monotone in $|\, M\,|$. For Bayesian networks, as well as for many other probabilistic types of models, the size $|\, M\,|$ typically is defined as the number of free parameters in M.

Assuming that the two network structures (a) and (b) in Figure 1 are equipped with their maximum likelihood parameters, they will both define the same distribution P_M, and, thus, obtain the same log-likelihood score $\log(P_M(D))$.

3 Identifiability

We are interested in learnability in the limit of large datasets. Apart from datasets consisting of independent samples, we also want to consider datasets that may consist, e.g., of a single long sequential observation of a temporal process, or a (large) graph. We therefore only assume that the space of all datasets is structured as follows.

Definition 1. *A* stratified dataspace *is a set*

$$\mathcal{D} = \cup_n \mathcal{D}^{(n)}$$

which is equipped with a partial order relation $\prec$, so that $D \prec D'$ implies that $D \in \mathcal{D}^{(j)}$, $D' \in \mathcal{D}^{(k)}$ with $j < k$. Furthermore, it is required that for every n and $D \in \mathcal{D}^{(n)}$ there exists $D' \in \mathcal{D}^{(n+1)}$ with $D \prec D'$.

The intention is that $D \prec D'$ means that D' is an extension of the dataset D.

Example 3. (a) Suppose the data consists of independent samples $\boldsymbol{s} = s_1, \ldots, s_n$ from a sample space S. Then $\mathcal{D}^{(n)} = S^n$, and $\boldsymbol{s} \prec \boldsymbol{s}'$ iff $\boldsymbol{s}'$ extends $\boldsymbol{s}$, i.e., $\boldsymbol{s} \in S^n$, $\boldsymbol{s}' \in S^{n'}$ with $n' > n$, and $s'_i = s_i$ for $i = 1, \ldots, n$.

(b) In learning from a single sequence of symbols from an alphabet Σ, we have $\mathcal{D}^{(n)} = \Sigma^n$, and $D \prec D'$ iff D is a prefix of D'.

(c) When learning from graphs (or, more generally, colored graphs, or relational data), we can have that $\mathcal{D}^{(n)}$ is the set $\mathcal{G}^n$ of all graphs with n nodes, and $G \prec G'$ iff G is embedded as a sub-graph in G'.

Learning in the limit of large datasets now is analyzed in terms of increasing sequences $D_1 \prec D_2 \prec \cdots$. We write $\uparrow D_n$ for such a sequence.

As in (1) and (2), most score functions will be composed from a number of simpler scores or measurements:

Definition 2. *A* score feature *is any function*

$$F: \mathcal{M} \times \mathcal{D} \to \mathcal{F}$$

with $\mathcal{F} \subseteq \mathbb{R}$. *When* $\mathcal{F} = \{0, 1\}$ *then* F *is called a Boolean feature. A score feature* $F: \mathcal{M} \to \mathcal{F}$ *that only depends on* $\mathcal{M}$ *also is called a* model feature*, and a feature* $F: \mathcal{D} \to \mathcal{F}$ *that is only a function of* $\mathcal{D}$ *is called a* data feature. *A* query space *is a set* Φ *of Boolean model features.*

The score function $\sigma(\cdot, \cdot)$ itself is a score feature. For probabilistic models, a key score feature is the likelihood $Lhood : (M, D) \mapsto P_M(D)$.

For a query space Φ we write $M \equiv_\Phi M'$ if $\phi(M) = \phi(M')$ for all $\phi \in \Phi$. $[M]_{\equiv_\Phi}$ denotes the equivalence class of M in $\mathcal{M}$ with regard to the equivalence relation $\equiv_\Phi$. We are now ready to introduce our central definition.

Definition 3. *Let* $\mathcal{M}$ *be a model space,* $\mathcal{D}$ *a stratified dataspace, and* Φ *a query space for* $\mathcal{M}$. *Let* σ *be a score function on* $\mathcal{M} \times \mathcal{D}$. *We say that* Φ *is* $(\mathcal{D}, \sigma)$*-identifiable in* $\mathcal{M}$*, if for all* $M \in \mathcal{M}$ *there exists* $\uparrow D_n$ *so that for all* $\phi \in \Phi$ *there exists* $n_\phi \in \mathbb{N}$*, so that for all* $n \geq n_\phi$*:*

$$\mathcal{M}(\sigma, D_n) \subseteq [M]_{\equiv_\phi} \tag{3}$$

Definition 3, thus, demands that for every possible M there exists some ideal data sequence $\uparrow D_n$ which enables us to obtain from the learned models in the large sample limit the same answers for all queries ϕ that we would obtain from the "true" model M.

Definition 3 only refers to Boolean queries. In many cases, however, we will also be interested in querying other kinds of model features. For probabilistic models M, for instance, a typical query can be the probability $P_M(E)$ assigned by the model to some event E, or the most probable configuration of a set of unobserved random variables. Identifiability of queries of this kind can be approximated by identifiability of suitable Boolean query spaces. For probability queries $P_M(E)$, for instance, one should only require that the answers obtained from the learned models converge to the true value in the large sample limit. Such a convergence is exactly captured by Definition 3 as identifiability of the Boolean features $P_M(E) \in I$, for all open intervals I.

Definition 3 only provides a basis for analyzing identifiability questions. Neither positive nor negative identifiability results purely in the sense of this definition are very interesting per se. A positive identifiability result in the sense of Definition 3 is not particularly useful in itself, since it would only say that based on some ideal data sequence $\uparrow D_n$ we would be able to identify Φ. To be practically relevant, we would need the sharper result that this will be true for all data sequences that in some sense are sufficiently informative, or representative of the true model, and that are the datasets we are likely to have for learning in practice. We will extend Definition 3 in this direction in Section 4.

The main focus of this paper, however, is to establish negative identifiability results, i.e., impossibility results for learning. A negative result just in the sense

of Definition 3 would not be very strong either, since it would only tell us that learning with a particular score function σ does not enable us to identify Φ. For non-identifiability results, however, we are interested in whether it is impossible to identify Φ using any member in a certain class of score functions, which represents all in some sense applicable learning algorithms. In fact, one may wonder why Definition 3 refers to a score function at all. Based on our original motivation described in Section 1, one may rather want to describe identifiability as an inherent relationship between the model, query, and data spaces, so that identifiability just means that the data contains a sufficient amount of query-relevant information about the model – which should be independent of any learning procedures. However, as the following example illustrates, it would likely be futile to try to address the identifiability problem without explicit consideration of admissible score functions, since otherwise one can always obtain trivial identifiability results via artificial score functions that do not capture realistic learning methods, but rather describe querying a perfect oracle:

Example 4. Assume that the cardinality of $\mathcal{D}$ is at least as big as the cardinality of $\mathcal{M}$, so that there is a one-to-one function $f : \mathcal{M} \to \mathcal{D}$. Then we can define

$$\sigma(M, D) := \begin{cases} 1 \text{ if } D = f(M) \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

For every $M \in \mathcal{M}$ then $\mathcal{M}(\sigma, f(M)) = \{M\}$, and (3) is satisfied by the constant sequence $D_n = f(M)$.

In the following, thus, we will derive non-identifiability results in the sense of Definition 3 for all σ in certain classes of score functions. We specify classes of score functions that correspond to realistic classes of learning algorithms in terms of the score features that they are based on. We write $\sigma(F_0(M, D), \ldots, F_k(M, D))$ for a score function σ that depends on M, D only through the features $F_i(M, D)$ (which may also include pure model or data features).

Definition 4. *A score function* $\sigma(F_0(M, D), \ldots, F_k(M, D))$ *is* monotone in *the score feature* F_0*, if for all* $f_i \in \mathcal{F}_i$ *(*$i = 1, \ldots, k$*), and* $r, s \in \mathcal{F}_0$*:* $r \leq s \Rightarrow \sigma(r, f_1, \ldots, f_k) \leq \sigma(s, f_1, \ldots, f_k)$.

Example 5. The PC/Alergia-learning score function (1) is monotone in all its features $R(M)$, $\mathit{Tests}(M, D)$, and $\mathit{Lhood}(M, D)$. The PLS score function (2) is monotone in $\mathit{Lhood}(M, D)$, as well as $-h(|\, D \,|)$ and $-h(|\, M \,|)$.

Both (1) and (2) are monotone in a specific model feature ($R(M)$, respectively $-h(|\, M \,|)$) that represents a preference or bias function on $\mathcal{M}$. Such a feature can express a bias towards small or canonical models, or, in a Bayesian framework, a prior probability of M.

In the following we pay particular attention to such bias features expressing prior knowledge or preferences, and we write $\sigma(B, \boldsymbol{F})$ for a score function that is monotone in a designated numerical bias model feature B, and that further depends in an arbitrary manner on other score features $\boldsymbol{F} = F_1, \ldots, F_k$.

For any subsets $\boldsymbol{M} \subseteq \mathcal{M}$, $\boldsymbol{D} \subseteq \mathcal{D}$ and feature vector $(F_0, \ldots, F_k)$ we write $(F_0, \ldots, F_k)(\boldsymbol{M}, \boldsymbol{D})$ for the set $\{(F_0(M, D), \ldots, F_k(M, D)) \mid M \in \boldsymbol{M}, D \in \boldsymbol{D}\}$. In the following we will be concerned with sets $\boldsymbol{M}$ that are equivalence classes $[]_\Phi$ for some set Φ of query properties.

Definition 5. *Let $\boldsymbol{M} \subseteq \mathcal{M}$, $\boldsymbol{D} \subseteq \mathcal{D}$. We say that B and $\boldsymbol{F}$ are orthogonal on $\boldsymbol{M} \times \boldsymbol{D}$, denoted $B \perp \boldsymbol{F}(\boldsymbol{M}, \boldsymbol{D})$, if $(B, \boldsymbol{F})(\boldsymbol{M}, \boldsymbol{D}) = B(\boldsymbol{M}) \times \boldsymbol{F}(\boldsymbol{M}, \boldsymbol{D})$.*

In other words, $B \perp \boldsymbol{F}(\boldsymbol{M}, \boldsymbol{D})$ means that on the set $\boldsymbol{M} \times \boldsymbol{D}$ a given value of $\boldsymbol{F}()$ does not constrain the possible values of $B()$, and vice-versa.

Proposition 1. *Let $\sigma(B, \boldsymbol{F})$ be a score function that is monotone in B. Let Φ be a query space. If there exist distinct, nonempty equivalence classes $[M]_\Phi, [M']_\Phi$, such that for all sufficiently large n, and all $D \in \mathcal{D}^{(n)}$:*

(i) $\boldsymbol{F}([M]_\Phi, D) = \boldsymbol{F}([M']_\Phi, D)$
(ii) $B \perp \boldsymbol{F}([M]_\Phi, D)$ *and* $B \perp \boldsymbol{F}([M']_\Phi, D)$,

then Φ is not $(\mathcal{D}, \sigma)$-identifiable.

Proof. Without loss of generality assume that

$$\sup B([M]_\Phi) \geq \sup B([M']_\Phi). \tag{5}$$

Then, for all sufficiently large n, all $D \in \mathcal{D}^{(n)}$, and all $\tilde{M}' \in [M']_\Phi$ there exists $\tilde{M} \in [M]_\Phi$ with $\sigma(\tilde{M}, D) \geq \sigma(\tilde{M}', D)$, and (3) does not hold for $[M']_\Phi$.

Example 6. (Non-identifiability of BN structure) Let $\mathcal{M}$ be the set of Bayesian networks over the variables A, B, C, D, and $\mathcal{D}^{(n)} = S^n$, where S is the sample space of joint observations of the variables. Let $\Phi = \{\phi_{X \to Y} \mid X, Y \in A, B, C, D\}$, where $\phi_{X \to Y}(M)$ is true iff the edge $X \to Y$ is included in M. Φ here is finite. The score functions (1) and (2) are based on the score features *Test()*, *Lhood()*, the data feature $g()$, as well as the bias features $R()$ (for PC) and $-h()$ (for PLS). We use Proposition 1 to show that Φ is not identifiable from data consisting of joint observations of A, B, C, D by any score function that is based on the score features *Test* and *Lhood*, together with any data features $F(D)$, and bias features such as $R()$ or $-h()$.

We consider the Φ-equivalence classes of M and M' given by Figure 1 (a) and (b). We first verify that (i) and (ii) hold for $[M]_\Phi$, $[M']_\Phi$, and $\boldsymbol{F} = (\mathit{Test}, \mathit{Lhood})$. We need not consider any pure data features $F(D)$ here: since D is fixed in (i) and (ii), any addition of pure data features $F(D)$ to $\boldsymbol{F}$ has no impact on the validity of (i) and (ii).

$[M]_\Phi$ and $[M']_\Phi$ contain exactly the Bayesian networks with the same structure as M and M', respectively. Since both structures represent the same conditional independencies, we have for all D that $\mathit{Test}([M]_\Phi, D) = \mathit{Test}([M']_\Phi, D) = \{0\}$ (the outcome of the independence tests on D are not compatible with the structures of M and M'), or $\mathit{Test}([M]_\Phi, D) = \mathit{Test}([M']_\Phi, D) = \{1\}$ (otherwise). Also, since exactly the same class of distributions can be represented by models

in $[M]_\Phi$ and in $[M']_\Phi$, one has $Lhood([M]_\Phi, D) = Lhood([M']_\Phi, D)$. This together shows (i) (note that in general it is not enough to show these identities separately for the value sets of each feature $F \in \boldsymbol{F}$; here it is sufficient because for the *Test* feature the value sets turned out to be just singletons).

Both the bias features $R()$ and $-h()$ have a unique value on the equivalence classes $[M]_\Phi$ and $[M']_\Phi$. This makes condition (ii) trivially satisfied for these and similar bias features.

Example 7. (Non-identifiability of PCTL) Let $\mathcal{M}^{nd}$ be the class of non-deterministic probabilistic finite automata, and let $\mathcal{D}^{(n)} = (\Sigma^*)^n$. Thus, even though the automata define a distribution over Σ^∞, we assume that data $D = (s_1, \ldots, s_n)$ consists of finite strings $s_i \in \Sigma^*$ only, where the s_i are obtained by sampling traces of the automaton up to some random length $l_i = \mid s_i \mid$. As mentioned in Section 1 we are interested in queries from the formal languages PLTL and PCTL. Full syntax and semantics definitions for these languages can be found in [2]. In the following we will only somewhat informally introduce specific properties they can represent. For this we will be needing formulas built using only the temporal operator "next (time point)", written $\bigcirc$. With PLTL one can specify probability bounds on the distribution over sequences defined by the automaton. For example, a PLTL-expressible property would be that the probability that the sequence starts with aaa is > 0.49, formally expressed by $P_{>0.49}(a \wedge \bigcirc a \wedge \bigcirc\bigcirc a)$. This property is satisfied by all three automata in Figure 2.

PCTL syntactically allows formulas in which probability quantifiers $P_{>\ldots}$ and temporal operators are nested. Semantically they represent model properties that refer to internal states of the automaton. One such property that we can formulate reads in formal PCTL syntax

$$\phi_1 \equiv P_{>0.4} \bigcirc P_{>0.9} \bigcirc a. \tag{6}$$

The meaning of this is that (from the initial state) there is a probability > 0.4 of reaching a state from which the probability then is > 0.9 of reaching by the next transition a state labeled with a. For the automata of Figure 2, ϕ_1 is false for M_a and M_b, and true for M_c. All automata of Figure 2 satisfy the following two formulas:

$$\phi_2 \equiv P_{=1}(a \wedge \bigcirc a) \tag{7}$$

$$\phi_3 \equiv P_{=0.5} \bigcirc \bigcirc a \tag{8}$$

Given that an automaton M satisfies ϕ_2 and ϕ_3, one has that $P_M(aaa\Sigma^\infty) = P_M(aab\Sigma^\infty) = 1/2$, and the distribution P_M then is fully specified by the two conditional distributions $P_M(\cdot \mid aaa)$ and $P_M(\cdot \mid aab)$. For the automata of Figure 2, both these conditionals are the uniform distribution on Σ^∞. One can show that also any other pair of conditional distributions that can be defined by a finite probabilistic automaton can be implemented both by automata for which ϕ_1 is true, and by automata for which ϕ_1 is false. For the first case, one can follow the basic structure of M_c, where one branches from the initial state

in the first step, followed by a deterministic transition in the second. For the second case, one builds on the structure of M_a, and constructs a model in which from every state reachable by the first transition there is a probability of 0.5 each for reaching an a, respectively b, state by the second transition.

Letting $\Phi = \{\phi_1, \phi_2, \phi_3\}$, the above considerations mean that the models in the two equivalence classes $[M_a]_\Phi$ and $[M_c]_\Phi$ can represent exactly the same distributions. This implies that for any vector of score features $\boldsymbol{F}$ that only contain features such as *Tests* and *Lhood*, which depend on M only through the distribution P_M, one has $\boldsymbol{F}([M_a]_\Phi, D) = \boldsymbol{F}([M_c]_\Phi, D)$ for all D. Thus Proposition 1 (i) is satisfied for $\boldsymbol{F}$ of this form, which includes Alergia-style learning (cf. (1)).

We here do not consider condition (ii) for a bias feature B. Alergia-style algorithms do not use a bias feature, and so there are no immediate canonical candidates. However, Alergia-style algorithms also learn deterministic automata only, whereas here we considered non-deterministic ones. Within the class $\mathcal{M}^{nd}$ a measure of the complexity of the model may be a reasonable bias to include in a learning process.

Proposition 1 may appear rather specific, and possibly narrow, due to the assumption that we are dealing with score functions that can be written as $\sigma(B, \boldsymbol{F})$. Furthermore, the structural conditions (i) and (ii) appear quite strong, and especially (ii) will probably not always hold, even in the case of non-identifiability. Thus, (i) and (ii) are simple sufficient, but certainly not necessary conditions for non-identifiability. On the other hand, Examples 6 and 7 show that Proposition 1 already does cover a certain range of different identifiability problems. The following example further indicates that the structural form of σ that we here assumed is in some sense natural and most general: already allowing a slight generalization in the form of σ, where one can have two different bias features B_1, B_2, again leads to trivial identifiability results similar to Example 4:

Example 8. Assume that both $\mathcal{M}$ and $\mathcal{D}$ are countably infinite. Let $f : \mathcal{M} \to \mathbb{N}$ and $g : \mathcal{D} \to 2\mathbb{N}$, where $2\mathbb{N}$ stands for the set of even natural numbers, such that both f and g are one-to-one and onto. Define the two model features $B_1(M) := f(M)$, $B_2(M) := -f(M)$, and consider $g(D)$ as a data feature. We can then define the score function

$$\sigma(M, D) := \begin{cases} B_1(M) & \text{if } B_1(M) \leq B_2(M) + g(D) \\ B_2(M) + g(D) & \text{otherwise} \end{cases} \tag{9}$$

It is straightforward to verify that σ is monotone in both B_1 and B_2. Also, Proposition 1 (i) and (ii) are trivially satisfied for $\boldsymbol{F} = g$ and $B = B_i$ $(i = 1, 2)$. For any given D, $\sigma(M, D)$ is maximized when $B_1(M) = B_2(M) + g(D)$, i.e., $f(M) = -f(M) + g(D)$, or $f(M) = g(D)/2$. By the assumptions on f and g there is a unique M that satisfies this condition. Thus, similarly as in Example 4 we can choose for a given M the constant sequence $D_n = D$, for the D with $g(D) = 2f(M)$.

We end this section by considering an example in relational learning. Probabilistic relational (or probabilistic logical) models define probability distributions over relational structures, i.e., over the interpretations I over finite domains

$C = \{c_1, \ldots, c_m\}$ of the relation symbols in a signature Σ. Most types of probabilistic relational models only define the conditional distributions $P(I \mid C)$ of interpretations for given domains. Only in some expressive representation languages such as BLOG [12] one can also specify distributions $P(C)$ over domains. For relational learning one can distinguish several types of stratified dataspaces:

$\mathcal{D}_1$: $\mathcal{D}^{(n)}$ consists of n independent samples $I_1, \ldots, I_n$ of interpretations over a fixed domain C.

$\mathcal{D}_2$: $\mathcal{D}^{(n)}$ consists of n independent samples $(I_1, D_1), \ldots, (I_n, D_n)$ of interpretations over different domains C_i. Example: molecular data, where each I_i represents a molecule described by attributes and bond-relations over a domain D_i of atoms.

$\mathcal{D}_3$: (cf. Example 3 (c)) $\mathcal{D}^{(n)}$ consists of one observation of an interpretation I over the domain $C_n = \{c_1, \ldots, c_n\}$, and $I \prec I'$ iff I is an interpretation over C_n, I' an interpretation over $C_{n'}$, $n < n'$, and I is the substructure induced by I' on C_n. Example: learning from a single database, such as the IMDB movie database, or DBLP bibliographic database. Increasing data here means that the database grows by adding more objects (movies, bibliographic entries...).

Probabilistic relational models learned from data for domains C_i may be used to perform inference for new domains C not represented in the data. This can be seen as the weakest form of model transfer, which in a much more ambitious form (also including a transformation of the relational signatures) becomes transfer learning in the sense of [11]. We now derive within our framework an impossibility result for relational model transfer. Again, the interest of the analysis does not so much lie in the concrete impossibility result we obtain, but in the demonstration that our general framework allows us to express in a rigorous manner what appears to be intuitively rather obvious.

Example 9. For concreteness' sake, let $\mathcal{M}$ be the class of Markov Logic Networks (MLNs) [14] for a signature of just a single attribute (unary relation) symbol $a(X)$. We assume that we learn from data of type $\mathcal{D}_3$, i.e., a dataset of size n consists of the domain C_n, and for each c_i the information whether $a(c_i)$ is true or false. We consider the query $\phi = P(a(c_1) \mid C = \{c_1\}) > 0.5$, i.e. we ask whether $P(a(c_1)) > 0.5$, for the single object c_1 in a domain of size one.

We now apply Proposition 1 to show that $\Phi = \{\phi\}$ is not $(\mathcal{D}_3, \sigma)$-identifiable by likelihood-based learning, i.e., for any σ that only uses the likelihood feature $\mathit{Lhood}(M, (I, C_n)) = P_M(I \mid C_n)$. Thus, we have to show (i) for $\boldsymbol{F} = \mathit{Lhood}$. Since Φ consists of a single Boolean query, there are only two equivalence classes $[]_\Phi$. Let $n \geq 2$, and $D = (I, C_n)$ a dataset of size n. Assume that in I $a(c_i)$ is true for $i = 1, \ldots, k$, and false for $i = k+1, \ldots, n$. Consider the two formulas

$$p(X_1) \wedge \ldots \wedge p(X_k) \wedge \neg p(X_{k+1}) \wedge \ldots \wedge \neg p(X_n) \wedge \bigwedge_{i \neq j} X_i \neq X_j \qquad (10)$$

$$p(X) \qquad (11)$$

Consider an MLN M consisting of formula (10) with a weight w. As $w \to \infty$, the MLN defines over the interpretations on C_n a distribution that is concentrated on the structures where $a()$ is true for exactly k objects, and false for $n-k$ objects. Since there are $\binom{n}{k}$ such structures, one obtains for these models likelihood values $P_M(I \mid C_n)$ converging to $1/\binom{n}{k}$. Since all possible MLNs for the given signature must assign equal probabilities to isomorphic structures, no higher likelihoods are obtainable by any other model. On the other hand, if $w \to -\infty$, then $P_M(I \mid C_n) \to 0$. For all settings of w one has $P_M(a(c_1) \mid C = \{c_1\}) = 0.5$, i.e., the query ϕ evaluates to false.

Now consider an MLN M' consisting of (10) with a weight w, and (11) with a weight $u = 1$. As w ranges in $]-\infty, \infty[$ one again has that the likelihood $P_{M'}(I \mid C_n)$ ranges in $]0, 1/\binom{n}{k}[$. Now, however, for all such M' $P_{M'}(a(c_1) \mid C = \{c_1\}) > 0.5$. Thus, one has that M and M' define two different Φ-equivalence classes, and $\boldsymbol{F}([M]_\Phi, D) = \boldsymbol{F}([M]_\Phi, D) =]0, 1/\binom{n}{k}[$.

For pure likelihood-based learning, i.e., in the absence of a bias feature $B(M)$, we thus obtain that ϕ is not identifiable. What happens if we were to add a bias feature that expresses a preference for syntactically simpler MLNs, as measured, for example, in terms of the number and length of formulas? MLNs containing formulas such as (10) would then obtain a low bias value $B(M)$. Simple MLNs with high $B(M)$ values, on the other hand, would most likely be unable to express the distribution that leads to the maximal likelihood value of $1/\binom{n}{k}$ for the data. Thus, condition (ii) would not be satisfied, and Proposition 1 does not establish non-identifiability for these scenarios.

Our examples show that Proposition 1 can be used to establish non-identifiability in some relevant cases, but it is far from being applicable in all cases. However, it appears that Proposition 1 is about as far as one can go without imposing some further restrictions on admissible score functions σ, or on the available data sequences $\uparrow D_n$.

4 Consistent Identifiability and PAC Learning

We now take a closer look at how Definition 3 can be strengthened by replacing the mere existence condition for a data sequence $\uparrow D_n$ with a condition only for the data-sequences that we are likely to see in practice. For this we now only consider spaces of probabilistic models that induce a distribution on the sample data.

Definition 6. *A model class $\mathcal{M}$ is* probabilistic *with associated stratified data space $\mathcal{D}$, if each $M \in \mathcal{M}$ defines*

- *a probability distribution $P_M^{(1)}$ on $\mathcal{D}^{(1)}$, and*
- *for each $n > 1$ a conditional distribution $P_M^{(n|n-1)}$ on $\mathcal{D}^{(n)}$ given $\mathcal{D}^{(n-1)}$, so that for $D \in \mathcal{D}^{(n-1)}$*

$$P_M^{(n|n-1)}(\{D' \mid D \prec D'\} \mid D) = 1.$$

The distributions $P_M^{(1)}, P_M^{(n|n-1)}$ jointly define probability distributions $P_M^{(n)}$ on $\mathcal{D}^{(n)}$ for all n.

We note that for continuous spaces $\mathcal{D}$ the above definition implicitly assumes some measurability conditions that we have not spelled out.

For probabilistic models we can now introduce *consistent identifiability*, which is an adaptation of the statistical concept of consistency for our type of learning scenario.

Definition 7. *Let $\mathcal{M}$ be a probabilistic model space with associated data space $\mathcal{D}$, σ a score function. Φ is* consistently σ-identifiable*, if for all $M \in \mathcal{M}$, all $\epsilon > 0$, and for all $\phi \in \Phi$ there exists $n_0 \geq 1$, such that for all $n \geq n_0$:*

$$P_M^{(n)}\{D_n \mid \mathcal{M}(\sigma, D_n) \subseteq [M]_{\equiv_\phi}\} \geq 1 - \epsilon \tag{12}$$

We can now formulate a positive identifiability result reported in [10]: for $\mathcal{M}^d$ the set of deterministic probabilistic finite automata, and Φ the class of PLTL queries: Φ is consistently σ-identifiable, where σ is an Alergia-style score function based on the model features *Tests* and *Lhood*. Comparing this result with Example 7 we find that PCTL is not identifiable in $\mathcal{M}^{nd}$ (under certain assumptions on the data and model features available for learning), whereas PLTL is identifiable in $\mathcal{M}^d$. This leaves as two open and important questions whether PCTL is identifiable in $\mathcal{M}^d$, or PLTL in $\mathcal{M}^{nd}$.

We close this section with some remarks on the relationship between condition (12) and PAC-learnability. For this assume that each $M \in \mathcal{M}$ defines a Boolean function f_M defined on a countable domain X, that $\mathcal{D}^{(n)}$ consists of sets of size n of pairs $(x, f_M(x))$ $(x \in X)$, and that $\Phi = X$. In addition, let M also define a probability distribution P_M over X, which then induces a distribution on $\mathcal{D}^{(n)}$ for each n. Finally, assume that here $\mathcal{M}(\sigma, D)$ is a singleton, which we denote $M(D)$. From consistent identifiability in the sense of Definition 7 we then obtain that for all $\epsilon, \delta > 0$ there exists n_0, so that for all $n > n_0$

$$P_M^{(n)}\{D_n \mid P_M\{x \mid f_{M(D)}(x) \neq f_M(x)\} \leq \delta\} \geq 1 - \epsilon \tag{13}$$

To obtain (13) from (12) we first observe that there exists a finite subset $X' \subseteq X$ with $P_M(X') > 1 - \delta$. Assume $|X'| = N$, and set ϵ to ϵ/N in (12). Then, for sufficiently large n_0, (12) holds simultaneously for all $\phi = x \in X'$, i.e., for all $n \geq n_0$

$$P_M^{(n)}\{D_n \mid M(D) \in [M]_{X'}\} \geq 1 - \epsilon$$

Since $M(D) \in [M]_{X'}$ implies $P_M\{x \mid f_{M(D)}(x) \neq f_M(x)\} \leq \delta$, we obtain (13).

Property (13) is structurally similar to PAC-learnability [18, 1]. One superficial difference between (13) and PAC-learnability is that the latter does not assume that there is an association between the (true) model (or hypothesis) and the distribution over X. Thus, where our definition says "for all $M \ldots$", PAC is expressed in terms of "$\ldots$ for all h (hypotheses) and all μ (distributions on X) $\ldots$". This, however, is no real difference, since if our model space contains

models for all possible combinations of functions f_M and distributions P_M, then the quantification over all M is the same as a quantification over all functions and all distributions. A second difference is much more significant: PAC is a uniform condition where the ϵ, δ-bounds are independent of h and μ, i.e., it is defined by the quantifier string $\forall \epsilon, \delta \exists n_0 \forall h, \mu \forall n > n_0 \ldots$, whereas our condition (13) is $\forall M, \epsilon, \delta \exists n_0 \forall n > n_0 \ldots$. Requiring in our Definition 7 a threshold n_0 that is uniform for M and ϕ would be too strong for most intended applications, since models M and queries ϕ may differ widely with respect to their complexity, and so it will be unrealistic to ask for uniform bounds on the necessary sample size for their identification.

5 Conclusion

We have developed a formal framework for analyzing learnability, or identifiability questions in learning scenarios where there may only be a loose association between model, data, and query space. The main contribution of this paper is to provide the conceptual tools for a rigorous and uniform analysis of a wide spectrum of such identifiability problems.

A key element of the proposed approach is to conceptualize learning as maximization of a score function with a dependence on a single distinguished model bias feature. Based on such score functions, we can formulate a first general sufficient condition for non-identifiability. This result is still somewhat limited in two ways: first, while easy to prove, the result is not necessarily easy to apply, since verifying conditions (i) and (ii) may require some non-trivial analysis in different applications. Second, Proposition 1 is a rather strong sufficient condition that may not actually be satisfied in many cases of non-identifiability. However, as Examples 4 and 8 indicate, it may prove difficult to obtain stronger results at the same high level of generality, and without restrictions to specific types of model spaces or learning approaches. A particularly relevant more specialized setting is that of consistent identifiability for probabilistic models. Future work will be focussed on obtaining more powerful analysis tools than Proposition 1 for this setting.

References

1. Angluin, D.: Queries and concept learning. Machine Learning 2, 319 – 342 (1988)
2. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
3. Carrasco, R., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: Grammatical Inference and Applications, Lecture Notes in Computer Science, vol. 862, pp. 139–152. Springer Berlin / Heidelberg (1994)
4. Carrasco, R.C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. ITA pp. 1–20 (1999)
5. De Raedt, L., Frasconi, P., Kersting, K., Muggleton, S. (eds.): Probabilistic Inductive Logic Programming, Lecture Notes in Artificial Intelligence, vol. 4911. Springer (2008)

6. Glymour, C., Spirtes, P., Richardson, T.: On the possibility of inferring causation from association without background knowledge. In: Glymour, C., Cooper, G.F. (eds.) Computation, Causation & Discovery, chap. 9, pp. 323 – 331. AAAI Press / MIT Press (1999)
7. Goodman, N.D., Mansinghka, V.K., Roy, D., Bonawitz, K., Tenenbaum, J.B.: Church: a language for generative models. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI-08) (2008)
8. de la Higuera, C.: Grammatical Inference: Learning Automata and Grammars. Cambridge University Press (2010)
9. Korb, K., Nicholson, A.: The causal interpretation of Bayesian networks. In: Holmes, D., Jain, L. (eds.) Innovations in Bayesian Networks, Studies in Computational Intelligence, vol. 156, pp. 83–116. Springer Berlin / Heidelberg (2008)
10. Mao, H., Chen, Y., Jaeger, M., Nielsen, T.D., Larsen, K.G., Nielsen, B.: Learning probabilistic automata for model checking. In: Proceedings of the 8th International Conference on Quantitative Evaluation of SysTems (QEST) (2011)
11. Mihalkova, L., Huynh, T., Mooney, R.J.: Mapping and revising markov logic networks for transfer learning. In: Proc. of AAAI-07 (2007)
12. Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D., Kolobov, A.: Blog: Probabilistic logic with unknown objects. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05). pp. 1352–1359 (2005)
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference. The Morgan Kaufmann series in representation and reasoning, Morgan Kaufmann, San Mateo, CA, rev. 2nd pr. edn. (1988)
14. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107 – 136 (2006)
15. Robins, J.M., Wasserman, L.: On the impossibility of inferring causation from association without background knowledge. In: Glymour, C., Cooper, G.F. (eds.) Computation, Causation & Discovery, chap. 8, pp. 305–321. AAAI Press / MIT Press (1999)
16. Sen, K., Viswanathan, M., Agha, G.: Learning continuous time Markov chains from sample executions. In: Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST). pp. 146–155 (2004)
17. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search. Springer Verlag (1993)
18. Valiant, L.G.: A theory of the learnable. Communications of the ACM 27(11), 1134 – 1142 (1984)