Using Learning Algorithms to Create, Exploit and Maintain Knowledge Bases: Principles of Constructivist Machine Learning

Thomas Schmid
Universität Leipzig
Machine Learning Group
Augustusplatz 10, D-04109 Leipzig, Germany
schmid@informatik.uni-leipzig.de

Abstract

Recently, interest has grown in connecting modern machine learning approaches with traditional expert systems. This can mean, e.g, to identify patterns with neural networks and integrate them with knowledge graphs. While such combined systems offer a variety of advantages, few domain-independent approaches are known to make a hybrid artificial intelligence applicable without human interaction. To this end, we present the implementation of a constructivist machine learning framework (conML). This novel paradigm uses machine learning to manage a knowledge base and thereby allows for both raw data-based and symbolic information processing on the same internal knowledge representation. Based on axioms for a constructivist machine learning, we describe which operations are required to create, exploit and maintain a knowledge base and how these operations may be implemented with machine learning techniques. The major practical obstacle in this approach is to implement an automated deconstruction process that avoids ambiguity, handles continuous learning and allows knowledge abstraction. As we demonstrate, however, these obstacles can be overcome and constructivist machine learning can be put into practice.

Combining machine learning and knowledge engineering is currently considered a potential game changing advancement in artificial intelligence. Neural networks and other machine learning techniques have proven strength in adapting to highly complex patterns and relationships, but are unable to represent existing knowledge explicitly and in an abstract fashion as expert systems can. Expert systems, on the other hand, operate on human-understandable knowledge representations but are highly domain-specific and, moreover, unable to process real-world data directly as machine learning can. Therefore, it is expected that joining both fields will produce a hybrid artificial intelligence that is "explainable, compliant and grounded in domain knowledge" (Martin et al. 2019). Such systems may, e.g., be able to identify patterns with neural networks and integrate them with knowledge graphs (Subasic, Yin, and Lin 2019).

In fact, the idea of a hybrid artificial intelligence has been discussed for more than 30 years (Gallant 1988; Hendler 1989; Skeirik 1990; Levey 1991; Morik et al. 1993). So far, however, most research in this field focuses on specific knowledge or application domains like medical diagnosis (Hudson, Cohen, and Anderson 1991; Karabatak and Ince 2009; Herrmann 1995). This is to a large extent due to the fact that knowledge bases are typically created manually, which is a highly time-consuming task that requires detailled knowledge of the domain (Kidd 2012). No less time-consuming are exploitation and maintenance of knowledge bases, which are typical follow-up phases within the life cycle of a knowledge base. While some progress has been made in employing algorithms for these tasks, several major challenges for an automated management of knowledge bases are still considered unresolved (Martinez-Gil 2015).

Considering recent performance advancements in machine learning, manually managed knowledge bases obviously constitute a serious bottleneck in creating efficient hybrid systems. For truely automated systems, however, an implementable semantic interface between inductive machine learning and deductive expert systems is required. To this end, we have introduced a constructivist machine learning paradigm (Schmid 2019) based on the concept of learnable models and their storage in a knowledge base. While machine learning is currently dominated by neuro-inspired approaches, constructivist theories root in educational research (Fox 2001) and, so far, few actual implementations have been proposed for a constructivist machine learning (Drescher 1989; Quartz 1993). Central challenge for putting this into practice is the implementation of an automated deconstruction process, which to the best of our knowledge has only once been addressed successfully (Schmid 2018).

Based on this paradigm, we designed a prototype for a constructivist machine learning that employs a meta data-based knowledge base. Here, we present the underlying operationalizations and concepts required to put constructivist machine learning into practice. The rest of the paper is organized as follows: In section I, we lay out guidelines for automated knowledge base management. In section II, we define Stachowiak-like models as building blocks for knowledge representations. In section III, we introduce principles for constructivist machine learning processes. In section IV, we summarize our approach and point out future goals.

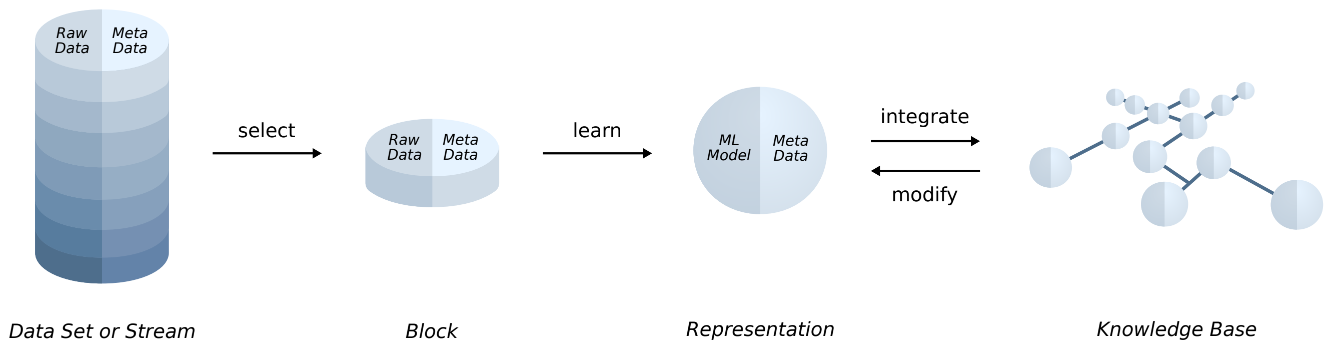

Figure 1: Transformation of real-world data into an abstract knowledge base. Using a constructivist machine learning approach, real-world data is processed block-wise by learning algorithms with the aim of identifying an optimal representation for a given block. Each representation is then integrated into an existing knowledge base consisting of previously identified representations.

I. Knowledge Management

In the context of knowledge engineering, a knowledge representation is typically a mathematical formalization like a logic, rule, frame or semantic net related to real-world aspects (Davis, Shrobe, and Szolovits 1993). We have recently argued that any such formalization should be regarded as a model in the sense of Stachowiak's General Model Theory (Schmid 2019). This implies that a formalization is not only a representation and an abstraction, but also limited to certain temporal constraints, certain subjects and a certain purpose (Stachowiak 1973). Here, we represent and use these three-dimensional limitations explicitly by employing meta data acquired together with raw data (Fig. 1).

Hierarchical Knowledge. By a basic definition, a knowledge base can be simply viewed as "a set of formulas" (Lifschitz, Morgenstern, and Plaisted 2008). In the present work, we use an extended definition and regard a set of meta data-enriched models as a knowledge base (Schmid 2019). We further assume a meta data-based hierarchical ordering of this set, as human knowledge is from an educational perspective assumed to be organized in distinct levels (Bloom 1956). Findings from neurobiology also indicate a hierarchical organization for cognitive brain areas (Markov and Kennedy 2013). Using machine learning-based models as knowledge representations, we reflect a hierarchical ordering by using the output of other such models as input.

Knowledge Domains. A revision of Bloom's taxonomy suggests that apart from levels also domains of human cognition should be distinguished (Anderson and Krathwohl 2001). Conceptual knowledge, e.g., may be described as knowledge about classifications, categories and structures. Procedural knowledge, in contrast, may be described as knowledge about subject-specific abilities, algorithms or selection criteria. We suggest to use individual knowledge bases for individual knowledge domains, i.e., factual, conceptual, procedural and metacognitive models. In the present work we wil focus on the conceptual knowledge domain and the mechanisms involved with this type of knowledge.

Automated Knowledge Base Management. Managing knowledge bases may be described by typical life cycle phases. Following Martinez-Gil (2015), a creation phase is characterized by acquisition, representation, storage and manipulation of knowledge, while an exploitation phase focusses on knowledge reasoning, retrieval and sharing; the maintenance phase is concerned with integration, validation and meta-modeling of knowledge. Issues raising within these phases have been recognized and discussed (Richardson and Domingos 2003; Guisado-Gámez, Dominguez-Sal, and Larriba-Pey 2013; Falkner and Haselböck 2013). Most work on operating knowledge bases use a semi-automated approach, leaving much space for more effective and efficient automated management strategies (Martinez-Gil 2015). Important issues to be addressed include, on one hand, automatic generation of large knowledge bases as well as automatic selection, combination and/or tuning of maintenance strategies. On the other hand, efficiency and explainablity of knowledge exploiting should be improved, too.

Employment of Machine Learning. Here, machine learning techniques will be used for automatic generation of knowledge bases as well as for automatic maintenance. In creation phases, machine learning algorithms are employed to identify and/or manipulate optimal knowledge representations. In maintenance phases, machine learning algorithms are used for validating such knowledge representations and for supporting their integration into the knowledge base. To this end, a major objective of maintenance is to keep the knowledge base ambiguity-free. For knowledge exploitation, machine learning-based models of such a knowledge base may be applied on new input data. This is due to the design aspect that each model is represented by a supervised learning algorithm, i.e. a classifier or regressor. Consequently, the underlying classifier or regressor may be used on new data after training. Matching and mismatching new data to a model can be achieved by the respective meta data. In particular, application of the knowledge base can be rejected if no knowledge is available for a given input.

II. Models as Knowledge Representations

In the following, models will be used to represent acquired knowledge. A model here is understood to be a pragmatic model in the sense of Stachowiak's General Model Theory (Stachowiak 1973). This includes mathematical functions as well as their representation or approximation by machine learning techniques. More importantly, however, Stachowiak-like models feature meta data about the validity of the model regarding subject, purpose and time.

The author, user or *subject* σ, of a model may in natural sciences be a sensor or a measuring device, and in observational studies or content analyses typically a human evaluator. The set of all model subjects σ_i for which a given model $\mathcal{M}$ is valid, is called $\Sigma_{\mathcal{M}}$ and defined as the subset of the (infinite) set Σ of all possible subjects:

$$\Sigma_{\mathcal{M}} \subset \Sigma \tag{1}$$

The target parameter of a model is referred to as *purpose* ζ. The set of all purposes ζ_i, for which a given model $\mathcal{M}$ is valid, is called $Z_{\mathcal{M}}$ and defined as subset of the (infinite) set Z of all possible model purposes:

$$Z_{\mathcal{M}} \subset Z \tag{2}$$

The temporal validity of a given model $\mathcal{M}$ is in general represented by a *time span* $T_{\mathcal{M}}$ or a minimum limit τ_{min} and a maximum limit τ_{max}, respectively:

$$T_{\mathcal{M}} = [\tau_{min}, \tau_{max}] \tag{3}$$

In contrast to Stachowiak's model concept, we limit our approach to two types of models: to vector models on the one hand and to algorithmically generated machine models on the other. For both, a distinction is made between models with and without explicitly defined pragmatic properties.

a) Vector Models

In supervised machine learning, a training vector consists of an m-dimensional input vector $I = (i_0, ..., i_{m-1})$ and an n-dimensional output vector $O = (o_0, ..., o_{n-1})$. Moreover, a mapping between I and O is implicitly assumed. Such vectors are referred to as *(complete) vector model* $\mathcal{V}$:

$$\begin{aligned} \mathcal{V} &= (I, O) && (4)\\ &= (i_0, ..., i_{m-1}, o_0, ..., o_{n-1}) && (5) \end{aligned}$$

If a given I is assigned an empty output vector, $O = \emptyset$, the corresponding $\mathcal{V} = (I, \emptyset)$ is termed an *incomplete vector model*. Typical incomplete vector models are training vectors used for an unsupervised machine learning process.

If the pragmatic properties T, Σ and Z are explicitly defined for a complete vector model $\mathcal{V}$, the resulting representation is called a *pragmatically defined vector model* $\mathcal{V}$*:

$$\mathcal{V}^* = (\mathcal{V}, T_{\mathcal{V}}, \Sigma_{\mathcal{V}}, Z_{\mathcal{V}}) \tag{6}$$

Note that the time span $T_{\mathcal{V}}$, within which $\mathcal{V}$ is valid, is defined by the time of data collection. In the following, we assume that error tolerances during data collection are negligible and that minimum and maximum borders are identical:

$$T_{\mathcal{V}} = \tau_{min} = \tau_{max} \tag{7}$$

b) Machine Models

If a finite set of j complete vector models is approximated by a machine learning algorithm, the resulting approximation is referred to as a *machine model* $\mathcal{M}$:

$$\mathcal{M} \sim \{\mathcal{V}_0, ..., \mathcal{V}_{j-1}\} \tag{8}$$

A machine model $\mathcal{M}$ with given $T_{\mathcal{M}}$, $\Sigma_{\mathcal{M}}$ and $Z_{\mathcal{M}}$ is called *pragmatically defined machine model* $\mathcal{M}^*$:

$$\mathcal{M}^* = (\mathcal{M}, T_{\mathcal{M}}, \Sigma_{\mathcal{M}}, Z_{\mathcal{M}}) \tag{9}$$

The temporal validity $T_{\mathcal{M}}$ of a machine model $\mathcal{M}$ can only be assumed to be hypothetical and defined by means of hypothetical interval limits. These interval limits are derived from the underlying n vector models $\mathcal{V}^*$ (Schmid 2018), which were used to train the machine learning algorithm:

$$T_{\mathcal{M}} = \Big[min(T_{\mathcal{V}^*_0}, ..., T_{\mathcal{V}^*_{n-1}}), max(T_{\mathcal{V}^*_0}, ..., T_{\mathcal{V}^*_{n-1}})\Big] \tag{10}$$

$\Sigma_{\mathcal{M}}$ defines the machine learning algorithms involved in creating and applying $\mathcal{M}$. In order to allow for automated model creation, we use generic descriptors. For a standard machine model, $\Sigma_{\mathcal{M}}$ will be a set containing only one element. If $|\Sigma_{\mathcal{M}^*}| > 1$ holds true for a given $\mathcal{M}^*$, i.e., if the model is valid for more than one machine learning algorithm, $\mathcal{M}^*$ is called an *intersubjective machine model*.

$Z_{\mathcal{M}}$ defines the target parameters of a machine model $\mathcal{M}$. In most cases, $Z_{\mathcal{M}}$ will be a set containing only one element. In order to allow for automated model creation, we use generic descriptors that are a combination of the corresponding knowledge domain, knowledge level and type of task (e.g. binary classification). If $\mathcal{M}^*$ is abstracted from machine models $\mathcal{M}_0, ..., \mathcal{M}_{n-1}$, a higher level of knowledge is defined for $\mathcal{M}^*$ than for $\mathcal{M}_0, ..., \mathcal{M}_{n-1}$.

c) Model Relationships

The pragmatic features T, Σ, Z of Stachowiak-like models may be employed to match and discriminate models automatically. With vector models, e.g., this allows to identify sets of pragmatically related vector models and define appropriate learning strategies for each relationship.

The degree of relationship between two given Stachowiak-like models $\mathcal{M}_a$ and $\mathcal{M}_b$ is termed

1. complete ($T\Sigma Z$),
   if $T_{\mathcal{M}_a} = T_{\mathcal{M}_b}$, $\Sigma_{\mathcal{M}_a} = \Sigma_{\mathcal{M}_b}$, $Z_{\mathcal{M}_a} = Z_{\mathcal{M}_b}$.
2. subjective-intentional (ΣZ),
   if $T_{\mathcal{M}_a} \neq T_{\mathcal{M}_b}$, $\Sigma_{\mathcal{M}_a} = \Sigma_{\mathcal{M}_b}$, $Z_{\mathcal{M}_a} = Z_{\mathcal{M}_b}$;
3. temporal-intentional (TZ),
   if $T_{\mathcal{M}_a} = T_{\mathcal{M}_b}$, $\Sigma_{\mathcal{M}_a} \neq \Sigma_{\mathcal{M}_b}$, $Z_{\mathcal{M}_a} = Z_{\mathcal{M}_b}$;
4. temporal-subjective ($T\Sigma$),
   if $T_{\mathcal{M}_a} = T_{\mathcal{M}_b}$, $\Sigma_{\mathcal{M}_a} = \Sigma_{\mathcal{M}_b}$, $Z_{\mathcal{M}_a} \neq Z_{\mathcal{M}_b}$;

Such matching and discriminating is also a prerequisite for automating a deconstruction process for machine models. Depending on the underlying pragmatic relationship, procedures for a ΣT, TZ, $T\Sigma$ or complete deconstruction can be defined (section III).

When applying existing machine models, pragmatic features also indicate applicability for a given task or input.

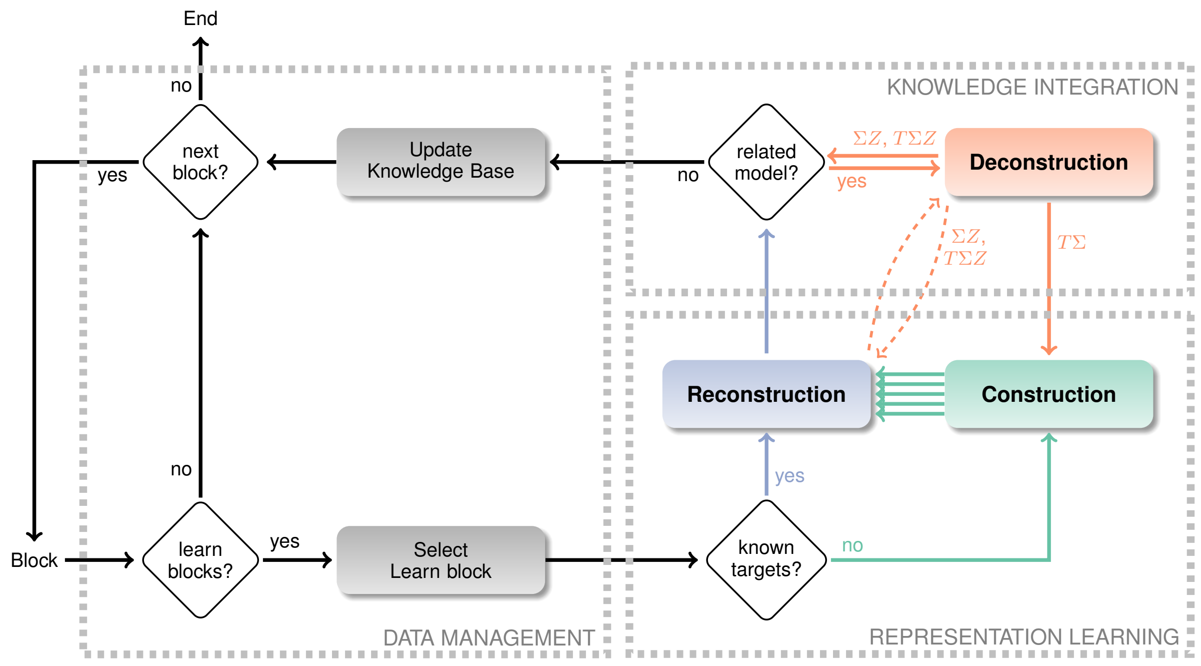

Figure 2: Principle of Constructivist Machine Learning. A given block of data and meta data is used to select and learn an optimal representation. This representation is then integrated into a knowledge base (for simplicity not depicted here) and/or modified accordingly.

III. Constructivist Machine Learning

According to modern educational concepts, human learning takes place through construction, reconstruction or deconstruction of models. Following this paradigm, we develop concepts to implement such learning processes. To put construction, reconstruction and deconstruction into practice, we require a corresponding knowledge base consisting of Stachowiak-like models (section II) and employ a data management process in order to organize for efficient learning.

Data Management. As Fig. 2 depicts, starting point for a constructivist machine learning procedure is an arbitrary set of pragmatically defined vector models (called block). From these samples, subsets of pragmatically related vector models are identified and re-grouped into learn blocks. Dependending on the sample relationship, ΣZ-, $T\Sigma$-, TZ- or completely related learn blocks may be found. The size of these learn blocks determines the following learning. Not all forms of relationship, however, are equally suitable for a model construction. Especially constructions based on completely and TZ-related learn blocks offer little added value. Learn blocks of completely related vector models that are not redundant but divergent even represent a serious source of error. Learn blocks of TZ-related models basically allow the generation of new models, which then, however, does not represent a construction process but an intersubjective reconstruction process. Therefore, for constructions learn blocks of ΣZ-related vector models are preferred.

If at least one learn block exceeds a user-defined minimum number of samples, the largest learn block is selected to undergo one or more learning processes. All other learn blocks are discarded. After the learning processes for this learn block have terminated, the knowledge base is updated according to the results of the learning processes. This may imply storing a newly reconstructed model as well as modifying or deleting existing models from the knowledge base. As long as further blocks exist, this sequence of selecting and processing data is repeated.

Representation Learning. Various combinations of learning processes are possible for a given learn block. In the most simple case, e.g., if the knowledge base contains no models yet and target values are defined for the learn block, only a reconstruction is carried out and the resulting machine model is stored in the knowledge base. In an educational context, reconstruction implies in general application, repetition or imitation, in particular the search for order, patterns or models (Reich 2004, p. 145). Similarly, the reconstruction of a machine model is here understood as supervised learning from given examples. In contrast to classical supervised learning, however, competing machine models are generated and evaluated with regard to their intersubjective validity.

If no target values are defined for the learn block, such targets are produced in a construction process, before the resulting model candidates enter the reconstruction process. In an educational context, construction is in general associated with creativity, innovation and production, and in particular with the search for new variations, combinations or transfers (Reich 2004, p. 145). For machine models this is interpreted as an unsupervised learning that identifies or defines alternative n-dimensional outputs to a set of incomplete vector models. Thereby, competing model candidates are created

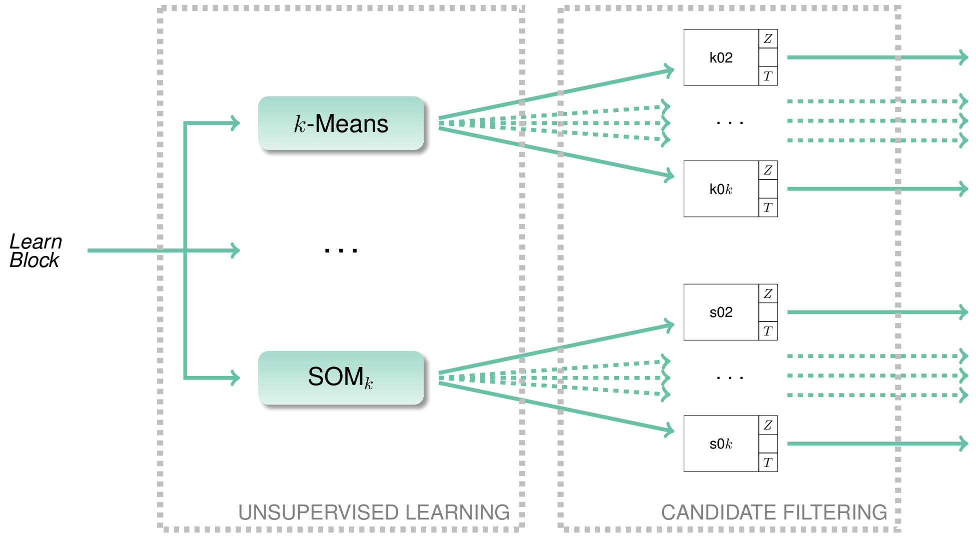

Figure 3: Construction process for conceptual knowledge. A given learn block is analyzed by k-Means and self-organizing maps (SOM) for cluster numbers 2,...,k. The resulting clusterings ($s02$,...,$s0k$,$k02$,...,$k0k$) are filtered on a user-defined basis.

that are evaluated in a following reconstruction process. Rationale behind this is that it is a priori unclear which of the models constructed from a learn block can be reconstructed with best accuracy and intersubjectivity.

Knowledge Integration. After successful reconstruction, mechanisms are needed to manage integration into the knowledge base. In particular, a deconstruction process is carried out to avoid redundancies and contradictions, if pragmatically related models exist in the knowledge base. In an educational context, deconstruction in general means the investigation of an already existing construct for incompleteness, for the unforeseen and the unconscious, and in particular the search for possible omissions, simplifications, additions and criticism (Reich 2004, p. 145). In constructivist machine learning, deconstruction is in particular associated with automated re-training of models and creating abstracted models. Deconstruction may result in modifying or discarding models of the knowledge base.

a) Construction

The aim of the construction process is to provide alternative interpretations, or model candidates, with alternative model purposes for a given learn block. In particular, more than one model candidate is created for the same data during construction and sent to a following reconstruction process. The key components of the construction process are unsupervised learning and candidate filtering (Fig. 3).

Unsupervised Learning. Depending on the knowledge domain under consideration, different types of unsupervised algorithms are employed. For conceptual knowledge in the sense of Bloom's taxonomy (section I), or knowledge about classifications, categories and structures, respectively, clustering algorithms are employed. The purpose of clustering in this case is to identify distinguishable categories within a learn block. In order to create diverse model candidates, it is desirable to identify as many different machine models as possible in as many different ways as possible. In a basic conceptual construction setting, the well-known k-Means clustering as well as the neuro-inspired self-organizing map (Baçao, Lobo, and Painho 2005) are used as alternative approaches. In a basic procedural construction setting, feature clustering (Chavent et al. 2012) and autoencoders (Hinton and Salakhutdinov 2006) may be employed. If the algorithm requires to define in advance the number k of clusters to be identified, all possible clusterings between 2 and k are being tested. This maximum number of clusters is called maximum categorical complexity κ_k in the following. With each clustering method $\kappa_k - 1$ machine models with $k = \{2, ..., \kappa_k\}$ clusters or categories are generated.

Candidate Filtering. Prerequisite for many clustering methods is the prior definition of a number of clusters to be determined. Usually, several runs with different cluster numbers are carried out with the same procedure and the clusterings obtained are evaluated with an external procedure (Jain 2010). Here, optimal clustering is determined by reconstructing model candidates. Before entering the reconstruction process, however, clusterings are filtered by user-defined settings for minimal cluster size and minimal clustering error (e.g. minimal intra cluster error, if applicable).

b) Reconstruction

The aim of the reconstruction process is to validate model candidates, assign model subjects and guarantee intersubjectivity. The key components of this process are preprocessing, supervised learning and intersubjectivity evaluation (Fig. 4). If more than one model candidate enters the reconstruction process from the construction process, only one model is selected as optimal learn block representation and transferred into the subsequent deconstruction process. All other reconstructed models are discarded.

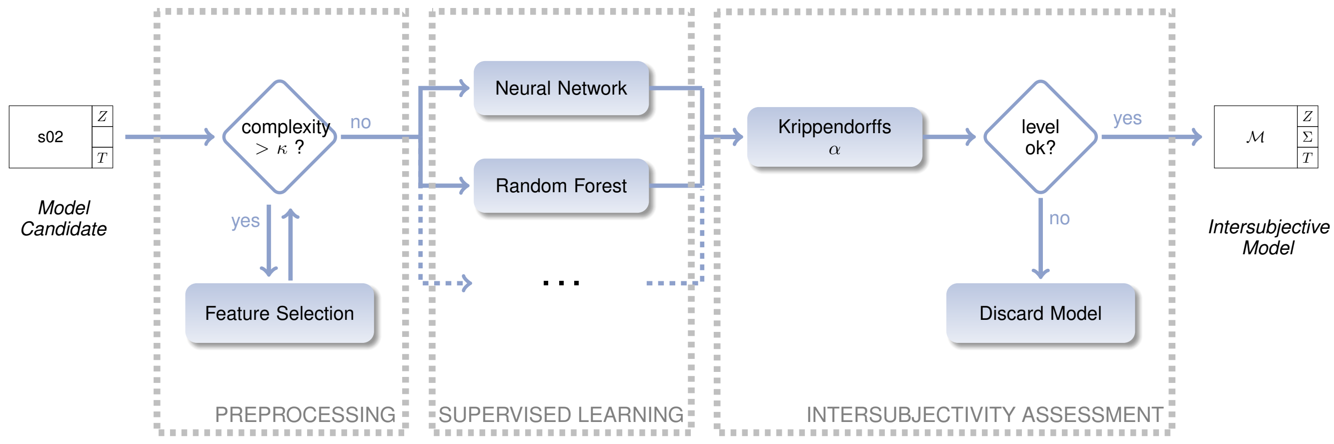

Figure 4: Reconstruction process. After initial preprocessing, supervised learning algorithms are applied on a given model candidate. The predictions of these algorithms for test data is then used to evalute intersubjetivity of the model candidate.

Preprocessing. For a given learn block or set of vector models, respectively, first the number of input variables is assessed. If this number is greater than a user-defined maximum model complexity κ, an algorithmic feature selection is carried out in order to reduce the complexity of the model to $\leq \kappa$. In principle, filter as well as wrapper and embedded methods could be used for this. The most convenient decision criterion for this is the amount of data to be processed or the selection processes to be performed. For large amounts of data, filter methods are preferable due to their efficiency and computation effort, but they do not always provide optimal feature subsets. Embedded feature selection methods, on the other hand, promise a particularly careful selection of features, but require considerably more computational effort than filters for large amounts of data.

For this reason, a hybrid approach is used here. As a basic principle, if an input data set consists of a smaller set of low-dimensional vector models, an embedded method is applied. Conversely, a filter is used if an input data set consists of a particularly large set of particularly high-dimensional vector models. Whether an embedded method is used or not is decided by means of user-defined auxiliary parameters for the maximum allowed number of input dimensions and the maximum allowed number of vector models. By default, Correlation-based Feature Selection (CFS) is used as filter, and a Random Forest as embedded method.

Supervised Learning. Preprocessing is followed by application of at least two alternative supervised learning algorithms that are carried out in parallel. It is important to note that these algorithms do act independently, but aim to achieve agreement. In order to enable broad application, these methods should be able to solve both classification and regression tasks. In order to facilitate automated configuration, they should also be non-parametric, i.e. they should not require a priori assumptions about the density distribution of the data. Furthermore, a fundamental diversity of the procedures is desirable in the sense of different procedural approaches. For this purpose, a biologically inspired and a statistically motivated learning procedure are applied in parallel. Considering these criteria as well as the availability of suitable implementations, the methods used for the reconstruction process are multi-layer perceptrons and random forests. In addition, further methods like support vector machines could be used to increase to diversity of methods.

Intersubjectivity. Each supervised learning yields individual target values for each input vector. Consequently, the question arises to what extent these competing methods agree. Analogously to empirical studies, this is quantified and evaluated with the interrater reliability coefficient Krippendorf's α. This coefficient can be calculated for both nominal and metric scales and can therefore be used for both classification and regression (Krippendorff 1970). In contrast to other reliability coefficients, it can also be applied to any number of raters. An α value of 1 indicates optimal reliability, while a value less than or equal to 0 implies that there is no match between scores. Krippendorf's α was repeatedly proposed as a standard measure for quantifying interrater reliability (Krippendorff 2004; Hayes and Krippendorff 2007). Those reconstructed models that have been trained successfully, but whose α value does not exceed a user-defined threshold value, are discarded. If none of the reconstructed models exceeds this α threshold, the current total reconstruction process is aborted.

Model Selection. If more than one model candidate passes the reconstruction process, it must be decided which of these competing models will be integrated into the knowledge base. In order to identify the learn block representation that is least dependent on specific methods, these models are ranked in descending order using Krippendorff's α. Since the maximum α value implies the maximum degree of intersubjectivity, the model with the maximum α value can be interpreted as the clearest model in the sense of Heinrich Hertz (Hertz 1894, p. 2f). If two or more models within the sequence have an identical α value, the model with the smallest original image space is selected from this new subset. This can be interpreted as the choice of the simplest model.

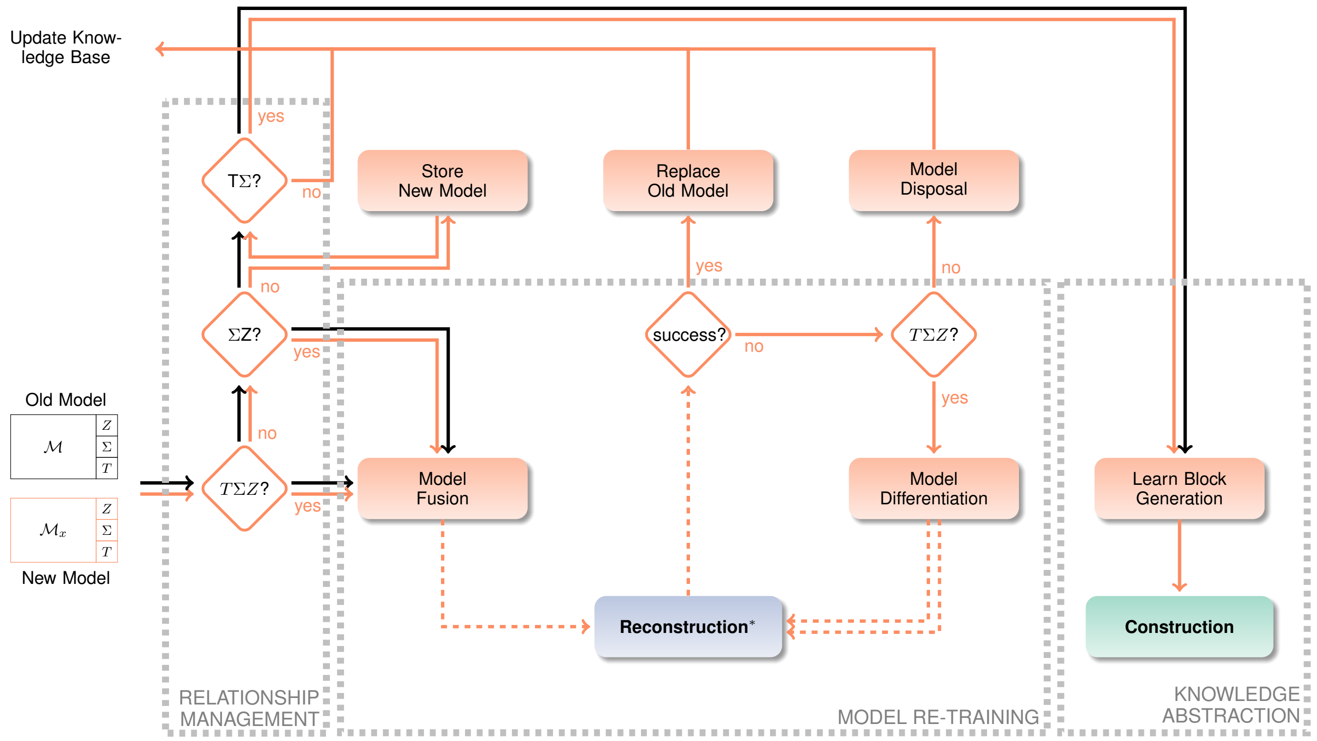

Figure 5: Deconstruction process. For deconstruction, two pragmatically related models are considered pairwise. They can undergo a re-training procedure, which makes use of the reconstruction process, or be used to abstract knowledge by undergoing a construction process.

c) Deconstruction

The aim of the deconstruction process is to combine new and old knowledge in a way that avoids ambiguity and allows to abstract knowledge automatically. The key components of this process are relationship management, model re-training and knowledge abstraction (Fig. 5). Prerequisite is that an existing model has been identified from the corresponding knowledge base that exhibits a pragmatic relationship to a newly reconstructed model. In the event that two or more related models are identified for a newly reconstructed model, these can either be deconstructed consecutively or the deconstruction process is aborted as soon as a complete, ΣZ, TZ or $T\Sigma$ deconstruction was successful.

Relationship Management. What procedures are carried out during deconstruction depends on the type of relationship (section II) between the two models entering the deconstruction process together. The decision on what measures to undertake is the initial task of the deconstruction process. In case of completely and ΣZ-related models, this relationship is assessed by model re-training, which makes use of the reconstruction process. In case of $T\Sigma$-related models, deconstruction is carried out in terms of a knowledge abstraction procedure, which makes use of the construction process. The case of TZ-related models would reflect that models with the same purpose and same temporal validity but differing subjects have been identified, which under a fixed intersubjective reconstruction scheme is not possible; therefore, this relationship is not explicitly handled in the following. If a newly reconstructed model shows a complete relationship to an existing model from the knowledge base, this may introduce error and contradiction into the knowledge base. Therefore, this relationship is handled with high priority.

Model Re-training. With ΣZ-related models, the aim of deconstruction is to extend or replace the existing model from the knowledge base. In particular, it is assessed whether the temporal validity of the existing model can be expanded according to the temporal validity of the new model. Both models are fused into a new model that is re-trained via the reconstruction process. If successful, the old model is replaced by the fused model, otherwise the new model and the fused model are discarded. For completely related models, re-training is initiated by model fusion as well as by model differentiation. Model differentiation means that it is tested whether the fused model may be split in two submodels of more limited temporal validity.

In contrast to ΣZ relationships, deconstruction of completely related models can not only extend but also falsify the validity of these models. If the model fusion is falsified in this case, the differentiation of the fused model is executed or, if necessary, one of the contradicting models is discarded. The disposal of models is carried out according to a user-defined regime, which makes a distinction between a conservative ($\mathcal{M}_{old}$ retained, $\mathcal{M}_{new}$ discarded) and an integrative ($\mathcal{M}_{old}$ discarded, $\mathcal{M}_{new}$ added to knowledge base) regime. Alternatively, if $\mathcal{M}_{new}$ is based on a larger set of vector models than $\mathcal{M}_{old}$, $\mathcal{M}_{new}$ is added to the knowledge

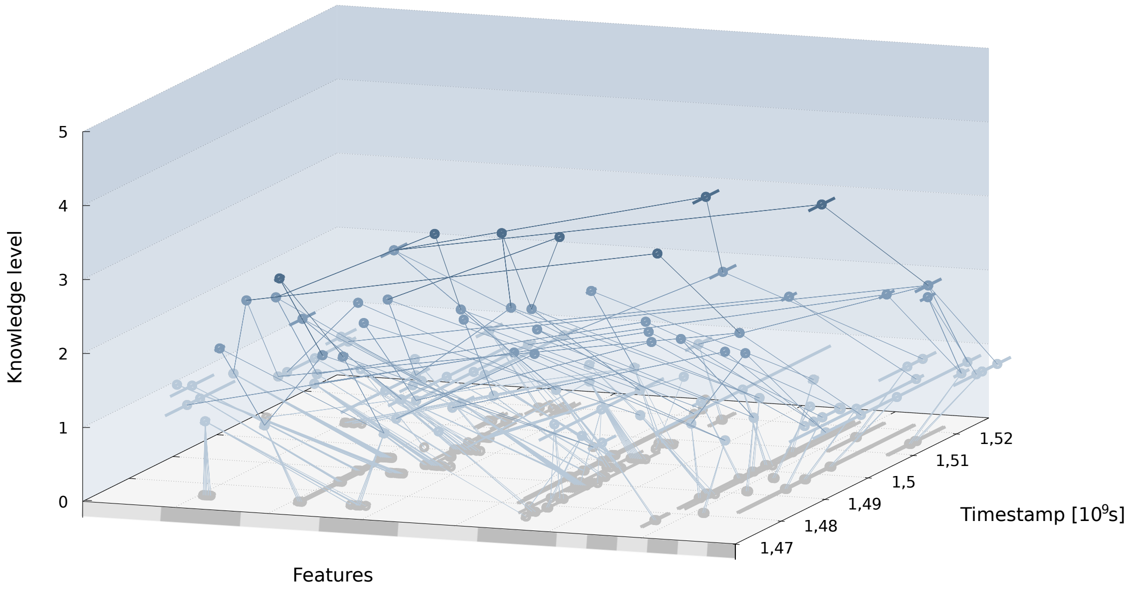

Figure 6: Three-dimensional visualization of a hierarchical knowledge base acquired by constructivist machine learning. Nodes represent machine models, except for level 0 whether nodes indicate which of the 349 features of the input data are used. Gray bars at the x-axis indicate sets of similar features of the input data. Graph edges indicate connections betweens machine models.

base and $\mathcal{M}_{old}$ is discarded; otherwise, $\mathcal{M}_{old}$ is retained and $\mathcal{M}_{new}$ discarded. This regime is referred to as opportunistic.

Knowledge Abstraction. A $T\Sigma$ relationship provides the basis to construct a new model on the next higher level of the knowledge base. In this case, both models share a congruent temporal validity and a common set of model subjectives while differing in their model purpose. First, the newly reconstructed model is stored to the knowledge base. The old model from the knowledge base is left unaltered. Using the outputs, or target values respectively, of the $T\Sigma$-related models, a new learn block without target values is formed. This learn block is assigned a higher level than the underlying models possess in the knowledge base and transferred to a construction process, from which all further learning processes may be passed. Thereby, repeated abstraction from a single learn block is possible. Knowledge abstraction may be limited by a user-defined maximum of knowledge levels.

IV. Conclusions and Future Work

With this work, we have defined implementation principles for a constructivist machine learning framework[1]. We have demonstrated that by combining Stachowiak's General Model Theory and constructivist learning theories, machine learning algorithms can be used to create, exploit and maintain hierarchical knowledge bases. In contrast to classical machine learning, this allows for an explicit representation of acquired knowledge. Further, the employed meta data structures support decisions on the applicability of a given model on a given input. High intersubjectivity and low ambiguity can be achieved for learned models and knowledge bases by implementing consent-oriented multi-algorithm supervised learning. The suggested deconstruction mechanisms allow to update a knowledge base automatically. Moreover, the deconstruction process defined even facilitates automated knowledge abstraction based on existing models of the knowledge base (Fig. 6). Given these features, constructivist machine learning is an ideal framework for applications in which diverse data sources need to be integrated, knowledge needs to be both assessible and automatically updated, and where ambiguity has to be resolved.

Based on the presented principles, we will extend our approach of combining machine learning and knowledge engineering in the future. While using generic descriptors for model purposes already allows to create a generic ontology, e.g., it is in practice desirable to match automatically learned knowledge representations with existing ontologies. By this, it may become possible to transform existing knowledge bases at least partially into automatically managed systems. Further, we need to emphasize that our approach is currently focusing on conceptual knowledge, but not limited to this domain. Future work will in particular include work on procedural knowledge and on ways to combine conceptual and procedural knowledge in a meta-cognitive domain.

Acknowledgements

I would like to thank Dmitrij Denisenko, Florian Große, Dennis Carrer and Michael Hermelschmidt for reviewing and discussing implementation details and working on re-implementations of the original conML prototype.

[1]Currently, we are implementing conML for Python and other languages. The project is available online from our Git repository at `http://git.informatik.uni-leipzig.de/ml-group`

References

Anderson, L. W., and Krathwohl, D. R. 2001. *A taxonomy for learning, teaching and assessing: a revision of Bloom's taxonomy of educational objectives.* New York: Longman.

Baçao, F.; Lobo, V.; and Painho, M. 2005. Self-organizing maps as substitutes for k-means clustering. In *Computational Science–ICCS 2005*. Springer. 476–483.

Bloom, B. S. 1956. *Taxonomy of educational objectives. Vol. 1: cognitive domain.* New York: McKay.

Chavent, M.; Kuentz-Simonet, V.; Liquet, B.; and Saracco, J. 2012. ClustOfVar: an R package for the clustering of variables. *Journal of Statistical Software* 50(13).

Davis, R.; Shrobe, H.; and Szolovits, P. 1993. What is a knowledge representation? *AI Magazine* 14(1):17–33.

Drescher, G. L. 1989. *Made-up minds : a constructivist approach to artificial intelligence.* Ph.D. Dissertation, Massachusetts Institute of Technology.

Falkner, A., and Haselböck, A. 2013. Challenges of knowledge evolution in practice. *AI Communications* 26(1):3–14.

Fox, R. 2001. Constructivism examined. *Oxford Review of Education* 27(1):23–35.

Gallant, S. I. 1988. Connectionist expert systems. *Commun. ACM* 31(2):152169.

Guisado-Gámez, J.; Dominguez-Sal, D.; and Larriba-Pey, J.-L. 2013. Massive query expansion by exploiting graph knowledge bases. arXiv 1310.5698.

Hayes, A. F., and Krippendorff, K. 2007. Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures* 1(1):77–89.

Hendler, J. A. 1989. On the need for hybrid systems. *Connection Science* 1(3):227–229.

Herrmann, C. S. 1995. A hybrid fuzzy-neural expert system for diagnosis. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Vol. 1*, 494–500. San Francisco, CA, USA: Morgan Kaufmann Publishers.

Hertz, H. 1894. Die Prinzipien der Mechanik in neuem Zusammenhange dargestellt. In Hertz, H., ed., *Gesammelte Werke*, volume 3. Leipzig: Barth.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Hudson, D. L.; Cohen, M. E.; and Anderson, M. F. 1991. Use of neural network techniques in a medical expert system. *International Journal of Intelligent Systems* 6(2):213–223.

Jain, A. K. 2010. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* 31(8):651–666.

Karabatak, M., and Ince, M. C. 2009. An expert system for detection of breast cancer based on association rules and neural network. *Expert Systems with Applications* 36(2, Part 2):3465–3469.

Kidd, A. 2012. *Knowledge acquisition for expert systems: A practical handbook.* Springer Science & Business Media.

Krippendorff, K. 1970. Bivariate agreement coefficients for reliability of data. In Bortatta., ed., *Sociological Methodology*. 139–150.

Krippendorff, K. 2004. Reliability in content analysis: some common misconceptions and recommendations. *Human Communication Research* 30(3):411–433.

Levey, C. A. 1991. Neural network having expert system functionality. U.S. patent US5398300A.

Lifschitz, V.; Morgenstern, L.; and Plaisted, D. 2008. Knowledge representation and classical logic. In van Harmelen, F.; Lifschitz, V.; and Porter, B., eds., *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*. Elsevier. chapter 1, 3–88.

Markov, N. T., and Kennedy, H. 2013. The importance of being hierarchical. *Current Opinion in Neurobiology* 23(2):187–194.

Martin, A.; Hinkelmann, K.; Gerber, A.; Lenat, D.; van Harmelen, F.; and Clark, P. 2019. Preface. In Martin, A.; Hinkelmann, K.; Gerber, A.; Lenat, D.; van Harmelen, F.; and Clark, P., eds., *Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019).*

Martinez-Gil, J. 2015. Automated knowledge base management: A survey. *Computer Science Review* 18:1–9.

Morik, K.; Kietz, B.-E.; Emde, W.; and Wrobel, S. 1993. *Knowledge Acquisition and Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers.

Quartz, S. R. 1993. Neural networks, nativism, and the plausibility of constructivism. *Cognition* 48(3):223–242.

Reich, K. 2004. *Konstruktivistische Didaktik. Lehren und Lernen aus interaktionistischer Sicht.* Munich: Luchterhan, 2nd edition.

Richardson, M., and Domingos, P. 2003. Building large knowledge bases by mass collaboration. In *Proceedings of the 2nd International Conference on Knowledge Capture*, K-CAP 03, 129–137. New York, NY, USA: Association for Computing Machinery.

Schmid, T. 2018. *Automatisierte Analyse von Impedanzspektren mittels konstruktivistischen maschinellen Lernens.* Ph.D. Dissertation, Leipzig.

Schmid, T. 2019. Deconstructing the final frontier of artificial intelligence: Five theses for a constructivist machine learning. In Martin, A.; Hinkelmann, K.; Gerber, A.; Lenat, D.; van Harmelen, F.; and Clark, P., eds., *Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019).*

Skeirik, R. D. 1990. Neural network/expert system process control system and method. U.S. patent US5121467A.

Stachowiak, H. 1973. *Allgemeine Modelltheorie*. Springer.

Subasic, P.; Yin, H.; and Lin, X. 2019. Building knowledge basethrough deep learningrelation extraction and wikidata. In Martin, A.; Hinkelmann, K.; Gerber, A.; Lenat, D.; van Harmelen, F.; and Clark, P., eds., *Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019).*