

# On Case-Based Learnability of Languages

Christoph Globig  
Universität Kaiserslautern  
FB Informatik  
PF 3049  
67653 Kaiserslautern, Germany  
globig@informatik.uni-kl.de

Klaus P. Jantke, Steffen Lange  
HTWK Leipzig  
FB IMN  
PF 30066  
04251 Leipzig, Germany  
{janos, steffen}@informatik.th-leipzig.de

Yasubumi Sakakibara  
Institute for Social Information Science  
Fujitsu Laboratories Ltd.  
140, Miyamoto, Numazu  
Shizuoka 410-03, Japan  
yasu@iias.flab.fujitsu.co.jp

This work summarizes and extends results from the authors' papers in the Proceedings of the 4th International Workshop on Analogical and Inductive Inference and the 5th International Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence 872, Springer-Verlag, Berlin Heidelberg, 1994, pp. 106 – 120 and pp. 532 – 546, respectively.

## Abstract

Case-based reasoning is deemed an important technology to alleviate the bottleneck of knowledge acquisition in Artificial Intelligence (AI). In case-based reasoning, knowledge is represented in the form of particular cases with an appropriate similarity measure rather than any form of rules. The case-based reasoning paradigm adopts the view that an AI system is dynamically changing during its life-cycle which immediately leads to learning considerations.

Within the present paper, we investigate the problem of case-based learning of indexable classes of formal languages. Prior to learning considerations, we study the problem of case-based representability and show that every indexable class is case-based representable with respect to a fixed similarity measure. Next, we investigate several models of case-based learning and systematically analyze their strengths as well as their limitations. Finally, the general approach to case-based learnability of indexable classes of formal languages is prototypically applied to so-called containment decision lists, since they seem particularly tailored to case-based knowledge processing.

# 1 Introduction

Case-based reasoning is deemed an important technology to alleviate the bottleneck of knowledge acquisition in Artificial Intelligence (AI). In case-based reasoning (cf. [Kol92], [Kol93], and [RS89]), knowledge is represented in the form of particular cases with an appropriate similarity measure rather than any form of rules. Those cases are collected during knowledge processing. For solving particular new problems, cases representing former experience are retrieved. The most similar cases are chosen as a basis for generating new solutions including techniques of case adaptation.

There is an enormous variety of case-based reasoning approaches. First, different approaches in case-based reasoning are characterized by the case representation and by the similarity concept in use (cf. [BB+93] for a remarkable collection of diverse case representation concepts within a single project, e.g.). Second, they are discriminated by the degree of automation in case retrieval and adaptation. Third, the invoked inference mechanisms may be quite different in character (cf. [CG94] for an evaluation of different case retrieval methods, e.g.).

Within case-based reasoning, case-based learning as investigated in [AKA91] is a natural way of designing learning procedures. The main task of case-based learning is to collect good cases which will be stored in the case base for describing knowledge and classifying unknown examples. Thus, case-based learning algorithms do not construct explicit generalizations from examples which most other supervised learning algorithms derive. Their hypotheses consist of case bases together with similarity concepts. Both constituents may be subject to learning.

In this paper, we investigate the power and the limitations of such case-based learning algorithms. A formal framework for case-based learning has recently been developed by [Jan92] and [JL93] in an inductive inference manner. Inductive inference is the very theory of learning from usually incomplete information (cf. [AS83], e.g.). In the majority of classical inductive inference publications, a learning device (frequently called an inductive inference machine) is just any computational apparatus meeting certain learning conditions. In other words, it is a computable function. Learning algorithms which have an architecture and a behavior motivated by artificial intelligence research seem particularly relevant both to the theory of inductive inference and to AI applications. A case-based learning approach as advocated in the present paper requires a closer look at and allows a more detailed understanding of the internal mechanism of this type of an inductive inference machine. Thus, hopefully, it will be easier to interpret the quite theoretical results in the focused area of artificial intelligence.

Learning of formal languages is one of the most thoroughly studied areas of inductive inference. In particular, there is a remarkable amount of work on inductive inference of *indexable classes* of formal languages (cf. [Ang80] and [ZL95], e.g.). An indexable class is a set of formal languages that has an effective enumeration which admits a uniform procedure for deciding membership. When learning languages of some given indexable class, their corresponding indices are suitable hypotheses for expressing guesses during learning (cf. [ZL95],

for a quite comprehensive investigation into the importance of choosing the appropriate space of hypotheses). Those hypotheses are generated in response to certain usually incomplete information about some target object. In language learning, one may either present only positive information about the language to be identified or provide both positive and negative information. In the sequel, we exclusively deal with case-based learning from *informant*, i.e., from both examples and counterexamples.

When some target language is presented, it is assumed that every word over the underlying alphabet eventually occurs as an example or counterexample, respectively, within the potentially infinite informant describing this object. Any case-based learning device has two fundamental constituents of its hypotheses: case bases and similarity concepts. Thus, either it may focus on learning case bases by appropriately collecting examples and counterexamples, it may focus on learning by tuning similarity measures to the evidence provided, or it may try to dovetail collecting cases and adapting similarity concepts suitably. A generated hypothesis consisting of some case base together with some similarity measure is expressing the learning device's current guess with respect to some fixed underlying interpretation.

A key question prior to learnability is the representability problem under some given semantics. This is the question for the class of target objects, i.e., formal languages, specifiable with respect to a particular similarity concept with varying case bases, or vice versa. Here, we are not going into the details of several formal semantics as investigated in [Jan92] and [JL95]. In dependence on the class of similarity measures taken into account, a certain possibly restricted class of formal languages is specifiable. There is a tradeoff between expressiveness and naturalness. If the class of admissible similarity measures is too restrictive, this becomes the main obstacle to learnability. Otherwise, if arbitrary computable similarity measures are permitted, this allows suspicious and undesirable coding tricks during learning. This means that recursion-theoretic ideas can be rewritten in terms of case bases and highly complex similarity concepts without meeting any intuition about case-based reasoning, whereas certain seemingly natural classes of similarity measures do not admit learnability of all target languages. Therefore, our research is especially focused on the effects of learning similarity concepts by intuitively natural methods of tuning defining parameters.

Our general approach to case-based learning of indexable classes of formal languages is prototypically applied to so-called *containment decision lists*. The formal languages acceptable by containment decision lists as investigated in [SS92] may be considered an indexable class, for instance. In our investigations, we have chosen these languages as target objects of learning, as they seem particularly tailored to case-based knowledge processing. The key reason is that prototypical cases occur as immediate constituents of containment decision lists they are describing. Consequently, collecting the right cases means accumulating the basic syntactic constituents of appropriate acceptors. Because of this extremely close correspondence between cases and generalized descriptions, this area seems particularly promising for gaining new insights into fundamental phenomena of case-based learning. Similarly to case-based learning of pattern languages (cf. [JL95]), the results obtained in the sequel may be interpreted as *lower bounds* for the difficulties of case-based learning in a huge variety of further problem domains. Intuitively, if learning of containment decision lists necessarily

fails within some formal setting, although the information presented does directly provide information about structural details of the target object to be synthesized, this bears abundant evidence of the even larger problems faced to in learning domains which do not enjoy such an immediate relationship between pieces of information presented and objects to be identified.

The paper is organized as follows. Section 2 presents notations and definitions. The announced results concerning case-based representability of indexable classes of formal languages are established in Section 3. In Sections 4 and 5 we investigate several formal models of case-based learning and study both their strengths as well as their limitations. A detailed investigation of case-based learning of containment decision lists can be found in Section 6. Finally, Section 7 summarizes the results obtained.

## 2 Preliminaries

Let  $\mathbb{N}$  and  $\mathbb{Q}$  be the set of all natural numbers and of all rational numbers, respectively. We set  $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ , and  $\mathbb{Q}_{[0,1]} = \{x \mid x \in \mathbb{Q}, 0 \leq x \leq 1\}$ . By  $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$  we denote **Cantor's pairing function**, i.e.,  $\langle x, y \rangle = ((x + y)^2 + 3x + y)/2$  for all  $x, y \in \mathbb{N}$ .

Within the present paper, we assume familiarity with formal language theory (cf. [HU79], e.g.). Let  $\Sigma$  be any fixed finite alphabet of symbols and let  $\Sigma^*$  be the free monoid over  $\Sigma$ . We set  $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ , where  $\varepsilon$  denotes the empty string. The length of a string  $s \in \Sigma^*$  is denoted by  $|s|$ . Any subset  $L \subseteq \Sigma^*$  is called a language. By  $\bar{L}$  we denote the complement of  $L$ , i.e.,  $\bar{L} = \Sigma^* \setminus L$ .

Every total computable function  $\sigma : \Sigma^* \times \Sigma^* \rightarrow \mathbb{Q}_{[0,1]}$  is said to be a **similarity measure**.

Let  $L$  be a language and let  $i = (s_0, d_0), (s_1, d_1), (s_2, d_2), \dots$  be an infinite sequence of elements of  $\Sigma^* \times \{0, 1\}$  such that  $\text{range}(i) = \{s_k \mid k \in \mathbb{N}\} = \Sigma^*$ ,  $i^+ = \{s_k \mid k \in \mathbb{N}, d_k = 1\} = L$ , and  $i^- = \{s_k \mid k \in \mathbb{N}, d_k = 0\} = \bar{L}$ . Then we refer to  $i$  as an **informant**. If  $L$  is classified via an informant then we also say that  $L$  is presented by **positive and negative data**. Let  $L$  be a language. We denote by  $\text{info}(L)$  the set of all informants for  $L$ . Moreover, let  $i = ((s_x, d_x))_{x \in \mathbb{N}}$  be an informant, and let  $x$  be a number. Then  $i_x$  denotes the initial segment of  $i$  of length  $x + 1$ , e.g.,  $i_2 = (s_0, d_0), (s_1, d_1), (s_2, d_2)$ . Furthermore,  $i_x^+$  and  $i_x^-$  refer to the sets  $\{s_k \mid k \leq x, d_k = 1\}$  and  $\{s_k \mid k \leq x, d_k = 0\}$ , respectively. Additionally, we use  $\rho \cdot \tau$  to denote the sequence obtained by concatenating two given finite sequences  $\rho$  and  $\tau$ .

A class  $\mathcal{C}$  of non-empty languages is said to be an **indexable class** provided that there are an effective enumeration  $L_0, L_1, L_2, \dots$  of  $\mathcal{C}$ , i.e.,  $\mathcal{C} = \{L_j \mid j \in \mathbb{N}\}$ , and a total recursive predicate  $p : \mathbb{N} \times \Sigma^* \rightarrow \{0, 1\}$  such that

$$p(j, s) = \begin{cases} 1, & \text{if } s \in L_j, \\ 0, & \text{otherwise.} \end{cases}$$

An **inductive inference machine** (abbr. IIM) is an algorithmic device working as follows: The IIM takes as its input larger and larger initial segments of an informant and

it first outputs a hypothesis, i.e., a number encoding a certain computer program, and then requests the next input element. As usual, we interpret the hypotheses output by an IIM with respect to some suitably chosen hypothesis space  $\mathcal{L}$ . Since we exclusively deal with indexable classes  $\mathcal{C}$ , we always take as a hypothesis space an effective enumeration  $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$  of recursive languages. Clearly,  $\mathcal{L}$  must *comprise* the target indexable class  $\mathcal{C}$ , i.e.,  $\mathcal{C} \subseteq \{L_j \mid j \in \mathbb{N}\}$ . When an IIM outputs a number  $j$ , we interpret it to mean that the machine is hypothesizing the language  $L_j$ .

Let  $M$  be an IIM, let  $i$  be an informant, and let  $x \in \mathbb{N}$ . Then we use  $M(i_x)$  to denote the number output by  $M$  in response to input  $i_x$ . We define convergence of IIMs as usual. Let  $i$  be an informant, and let  $M$  be an IIM. The sequence  $(M(i_x))_{x \in \mathbb{N}}$  is said to **converge** to a number  $j$  iff all but finitely many terms of  $(M(i_x))_{x \in \mathbb{N}}$  are equal to  $j$ .

Now we are ready to define learning in the limit (cf. [Gol67]).

**Definition 1.** *Let  $\mathcal{C}$  be an indexable class, and let  $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$  be a hypothesis space that comprises  $\mathcal{C}$ . An IIM  $M$  identifies  $\mathcal{C}$  in the limit with respect to  $\mathcal{L}$  iff for every  $L \in \mathcal{C}$ , and for every  $i \in \text{info}(L)$ , there exists a  $j \in \mathbb{N}$  such that the sequence  $(M(i_x))_{x \in \mathbb{N}}$  converges to  $j$  and  $L = L_j$ .*

The following proposition states that every indexable class is learnable in the limit. Furthermore, every indexable class is identifiable in the limit with respect to every hypothesis space that comprises the target class (cf. [Gol67]).

**Proposition 1.** *Let  $\mathcal{C}$  be an indexable class, and let  $\mathcal{L}$  be a hypothesis space that comprises  $\mathcal{C}$ . Then, there is an IIM  $M$  that identifies  $\mathcal{C}$  in the limit with respect to  $\mathcal{L}$ .*

### 3 Case-Based Representability

A case-based learning algorithm, when successively processing information about a particular target object, is generally supposed to come up with some finite case base and some similarity measure describing the particular target object. This obviously assumes some interpretation of pairs built from case bases and similarity measures in terms of the objects under consideration. Formally speaking, one needs some well-defined semantics. In our study, the objects under consideration are formal languages. Representing a formal language by means of a finite set of labeled strings, also called cases, and a similarity function results in a finite description of an acceptor that is different from those usually used in formal language theory (cf. [HU79], e.g.).

In [JL93] two slightly different semantics have been introduced to describe formal languages in a case-based manner. In all what follows we restrict ourselves to only one semantics, the so-called *standard semantics* (cf. [JL93]).

**Definition 2.** *Let  $L \subseteq \Sigma^*$ , and let  $\sigma$  be any similarity measure.  $L$  is case-based representable with respect to  $\sigma$  iff there are finite sets  $CB^+ \subseteq L$  and  $CB^- \subseteq \bar{L}$  such that  $L = \{u \mid u \in \Sigma^*, \exists w \in CB^+ [\sigma(w, u) > 0 \wedge \forall v \in CB^- [\sigma(w, u) > \sigma(v, u)]]\}$ .*

Subsequently, we use the shorthands  $CB = \{(w, 1) \mid w \in CB^+\} \cup \{(v, 0) \mid v \in CB^-\}$  and  $L(CB, \sigma) = \{u \mid u \in \Sigma^*, \exists w \in CB^+[\sigma(w, u) > 0 \wedge \forall v \in CB^-[\sigma(w, u) > \sigma(v, u)]]\}$  to refer to the case base defined by  $CB^+$  and  $CB^-$  and the language represented by the pair  $(CB, \sigma)$ , respectively. We call  $(w, 1)$  a **positive case** and  $(v, 0)$  a **negative case**. Furthermore, if  $CB$  consists of only positive cases,  $CB$  is said to be a **positive case base**.

Clearly, given any non-empty recursive language  $L$ , one can easily define a similarity measure  $\sigma_L$  such that  $L$  is case-based representable with respect to  $\sigma_L$ . Moreover,  $\sigma_L$  can be selected in a way such that any positive case  $(w, 1)$ ,  $w \in L$ , forms a singleton case base that is suited to represent  $L$  (cf. the proof of Theorem 4 for some more details). Consequently, the more ambitious question is whether or not all objects of some target class  $\mathcal{C}$  can be represented in a uniform manner, i.e., by a similarity measure that guarantees case-based representability of every language in  $\mathcal{C}$ .

**Definition 3.** *Let  $\mathcal{C}$  be any indexable class of recursive languages.  $\mathcal{C}$  is **uniformly case-based representable** iff there exists a similarity measure  $\sigma$  such that every  $L \in \mathcal{C}$  is case-based representable with respect to  $\sigma$ .*

Whereas positive case bases are sufficient to represent every recursive language by means of an individual similarity measure, this approach is too restrictive to case-based represent quite simple indexable classes in a uniform manner. In order to make this explicit we use the following notion. A language class is said to be **superfinite** if it contains all finite languages and at least one infinite language.

**Lemma 1.** *Let  $\mathcal{C}$  be any superfinite indexable class. Then,  $\mathcal{C}$  is not uniformly case-based representable, if exclusively positive case bases are admissible.*

*Proof.* This proof is quite simple. The key problem is that, given any singleton language  $L = \{w\}$ ,  $w \in \Sigma^*$ , there is exactly one positive case base for  $L$ , namely  $CB_L = \{(w, 1)\}$ . This implies that the similarity measure is constrained to satisfy  $\sigma(w, v) > 0$  iff  $w = v$ . Note that all these similarity measures can be normalized by increasing every  $\sigma(w, v) > 0$  to  $\sigma(w, v) = 1$  without changing expressiveness. Thus, up to normalization, there is a unique similarity measure able to characterize all singleton languages: the recursive predicate testing equality over  $\Sigma^*$ . But this does not permit the definition of a finite case base for the infinite language belonging to  $\mathcal{C}$ . This completes the proof.  $\square$

Obviously, the recursive predicate testing equality over  $\Sigma^*$  is well-suited to represent each finite language by simply storing up all its elements. Thus, the main difficulty consists in representing simultaneously finite and infinite languages. In order to overcome this problem, positive and negative cases are needed.

**Theorem 2.** *Let  $\mathcal{C}$  be an indexable class of recursive languages. Then,  $\mathcal{C}$  is uniformly case-based representable.*

*Proof.* Set  $\hat{\mathcal{C}} = \mathcal{C} \cup \{\Sigma^*\}$ , and select any effective enumeration  $(L_j)_{j \in \mathbb{N}}$  of  $\hat{\mathcal{C}}$  satisfying  $L_{2j+1} \in \mathcal{C}$  and  $L_{2j} = \Sigma^*$  for all  $j \in \mathbb{N}$ . Furthermore, let  $(w_k)_{k \in \mathbb{N}}$  be any effective repetition free enumeration of all strings in  $\Sigma^*$ .

First, we define a unary total recursive function  $r$  that assigns to each string  $w_k$  a par-

ticular language  $L_{r(k)}$ . Afterwards, we define a similarity measure  $\sigma$  in a way such that  $w_k$  serves as a *representative case* for  $L_{r(k)}$ , i.e.,  $L(\{(w_k, 1)\}, \sigma) = L_{r(k)}$ .

Initially, set  $r(0) = 0$ . We proceed inductively. Let  $k \in \mathbb{N}$ . We set  $r(k+1) = j$ , if  $j$  is the least index  $\hat{j} \leq 2k$  satisfying  $w_{k+1} \in L_{\hat{j}}$  and  $r(n) \neq \hat{j}$  for all  $n \leq k$ . By definition,  $r$  is total recursive. Furthermore, for each  $j \in \mathbb{N}$ , we have:

**Observation.** If  $L_j$  is infinite, then there is a  $k \in \mathbb{N}$  such that  $r(k) = j$ .

Next, we define the desired similarity measure  $\sigma$ . Let  $k, n \in \mathbb{N}$ . We set:

$$\sigma(w_k, w_n) = \begin{cases} 1, & \text{if } w_k = w_n, \\ 1 - \frac{1}{k+2}, & \text{if } w_n \in L_{r(k)} \setminus \{w_k\}, \\ 0, & \text{otherwise.} \end{cases}$$

By construction,  $L(\{(w_k, 1)\}, \sigma) = L_{r(k)}$  for all  $k \in \mathbb{N}$ , and therefore  $w_k$  serves as a representative case for  $L_{r(k)}$ .

**Claim.**  $\mathcal{C}$  is case-based representable with respect to  $\sigma$ .

Let  $L$  be any language in  $\mathcal{C}$ , and let  $j$  be the least index that meets  $L_j = L$ . We distinguish the following cases.

*Case 1.*  $L_j$  is an infinite language.

By the above observation there exists a  $k$  with  $r(k) = j$ . Hence,  $L(\{(w_k, 1)\}, \sigma) = L_j$ , and we are done.

*Case 2.*  $L_j$  is a finite language.

Let  $m = \max\{z \mid w_z \in L_j\}$ . Since  $L_{2n} = \Sigma^*$  for all  $n \in \mathbb{N}$ , there exists, by the above observation, some  $k > m$  such that  $w_k$  is a representative case for  $\Sigma^*$ . Now, set  $CB_j = \{(w_k, 0)\} \cup \{(w, 1) \mid w \in L_j\}$ . By  $\sigma$ 's definition  $L_j \subseteq L(CB_j, \sigma)$ . Furthermore, since  $L_{r(k)} = \Sigma^*$ ,  $z < k$  implies  $\sigma(w_k, w_\ell) = 1 - \frac{1}{k+2} > 1 - \frac{1}{z+2} \geq \sigma(w_z, w_\ell)$  for all  $\ell \in \mathbb{N} \setminus \{z, k\}$ . This, in particular, results in  $\sigma(w_k, v) > \sigma(w, v)$  for all  $w \in L_j$ ,  $v \in \overline{L_j}$ . Thus,  $L(CB_j, \sigma) = L_j$  follows, and the theorem is proved.  $\square$

Having a closer look at the proof above, one verifies that mainly the “there exists” flavor of Definition 2 is exploited in order to assign to every language  $L_j \in \mathcal{C}$  a corresponding case base. Obviously, in case that it is uniformly decidable whether or not a given language  $L_j$  is finite, one may *effectively* construct a case base  $CB_j$  for each  $L_j$ , i.e.,  $L(CB_j, \sigma) = L_j$ . Furthermore, if finiteness is undecidable, the used definition of the corresponding case bases is, clearly, *non-effective*.

Within the proof of our next result, we provide a way to overcome the mentioned peculiarities of the construction underlying the proof of Theorem 2. However, our overall goal is a bit more ambitious. We show that case bases consisting of exactly two cases are still sufficient to represent each language of any given indexable class in a uniform manner. Furthermore, this upper bound is tight. To see this note that at least one positive case has to be stored up in the case base to represent any non-empty language (cf. Definition 2). By Lemma 1 we

know that positive case bases are not sufficient to uniformly represent any given superfinite indexable class  $\mathcal{C}$ , and thus there has to be at least one  $L \in \mathcal{C}$  that requires a case base containing one positive and one negative case.

**Theorem 3.** *Let  $\mathcal{C}$  be an indexable class of recursive languages. Then, there is a similarity measure  $\sigma$  such that every  $L \in \mathcal{C}$  is case-based representable with respect to  $\sigma$  by a case base that contains 2 cases.*

*Proof.* Choose any effective enumeration  $(L'_j)_{j \in \mathbb{N}}$  of  $\mathcal{C}$ , and set  $\hat{\mathcal{C}} = \mathcal{C} \cup \{\Sigma^*\}$ . Furthermore, let  $(w_k)_{k \in \mathbb{N}}$  be any effective repetition free enumeration of all strings in  $\Sigma^*$ . First, we construct a suitable enumeration  $(L_j)_{j \in \mathbb{N}}$  of  $\hat{\mathcal{C}}$ . Afterwards, we define the wanted similarity measure  $\sigma$  and an algorithm  $\mathcal{A}$  that, given any  $j \in \mathbb{N}$ , computes a case base  $CB_j$  for  $L_j$ , i.e.,  $L(CB_j, \sigma) = L_j$ .

Initially, we set  $L_0 = \Sigma^*$ . We proceed inductively. Let  $x \geq 1$ . For all  $z \leq x$  do the following: Test, for all  $y \leq x - 1$ , whether or not

$$\{w_r \mid r \leq x, w_r \in L'_z\} \neq \{w_r \mid r \leq x, w_r \in L_y\}.$$

In case there is at least one index  $z \leq x$  fulfilling the above test for all  $y \leq x - 1$ , then select the minimal one and set  $L_x = L'_z$ . Otherwise, set  $L_x = \Sigma^*$ .

Note that  $(L_j)_{j \in \mathbb{N}}$  is an effective enumeration of  $\hat{\mathcal{C}}$  having a decidable equality problem, i.e., there is a recursive predicate  $q$  such that, for all  $j, k \in \mathbb{N}$ ,  $q(j, k) = 1$  iff  $L_j = L_k$ .

Now, we are ready to define the desired similarity. Let  $j, k, n, m \in \mathbb{N}$ .

$$\sigma(w_{\langle j, n \rangle}, w_{\langle k, m \rangle}) = \begin{cases} 1, & \text{if } w_{\langle j, n \rangle} = w_{\langle k, m \rangle}, \\ 1 - \frac{1}{\langle j, n \rangle + 2}, & \text{if } w_{\langle j, n \rangle} \in L_j, w_{\langle k, m \rangle} \in L_j \setminus \{w_{\langle j, n \rangle}\}, \\ \frac{1}{\langle j, n \rangle + 2}, & \text{if } w_{\langle j, n \rangle} \in L_j, w_{\langle k, m \rangle} \in \overline{L_j}, \\ 1 - \frac{1}{\langle j, n \rangle + 2}, & \text{if } w_{\langle j, n \rangle} \in \overline{L_j}, w_{\langle k, m \rangle} \in \overline{L_j} \setminus \{w_{\langle j, n \rangle}\}, \\ \frac{1}{\langle j, n \rangle + 2}, & \text{if } w_{\langle j, n \rangle} \in \overline{L_j}, w_{\langle k, m \rangle} \in L_j. \end{cases}$$

Obviously,  $\sigma$  is computable. The above definition provides a different concept of representative cases. By definition, every string  $w \in \Sigma^*$  serves as a representative case for  $\Sigma^*$ , i.e.,  $L(\{w, 1\}, \sigma) = \Sigma^*$ . Furthermore, almost every string  $w_{\langle j, z \rangle} \in \Sigma^*$  may play the role of a representative case for a language  $L_j \neq \Sigma^*$  provided we put it together with a well-chosen string  $u \neq w_{\langle j, z \rangle}$  into a case base for the target language  $L_j$ . More formally, we have the following observation.

**Observation.** Let  $L_j \neq \Sigma^*$ , let  $w_a \in L_j$  and  $w_b \notin L_j$ , and let  $z > a + b$ . Then, we have:  $L(\{(w_b, 0), (w_{\langle j, z \rangle}, 1)\}, \sigma) = L_j$ , if  $w_{\langle j, z \rangle} \in L_j$ , and  $L(\{(w_a, 1), (w_{\langle j, z \rangle}, 0)\}, \sigma) = L_j$ , otherwise.

Suppose that  $w_{\langle j, z \rangle} \in L_j$ , and set  $CB_j = \{(w_b, 0), (w_{\langle j, z \rangle}, 1)\}$ . Clearly,  $w_b \notin L(CB_j, \sigma)$ , but  $w_{\langle j, z \rangle} \in L(CB_j, \sigma)$ . Let  $v$  be any string being different from  $w_b$  and  $w_{\langle j, z \rangle}$ , respectively. First, assume  $v \in L_j$ . Since  $\langle j, z \rangle \geq z > b$ , we obtain  $\sigma(w_{\langle j, z \rangle}, v) = 1 - \frac{1}{\langle j, z \rangle + 2} > 1 - \frac{1}{b+2} \geq \sigma(w_b, v)$

and, therefore,  $v \in L(CB_j, \sigma)$ . Second, let  $v \notin L_j$ . Now,  $\langle j, z \rangle > b$  implies  $\sigma(w_{\langle j, z \rangle}, v) = \frac{1}{\langle j, z \rangle + 2} < \frac{1}{b+2} \leq \sigma(w_b, v)$  and, consequently,  $v \notin L(CB_j, \sigma)$ . Hence,  $L_j = L(CB_j, \sigma)$ .

Obviously, the same arguments apply *mutatis mutandis* to handle the remaining part, i.e., the  $w_{\langle j, z \rangle} \notin L_j$  case, and the above observation follows.

**Claim.**  $\mathcal{C}$  is case-based representable with respect to  $\sigma$ .

Now, we define the desired algorithm  $\mathcal{A}$ . On input  $j \in \mathbb{N}$ ,  $\mathcal{A}$  executes (A1) and (A2).

- (A1) Check whether or not  $L_j = L_0$ .  
 In case it is, set  $CB_j = \{(w_0, 1), (w_1, 1)\}$ , output  $CB_j$ , and stop.  
 Otherwise, goto (A2).
- (A2) Determine the least  $a, b \in \mathbb{N}$  such that  $w_a \in L_j$  and  $w_b \notin L_j$ , and fix  $n \in \mathbb{N}$  with  $n = \langle j, a + b + 1 \rangle$ .  
 If  $w_n \in L_j$ , set  $CB_j = \{(w_b, 0), (w_n, 1)\}$ , output  $CB_j$ , and stop.  
 Otherwise, set  $CB_j = \{(w_a, 1), (w_n, 0)\}$ , output  $CB_j$ , and stop.

Clearly, Instruction (A1) is effectively executable. Since  $L_j \neq \emptyset$  for all  $j \in \mathbb{N}$ , Instruction (A2) is effectively executable as well, and thus  $\mathcal{A}$  is an algorithm that assigns to every language  $L_j$  a finite case base  $CB_j$  consisting of two cases. Furthermore,  $L(CB_j, \sigma) = L_j$  as the above observation witnesses.  $\square$

Clearly, the definition of the used case bases is indeed *effective*. This nicely contrasts the construction underlying the demonstration of Theorem 2. The above proof has another interesting feature we want to point out. Applying the stated observation it can be easily verified that one can *effectively* assign to every language  $L_j$  infinitely many case bases. As we will see, the latter makes the constructed similarity measure a good choice for case-based learning algorithms.

## 4 Case-Based Learning

In the sequel, we discuss the power as well as the limitations of case-based language learning. Obviously, this requires a precise notion of what does it mean to learn a target language in a case-based manner. The model we are going to investigate adapts the notion of learning in the limit (cf. Definition 1).

Similarly to an IIM, a **case-based learner** (abbr. CBL) is an algorithmic device that takes as its input larger and larger initial segments of an informant and it first outputs a hypothesis, and then requests the next input element. However, the hypotheses output by a case-based learner are different from those generated by an ordinary IIM. Intuitively, when fed an initial segment  $i_x$  of some informant  $i$ , a CBL  $M$  is requested to return a pair consisting of a finite set of labeled strings and a similarity measure, denoted by  $CB_x$  and  $\sigma_x$ , respectively. More formally,  $M$  is supposed to output a pair  $(k_x, j_x)$ , where  $k_x$  is encoding the

finite set  $CB_x$  and  $j_x$  is encoding a certain computer program for  $\sigma_x$ . Subsequently, we drop the related technicalities and refer to the hypothesis  $M$  computes when fed  $i_x$  by  $(CB_x, \sigma_x)$ .

Let  $i$  be an informant, and let  $M$  be a CBL. The sequence  $(M(i_x))_{x \in \mathbb{N}}$  of  $M$ 's outputs is said to **converge** to a pair  $(CB, \sigma)$  iff all but finitely many terms of  $(M(i_x))_{x \in \mathbb{N}}$  are equal to  $(CB, \sigma)$ .

**Definition 4.** *Let  $\mathcal{C}$  be an indexable class, and let  $\mathcal{S}$  be an effectively enumerable class of similarity measures. A CBL  $M$  case-based learns  $\mathcal{C}$  with respect to  $\mathcal{S}$  iff for every  $L \in \mathcal{C}$ , and for every  $i \in \text{info}(L)$ , the following conditions are satisfied:*

- (1) *for all  $x \in \mathbb{N}$ ,  $CB_x \subseteq i_x^+ \times \{1\} \cup i_x^- \times \{0\}$  and  $\sigma_x \in \mathcal{S}$ , where  $(CB_x, \sigma_x) = M(i_x)$ ,*
- (2) *there exists a pair  $(CB, \sigma)$  such that the sequence  $(M(i_x))_{x \in \mathbb{N}}$  converges to  $(CB, \sigma)$  and  $L = L(CB, \sigma)$ .*

Finally,  $\mathcal{C}$  is said to be case-based learnable iff there are an effectively enumerable class of similarity measures  $\mathcal{S}$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  with respect to  $\mathcal{S}$ .

Our first result points to one strength of this approach. That is, every indexable class is case-based learnable if the underlying class of similarity measures is appropriately chosen. The proof of Theorem 4 is based on an idea taken from [Jan92].

**Theorem 4.** *Let  $\mathcal{C}$  be an indexable class. Then, there are an effectively enumerable class of similarity measures  $\mathcal{S}$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  with respect to  $\mathcal{S}$ .*

*Proof.* Let  $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$  be any effective enumeration of the target class  $\mathcal{C}$ . By Proposition 1, there is an IIM  $\hat{M}$  which identifies  $\mathcal{C}$  in the limit with respect to the hypothesis space  $\mathcal{L}$ . The desired CBL  $M$  simulates  $\hat{M}$ , and case-based learns  $\mathcal{C}$  with respect to the following class of similarity measures  $\mathcal{S}$ .

For every  $j \in \mathbb{N}$ , and for all  $u, v \in \Sigma^*$ , we set:

$$\sigma_j(u, v) = \begin{cases} 1, & \text{if } u \in L_j, v \in L_j, \\ 0, & \text{otherwise.} \end{cases}$$

Let  $\mathcal{S} = \{\sigma_j \mid j \in \mathbb{N}\}$ . Since membership is uniformly decidable for  $\mathcal{L}$ , one immediately sees that  $\mathcal{S}$  is an effectively enumerable class of similarity measures. Moreover, every non-empty positive case base  $CB \subseteq L_j \times \{1\}$  can be used to represent the language  $L_j$  with respect to its individual similarity measure  $\sigma_j$ , i.e.,  $L_j = L(CB, \sigma_j)$ .

Let  $L \in \mathcal{C}$ , let  $i = ((s_x, d_x))_{x \in \mathbb{N}}$  be any informant for  $L$ , and let  $x \in \mathbb{N}$ . On input  $i_x$ ,  $M$  proceeds as follows. If  $i_x^+ = \emptyset$ , then  $M$  outputs the empty case base and the similarity measure  $\sigma_0$ . Otherwise,  $M$  determines the least  $y \leq x$  with  $d_y = 1$ , simulates  $\hat{M}$  on input  $i_x$ , and outputs the singleton case base  $\{s_y, 1\}$  together with the similarity measure  $\sigma_{\hat{M}(i_x)}$ .

Since  $L \neq \emptyset$ , the CBL  $M$  sometimes receives a first positive case, say  $(s_y, 1)$ , and outputs in this and every subsequent step the same case base  $CB = \{(s_y, 1)\}$ . Finally, since  $\hat{M}$  identifies  $\mathcal{C}$  in the limit with respect to the hypothesis space  $\mathcal{L}$ , the IIM  $\hat{M}$ , when fed the informant  $i$  for  $L$ , converges to a number  $j$  with  $L_j = L$ . Therefore,  $M$  converges to the pair  $(CB, \sigma_j)$ . Clearly,  $L_j = L(CB, \sigma_j)$ , and thus  $M$  case-based learns  $L$ .  $\square$

A closer look at the above construction shows that one may design remarkably powerful case-based learners that do not care so much about the problem which cases they are going to store in the case base. Thus, one is tempted to conclude that case-based learning is only the task of suitably adapting the used similarity measure. However, the quite opposite approach, i.e., designing case-based learners that carefully select appropriate cases to be stored within the case base and that do not change the used similarity measure at all, turns out to be successful as well.

**Theorem 5.** *Let  $\mathcal{C}$  be an indexable class. Then, there are a similarity measure  $\sigma$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  with respect to  $\mathcal{S} = \{\sigma\}$ .*

*Proof.* Choose any effective repetition free enumeration  $(w_k)_{k \in \mathbb{N}}$  of all strings in  $\Sigma^*$ , and any effective enumeration  $(L'_j)_{j \in \mathbb{N}}$  of  $\mathcal{C}$ . Based on both enumerations, define the enumeration  $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$  of  $\hat{\mathcal{C}} = \mathcal{C} \cup \{\Sigma^*\}$  and the similarity measure  $\sigma$  as demonstrated in the proof of Theorem 3. For every string  $s \in \Sigma^*$ , let  $\#(s) = k$  be the uniquely determined index  $k$  with  $w_k = s$ . Note that  $\#(s)$  can be effectively determined.

By Proposition 1, there exists an IIM  $\hat{M}$  which identifies  $\mathcal{C}$  in the limit with respect to the hypothesis space  $\mathcal{L}$ . Again, the wanted case-based learner  $M$  uses the IIM  $\hat{M}$  as a subroutine. So let  $L \in \mathcal{C}$ , and let  $i$  be any given informant for  $L$ , and let  $x \in \mathbb{N}$ . On input  $i_x$ ,  $M$  works as described in Figure 1.

---

**Case-based learner  $M$ :**

**begin**

  read  $i_x = (s_0, d_0), \dots, (s_x, d_x)$ ;

**if**  $i_x^+ = \emptyset$  **or**  $i_x^- = \emptyset$

**then**  $CB_x := \{(s_0, d_0)\}$ ;

**else**  $j := \hat{M}(i_x)$ ;  $y_p := \min\{y \mid y \leq x, d_y = 1\}$ ;  $a := \#(s_{y_p})$ ;

$y_n := \min\{y \mid y \leq x, d_y = 0\}$ ;  $b := \#(s_{y_n})$ ;  $n := \langle j, a + b + 1 \rangle$ ;

**if**  $w_n \notin i_x^+ \cup i_x^-$

**then**  $CB_x := \emptyset$

**else if**  $w_n \in i_x^+$

**then**  $CB_x := \{(w_b, 0), (w_n, 1)\}$

**else**  $CB_x := \{(w_a, 1), (w_n, 0)\}$ ;

  output  $(CB_x, \sigma)$ ;

**end**

---

Figure 1: Case-based learning with respect to a fixed similarity measure.

Clearly,  $M$  is a case-based learner that works with respect to  $\mathcal{S} = \{\sigma\}$ . In order to verify  $M$ 's correctness we distinguish the following cases.

*Case 1.*  $L = \Sigma^*$ .

Hence,  $i_x^- = \emptyset$  for all  $x \in \mathbb{N}$ , and therefore  $M$  converges to the pair  $(\{(s_0, 1)\}, \sigma)$ . By

definition of  $\sigma$  (cf. the demonstration of Theorem 3),  $L = (\{(s_0, 1)\}, \sigma)$ , and we are done.

*Case 2.*  $L \neq \Sigma^*$ .

Thus,  $M$  must eventually receive a positive and a negative case. Let  $y_p$  and  $y_n$  be the least indices such that an element  $(s_{y_p}, 1)$  and  $(s_{y_n}, 0)$ , respectively, occur in  $i$ , and let  $a = \#(s_{y_p})$  and  $b = \#(s_{y_n})$ . By definition, on input  $i_x$ ,  $x \geq \max\{y_p, y_n\}$ , the CBL  $M$  simulates  $\hat{M}$  on input  $i_x$ . Since  $\hat{M}$ , when successively fed  $i$ , converges to a correct number  $j$  for  $L$ , i.e.,  $L_j = L$ , there is an  $\hat{x} \geq x$  such that  $\hat{M}(i_{\hat{x}+r}) = j$  for all  $r \in \mathbb{N}$ . Consequently,  $M$ , when processing the initial segment  $i_{\hat{x}}, i_{\hat{x}+1}, i_{\hat{x}+2}, \dots$ , respectively, is always determining the same index  $n$ , i.e.,  $n = \langle j, a + b + 1 \rangle$ . Furthermore, there is a least  $y \in \mathbb{N}$  such that  $w_n = s_y$ , and thus  $\hat{M}$  converges as well, say to the pair  $(CB, \sigma)$ , where  $CB = \{(w_b, 0), (w_n, d_y)\}$  and  $CB = \{(w_a, 1), (w_n, d_y)\}$ , respectively. Finally, having a closer look at the observation stated within the demonstration of Theorem 3, one immediately sees that  $L(CB, \sigma)$  equals  $L_j$ . Finally, since  $L_j = L$ , the CBL  $M$  case-based learns  $L$  as required and, thus, the theorem follows.  $\square$

## 5 Case-Based Learning by Collecting Cases

The case-based reasoning paradigm adopts the view that an AI system is dynamically changing during its life-cycle. These changes may be required thanks to slight and natural changes with time in the goals of the reasoning system.

Case-based learning, in particular, refers to the learning behavior of an AI system which proceeds by collecting cases and tuning similarity concepts throughout an extended period of time. Thus, a more natural formalization should not include access to the whole history of information processing. Basic approaches of this type of learning have been provided by [Jan92] and [JL93].

A crucial difficulty, then, of such CBL systems is that they are constrained to build their case bases from the *current* information available: the current data. They are restricted, then, in not being able to dig into their entire history of data received. This characteristic was called *operational incremental learning* in [Jan93].

**Definition 5.** *Let  $\mathcal{C}$  be an indexable class, and let  $\mathcal{S}$  be an effectively enumerable class of similarity measures. A CBL  $M$  case-based learns  $\mathcal{C}$  by collecting cases with respect to  $\mathcal{S}$  iff*

- (1)  $M$  case-based learns  $\mathcal{C}$  with respect to  $\mathcal{S}$ ,
- (2) for every  $L \in \mathcal{C}$  and for every  $i = ((s_x, d_x))_{x \in \mathbb{N}} \in \text{info}(L)$ , the following condition is satisfied:  $CB_0 \subseteq \{(s_0, d_0)\}$ , and  $CB_{x-1} \subseteq CB_x \subseteq CB_{x-1} \cup \{(s_x, d_x)\}$ , for all  $x \geq 1$ , where  $(CB_x, \sigma_x) = M(i_x)$ .

Finally,  $\mathcal{C}$  is said to be case-based learnable by collecting cases iff there are an effectively enumerable class of similarity measures  $\mathcal{S}$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  by collecting cases with respect to  $\mathcal{S}$ .

In contrast to Definition 4, a case-based learner  $M$  that learns by collecting cases is not allowed to delete any case that was once included in its case base. Moreover, when processing the initial segment  $i_{x+1} = (s_0, d_0), \dots, (s_{x+1}, d_{x+1})$ ,  $M$  is constrained to store up, if any, the new case  $(s_{x+1}, d_{x+1})$ , only. Since it may happen that a particular case appears exactly once in the informant presented,  $M$  has carefully to examine whether or not it adds the new case  $(s_{x+1}, d_{x+1})$  to its actual case base  $CB_x$ .

On a first glance, one may expect that these additional constraints on the behavior of a case-based learner result in a remarkable loss of learning power. However, one straightforwardly verifies that the CBL  $M$  used within the demonstration of Theorem 4 fulfills the requirements of Definition 5. Thus, we may conclude:

**Corollary 6.** *Let  $\mathcal{C}$  be an indexable class. Then, there are an effectively enumerable class of similarity measures  $\mathcal{S}$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  by collecting cases with respect to  $\mathcal{S}$ .*

Naturally, the question arose whether or not Theorem 5 remains valid, too, if case-based learners are requested to learn by collecting cases. As our next theorem rigorously proves, a case-based learner of the latter type must generally be allowed to change the used similarity measure within the learning process.

**Theorem 7.** *Let  $\mathcal{C}$  be an indexable class that contains all finite and all co-finite languages, and let  $\sigma$  be any similarity measure. Then, there is no CBL  $M$  that case-based learns  $\mathcal{C}$  by collecting cases with respect to  $\mathcal{S} = \{\sigma\}$ .*

*Proof.* Assume to the contrary that there are a similarity measure  $\sigma$  and a CBL  $M$  which learns  $\mathcal{C}$  with respect to  $\mathcal{S} = \{\sigma\}$ . From this assumption, we will derive a contradiction.

We start with the following claim which, in particular, states that there exists a **locking sequence** for the infinite language  $\Sigma^* \in \mathcal{C}$  (cf. [BB75]).

**Claim.** There exist an  $i \in \text{info}(\Sigma^*)$  and an  $x \in \mathbb{N}$  such that  $M(i_x) = M(i_x \cdot (u, 1))$  for all  $u \in \Sigma^*$ .

Suppose the converse, i.e., for all  $i \in \text{info}(\Sigma^*)$  and for every  $x \in \mathbb{N}$  there is a string  $u \in \Sigma^*$  such that  $M(i_x) \neq M(i_x \cdot (u, 1))$ . Based on this assumption we define a particular informant  $i$  for  $\Sigma^*$  on which  $M$  performs infinitely many mind changes. Let  $(w_k)_{k \in \mathbb{N}}$  be any repetition free enumeration of all strings in  $\Sigma^*$ . By assumption, there is a  $y \in \mathbb{N}$  such that  $M((w_0, 1)) \neq M((w_0, 1) \cdot (w_y, 1))$ . Set  $i^{(0)} = (w_0, 1) \cdot (w_y, 1) \cdot (w_1, 1)$ . We proceed inductively. Let  $k \in \mathbb{N}$ , and  $i^{(k)}$  be the finite sequence that has been already defined. Again, there is a  $y \in \mathbb{N}$  such that  $M(i^{(k)}) \neq M(i^{(k)} \cdot (w_y, 1))$ . Now set  $i^{(k+1)} = i^{(k)} \cdot (w_y, 1) \cdot (w_{k+1}, 1)$ . By construction,  $i = \lim_{k \rightarrow \infty} i^{(k)}$  is an informant for  $\Sigma^*$  on which  $M$  changes its mind infinitely often. Hence,  $M$  fails to case-based learn  $\Sigma^* \in \mathcal{C}$  when successively fed its informant  $i$ , a contradiction, and the claim follows.

Select an informant  $i \in \text{info}(\Sigma^*)$ , and an  $x \in \mathbb{N}$  in accordance with the claim above. Since  $M$ , by assumption, case-based learns  $\Sigma^*$ ,  $M(i_x) = M(i_x \cdot (u, 1))$  for all  $u \in \Sigma^*$  immediately implies  $L(CB_x, \sigma) = \Sigma^*$  where  $(CB_x, \sigma) = M(i_x)$ . Next, we define a finite language  $\hat{L} \supseteq i_x^+$  and a co-finite language  $\tilde{L} \supseteq i_x^+$ , and show that  $M$  fails to case-based learn at least one

of them.

In order to construct  $\hat{L}$  and  $\tilde{L}$ , set  $L = i_x^+$  and consider  $M$ 's behavior when fed any informant  $i'$  for the finite language  $L$  with  $i'_x = i_x$ . Since  $M$ , in particular, learns  $L \in \mathcal{C}$  when fed  $i'$ , there has to be a  $y > x$  such that  $L(CB_y, \sigma) = L$ , where  $(CB_y, \sigma) = M(i'_y)$ . By Definition 5,  $CB_x \subseteq CB_y$ , and since  $L \neq \Sigma^* = L(CB_x, \sigma)$ , we know that  $CB_x \subset CB_y$ . Choose any string  $u \in \Sigma^* \setminus (i'_y{}^+ \cup i'_y{}^-)$ . Since  $u \notin L$  and  $u \in L(CB_x, \sigma)$ , there has to be a negative case in  $CB_y$  causing  $u$  not to belong to  $L(CB_y, \sigma) = L$ . Such a negative case, say  $(v, 0)$ , must meet  $\sigma(w, u) < \sigma(v, u)$  for all  $(w, 1) \in CB_y$ . Otherwise, we would obtain  $u \in L(CB_y, \sigma)$  which contradicts  $L(CB_y, \sigma) = L$ .

Now, we are ready to define  $\hat{L}$  and  $\tilde{L}$ . For that purpose, select two strings  $u$  and  $v$  that satisfy  $\sigma(w, u) < \sigma(v, u)$  for all  $(w, 1) \in CB_x$ , set  $\hat{L} = i_x^+ \cup \{u\}$ , and set  $\tilde{L} = \Sigma^* \setminus \{v\}$ . Clearly, the finite sequence  $i_x \cdot (u, 1) \cdot (v, 0)$  may serve as an initial segment of an informant for  $\hat{L}$  and  $\tilde{L}$ , respectively. Let  $(CB_{x+2}, \sigma) = M(i_x \cdot (u, 1) \cdot (v, 0))$ . By the choice of  $i_x$ , we know that  $(u, 1) \notin CB_{x+2}$  (cf. the above claim). We distinguish the following cases.

*Case 1.*  $(v, 0) \notin CB_{x+2}$ .

Let  $\tilde{i}$  denote any informant for the co-finite language  $\tilde{L}$  with  $\tilde{i}_{x+2} = i_x \cdot (u, 1) \cdot (v, 0)$  that contains the negative example  $(v, 0)$  exactly once. By definition, the CBL  $M$ , when fed  $\tilde{i}$ ,  $z > x + 2$ , can only add further positive cases to its actual case base  $CB_{x+2} = CB_x$ . Clearly, this does not help to accumulate a case base that allows to represent the target language  $\tilde{L} \subset \Sigma^* = L(CB_{x+2}, \sigma)$ .

*Case 2.*  $(v, 0) \in CB_{x+2}$ .

Now, let  $\hat{i}$  be any informant for the finite language  $\hat{L}$  with  $\hat{i}_{x+2} = i_x \cdot (u, 1) \cdot (v, 0)$  that contains only negative examples past point  $x + 2$ . By assumption,  $CB_x$  is a positive case base. Since  $\sigma(w, u) < \sigma(v, u)$  for all  $(w, 1) \in CB_x = CB_{x+2} \setminus \{(v, 0)\}$ , we obtain  $u \notin L(CB_{x+2}, \sigma)$ . In every subsequent step,  $M$  can only add negative cases to its actual case base  $CB_{x+2}$  which does not help to generate a case base  $CB$  satisfying  $u \in L(CB, \sigma)$ .

As we have seen, the CBL  $M$ , when fed  $\tilde{i}$  and  $\hat{i}$ , respectively, either fails to collect a case base that is suited to represent the co-finite language  $\tilde{L}$  or it cannot accumulate an appropriate case base for the finite language  $\hat{L}$ . Hence,  $M$  does not case-based learn  $\mathcal{C}$ , a contradiction, and the theorem follows.  $\square$

Since the class of all regular languages contains all finite and all co-finite languages we may conclude:

**Corollary 8.** *Let  $\sigma$  be any similarity measure. Then, there is no CBL  $M$  that case-based learns the class of all regular languages by collecting cases with respect to  $\mathcal{S} = \{\sigma\}$ .*

Consequently, powerful case-based learners that learn by collecting cases must be allowed to learn similarity measures, too. Next, we investigate the question to what extent the used similarity measure has to be modified within the learning process. As we shall see, it is still sufficient to tune a few parameters of an *a priori* fixed similarity measure, i.e., case-based learners will do that learn with respect to a class of similarity functions containing only slightly modified versions of an *a priori* fixed similarity measure.

The class of similarity measures we are going to apply is defined by adapting a weighting scheme for cases like that used in the weighted nearest neighbor algorithm [CS93]. This scheme is based on the idea that the cases in the case base are of different importance in order to represent a target concept. Intuitively, important cases are given larger weights making them more similar to elements belonging to the target concept. Since the used case bases are finite, this approach can be easily realized by explicitly assigning weights to those cases stored in the case base, only.

**Definition 6.** *Let  $\sigma$  be a similarity measure, let  $\alpha$  be a function of finite support, i.e.,  $\alpha : \Sigma^* \rightarrow \mathbb{Q}$  is a total function that meets  $\alpha(w) = 1$  for almost all  $w \in \Sigma^*$ . Then, we set  $\sigma_\alpha(u, v) = \alpha(u) \cdot \sigma(u, v)$  for all  $u, v \in \Sigma^*$ .*

*Furthermore, we call  $\sigma_\alpha$  a **weighted similarity measure based on  $\sigma$** . A similarity measure  $\delta$  belongs to the class  $\mathcal{S}[\sigma]$  provided there is a function  $\alpha$  of finite support such that  $\sigma_\alpha = \delta$ .*

Clearly, every function of finite support is total recursive. Furthermore, since the class of all functions of finite support is effectively enumerable, the class of similarity measures  $\mathcal{S}[\sigma]$  is effectively enumerable, too, no matter which similarity measure  $\sigma$  is chosen.

Based on Definition 5 we next give a precise notion of what does it mean to case-based learn a target class of languages by collecting cases and tuning only a few parameters of an *a priori* fixed similarity measure.

**Definition 7.** *Let  $\mathcal{C}$  be an indexable class, and let  $\sigma$  be a similarity measure. A **CBL  $M$  case-based learns  $\mathcal{C}$  by collecting cases and tuning parameters with respect to  $\sigma$**  iff  $M$  case-based learns  $\mathcal{C}$  by collecting cases with respect to  $\mathcal{S} = \mathcal{S}[\sigma]$ .*

*Finally,  $\mathcal{C}$  is said to be case-based learnable by collecting cases and tuning parameters iff there are a similarity measure  $\sigma$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  by collecting cases and tuning parameters with respect to  $\sigma$ .*

As our next result impressively shows, case-based learners that are allowed to collect cases and to tune parameters of a given similarity measure are of remarkable learning power provided that the used similarity measure is well-chosen. Interestingly, the additional learning power comes mainly from the following source: A case-based learner of the above type is able to simulate “forgetting,” i.e., a case that has been previously included in its case base can be paralyzed by simply setting the corresponding weight to 0. Furthermore, a case that has been once paralyzed can be, subsequently, reactivated by increasing its weight. Hence, paralyzing a certain case once included does not mean to strike off this case.

**Theorem 9.** *Let  $\mathcal{C}$  be an indexable class. Then, there are a similarity measure  $\sigma$  and a CBL  $M$  that case-based learns  $\mathcal{C}$  by collecting cases and tuning parameters with respect to  $\sigma$ .*

*Proof.* Let  $(w_k)_{k \in \mathbb{N}}$  be an effective repetition free enumeration of all strings in  $\Sigma^*$ . For every string  $s \in \Sigma^*$ , let  $\#(s) = k$  be the uniquely determined index  $k$  with  $w_k = s$ . Furthermore, choose any effective enumeration  $(L'_j)_{j \in \mathbb{N}}$  of  $\mathcal{C}$ , and define the enumeration  $\mathcal{L} = (L_j)_{j \in \mathbb{N}}$  of  $\hat{\mathcal{C}} = \mathcal{C} \cup \{\Sigma^*\}$  and the similarity measure  $\sigma$  as demonstrated in the proof of Theorem 3.

Applying Proposition 1 we may fix an IIM  $\hat{M}$  which identifies  $\mathcal{C}$  in the limit with respect to the hypothesis space  $\mathcal{L}$ . Similarly as in Theorem 5, the desired case-based learner  $M_c$  uses the IIM  $\hat{M}$  as a subroutine. So let  $L \in \mathcal{C}$ , and let  $i = ((s_x, d_x))_{x \in \mathbb{N}}$  be any given informant for  $L$ .

On input  $i_0$ , the CBL  $M_c$  stores the pair  $(s_0, d_0)$  in its case base, and outputs the pair  $(\{(s_0, d_0)\}, \sigma)$ . As long as no element  $(s, d)$  with  $d \neq d_0$  occurs,  $M_c$  simply repeats its first hypothesis  $(\{(s_0, d_0)\}, \sigma)$ . Otherwise, when  $i_y = (s_0, d_0), \dots, (s_y, d_y)$  contains for the first time a pair  $(s, d)$  with  $d \neq d_0$ ,  $M_c$  builds the case base  $CB_y = \{(s_0, d_0), (s_y, d_y)\}$ , and outputs the guess  $(CB_y, \sigma_y)$ , where  $\sigma_y$  equals  $\sigma$ . In every subsequent step, i.e., on every input  $i_{x+1}$ ,  $x \geq y$ ,  $M_c$  is defined as described in Figure 2. Note that, for the sake of readability, we specify only the weights for those strings  $u \in \Sigma^*$  with  $\alpha(u) \neq 1$ .

---

**Case-based learner  $M_c$ :**

**begin**

```

read  $(s_0, d_0), \dots, (s_x, d_x), (s_{x+1}, d_{x+1})$ ;
if  $i_{x+1}^+ \subseteq L(CB_x, \sigma_x)$  and  $i_{x+1}^- \cap L(CB_x, \sigma_x) = \emptyset$ 
  then  $CB_{x+1} := CB_x$ ;  $\sigma_{x+1} := \sigma_x$ 
  else  $j := \hat{M}(i_x)$ ;  $y := \min\{z \mid d_z \neq d_0\}$ ;
       $m := \max\{\#(s_z) \mid (s_z, d_z) \in CB_x\}$ ;
      if  $\#(s_{x+1}) = \langle j, \ell \rangle$  for some  $\ell > m$ 
        then if  $d_0 = d_{x+1}$ 
          then  $\alpha(s_z) := 0$ , if  $(s_z, d_z) \in CB_x \setminus \{(s_y, d_y)\}$ 
          else  $\alpha(s_z) := 0$ , if  $(s_z, d_z) \in CB_x \setminus \{(s_0, d_0)\}$ ;
           $CB_{x+1} := CB_x \cup \{(s_{x+1}, d_{x+1})\}$ ;  $\sigma_{x+1} := \sigma_x$ 
        else  $CB_{x+1} := CB_x$ ;  $\sigma_{x+1} := \sigma_x$ ;
output  $(CB_{x+1}, \sigma_{x+1})$ 

```

**end**

---

Figure 2: Case-based learning by collecting cases and tuning parameters.

By construction,  $M_c$  is a CBL that collects cases and tunes parameters with respect to  $\sigma$ . Note that  $\mathcal{S}[\sigma]$  contains total computable functions, only, and therefore  $M_c$ , when fed an initial segment  $i_{x+1}$ , can effectively test whether or not its last hypothesis  $(CB_x, \sigma_x)$  is consistent with all the data read so far, i.e.,  $i_{x+1}^+ \subseteq L(CB_x, \sigma_x)$  and  $i_{x+1}^- \cap L(CB_x, \sigma_x) = \emptyset$ . In order to verify  $M_c$ 's correctness we distinguish the following cases.

*Case 1.*  $L = \Sigma^*$ .

Hence,  $d_0 = 1$ . By construction,  $M_c$  outputs in every step the hypothesis  $(\{(s_0, 1)\}, \sigma)$ . Clearly,  $L(\{(s_0, 1)\}, \sigma) = \Sigma^*$  (cf. the demonstration of Theorem 3), and we are done.

*Case 2.*  $L \neq \Sigma^*$ .

Since  $L \neq \Sigma^*$ , there is a least  $y \in \mathbb{N}$  such  $d_0 \neq d_y$ , and thus  $CB_y$  contains a positive

and a negative case. By assumption, the IIM  $\hat{M}$  identifies  $L$  when fed  $i$ , and therefore there exists a least  $x > y$  such that  $j = \hat{M}(i_x) = \hat{M}(i_{x+r})$  for all  $r \in \mathbb{N}$ . We argue that  $M_c$  will change its mind at most once past this point.

Clearly, if  $L(CB_x, \sigma_x) = L$  we are done, since  $M_c$  changes its mind only in case that an inconsistency with the given input data has been detected.

Now, suppose that  $L(CB_x, \sigma_x) \neq L$ , and therefore  $M_c$  sometimes, say when fed  $i_n$ ,  $n \geq x$ , detects an inconsistency, i.e.,  $i_n^+ \not\subseteq L(CB_n, \sigma_n)$  or  $i_n^- \cap L(CB_n, \sigma_n) \neq \emptyset$ , where  $CB_n = CB_x$  and  $\sigma_n = \sigma_x$ . In every subsequent step,  $M_c$  either does not change its mind, i.e., it is repeating the guess  $(CB_x, \sigma_x)$ , or it adds a new case  $(s_{\hat{n}}, d_{\hat{n}})$ ,  $\hat{n} \geq n$ , to its current case base  $CB_x$  that meets  $\#(s_{\hat{n}}) = \langle j, \ell \rangle$  for some  $\ell > \max\{\#(s_z) \mid (s_z, d_z) \in CB_x\}$ . Since  $range(i) = \Sigma^*$ , the CBL  $M_c$  must eventually receive a suitable case  $(s_{\hat{n}}, d_{\hat{n}})$  and outputs, therefore, a different hypothesis  $(CB_{\hat{n}}, \sigma_{\hat{n}})$ , where  $CB_{\hat{n}} = CB_x \cup \{(s_{\hat{n}}, d_{\hat{n}})\}$ . Having a closer look at the definition of  $\sigma_{\hat{n}}$  one immediately sees that  $L(CB_{\hat{n}}, \sigma_{\hat{n}})$  either equals  $L(\{(s_{\hat{n}}, d_{\hat{n}}), (s_0, d_0)\}, \sigma)$  or it equals  $L(\{(s_{\hat{n}}, d_{\hat{n}}), (s_y, d_y)\}, \sigma)$ . Since  $\#(s_{\hat{n}}) = \langle j, \ell \rangle$  and, in particular,  $\#(s_{\hat{n}}) > \max\{\#(s_0), \#(s_y)\}$ , we may conclude that  $L(CB_{\hat{n}}, \sigma_{\hat{n}}) = L_j$  (cf. the observation stated within the proof of Theorem 3). Finally, since  $L_j = L$ , and since  $M_c$  is never modifying a correct hypothesis,  $M_c$  case-based learns  $L$  as required, and the theorem follows.  $\square$

## 6 Application: Case-Based Learning of CDLs

**Containment decision lists** (CDLs, for short) have been introduced in [SS92] to study the effect of noise on learning languages in the PAC learning framework. The class of languages acceptable by containment decision lists forms a proper subclass of the class of all regular languages and is based on the notion of **decision lists** [Riv87]. Within the present section, we study the case-based learnability of languages acceptable by CDLs within the different models of case-based learning introduced above.

The definition of the language represented by a containment decision list is based on the subword relation between strings. Given two strings  $u, v \in \Sigma^*$ ,  $u$  is said to be a **substring** of  $v$ , denoted by  $u \preceq v$ , iff there are  $w_1, w_2 \in \Sigma^*$  such that  $w_1 \cdot u \cdot w_2 = v$ .

**Definition 8.** Any non-empty list  $T = [(u_0, d_0), (u_1, d_1), \dots, (u_n, d_n), (u_{n+1}, d_{n+1})]$  is said to be a **containment decision list** over  $\Sigma$ , where  $u_i \in \Sigma^+$ ,  $i \leq n$ ,  $u_{n+1} = \varepsilon$ , and  $d_i \in \{0, 1\}$ ,  $i \leq n + 1$ . The language  $L(T)$  generated by the containment decision list  $T$  is defined as follows:  $L(T) = \{w \mid w \in \Sigma^*, d_k = 1 \text{ where } k = \min\{i \mid i \leq n + 1, u_i \preceq w\}\}$ .

Finally,  $\mathcal{C}_{CDL}$  denotes the class of languages that are acceptable by containment decision lists.

One may think of a containment decision list as an “**if – then – elseif – ... else –**” rule. For example, the containment decision list  $T = [(abba, 1), (aa, 0), (bbb, 1), (\varepsilon, 0)]$  may be pictured as in Figure 3, where  $\Sigma = \{a, b\}$ . There is no need to represent  $\varepsilon$  by a node. For example,  $aabbab \in L(T)$ , but  $aaabb \notin L(T)$ , since  $abba \not\preceq aaabb$  and  $aa \preceq aaabb$ .

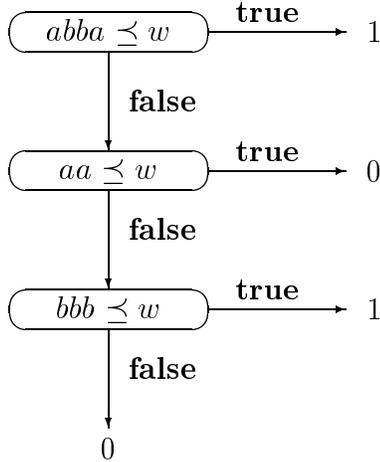


Figure 3: Diagram of the containment decision list  $[(abba, 1), (aa, 0), (bbb, 1), (\varepsilon, 0)]$

One easily verifies that every finite and every co-finite language, respectively, is acceptable by an appropriately chosen containment decision list. Hence, the following non-learnability result is an immediate consequence of Theorem 7.

**Corollary 10.** *Let  $\sigma$  be any similarity measure. Then, there is no CBL  $M$  that case-based learns  $\mathcal{C}_{CDL}$  by collecting cases with respect to  $\mathcal{S} = \{\sigma\}$ .*

On the other hand, we already know from Theorem 9 that there are proper superclasses of the language class  $\mathcal{C}_{CDL}$ , the class of all regular languages, for instance, that are case-based learnable by collecting cases and tuning parameters. Hence, we may conclude:

**Corollary 11.** *The class  $\mathcal{C}_{CDL}$  is cased-based learnable by collecting cases and tuning parameters.*

However, the similarity measure used within the demonstration of Theorem 9 is quite general, since it allows to uniformly represent every indexable class. Therefore, the question arose whether other more natural similarity measures guarantee the case-based learnability of the class  $\mathcal{C}_{CDL}$  as well.

Next, we define a simple and seemingly natural similarity measure  $\sigma_{\preceq}$  that reflects the characteristic features of languages acceptable by CDLs quite well. For two strings  $u, w \in \Sigma^*$  we define:

$$\sigma_{\preceq}(u, w) = \begin{cases} 1, & \text{if } u \preceq w \\ 0, & \text{otherwise.} \end{cases}$$

Given any CDL  $T$  the strings used within the nodes of  $T$  form a case-base that is suited to case-based represent  $L(T)$  with respect to a well-chosen weighted similarity measure based on  $\sigma_{\preceq}$ .

**Theorem 12.** *Let  $T$  be any containment decision list. Then, there is a similarity measure  $\sigma \in \mathcal{S}[\sigma_{\preceq}]$  such that  $L(T)$  is case-based representable with respect to  $\sigma$ .*

*Proof.* Let  $T = [(u_0, d_0), \dots, (u_n, d_n), (\varepsilon, d_{n+1})]$ . We set  $CB^+ = \{u_r \mid r \leq n, d_r = 1\}$  and  $CB^- = \{u_r \mid r \leq n, d_r = 0\}$ . In order to define the desired similarity measure  $\sigma \in \mathcal{S}[\sigma_{\preceq}]$ , we choose the corresponding weights as follows. For all  $r \leq n + 1$ , we set  $\alpha_{u_r} = 1/r + 1$ . Furthermore, let  $\alpha(v) = 1$  for all  $v \notin CB^+ \cup CB^-$ . Finally, it is not hard to verify that  $L(CB, \sigma) = L(T)$ , where  $CB = CB^+ \times \{1\} \cup CB^- \times \{0\}$ . We omit further details.  $\square$

Surprisingly enough, the seemingly natural similarity measure  $\sigma_{\preceq}$  does not allow to case-based learn the class of all language acceptable CDLs by collecting cases and tuning parameters.

**Theorem 13.** *There is no CBL  $M$  that case-based learns  $\mathcal{C}_{CDL}$  by collecting cases and tuning parameters with respect to  $\sigma_{\preceq}$ .*

*Proof.* As above, let  $\Sigma = \{a, b\}$  be the underlying alphabet. First, we define a particular class of containment decision lists over  $\Sigma$ . We set  $T_0 = [(bab, 1), (aba, 0), (\varepsilon, 1)]$  and  $T_k = [(bab, 1), (aba, 0), (ba^{k+1}b, 1), (ba^{k+1}, 0), (a^{k+1}b, 0), (\varepsilon, 1)]$  for all  $k \in \mathbb{N}^+$ .

By definition we may easily conclude:

**Observation 1.** Let  $k \in \mathbb{N}^+$ , and let  $w \in \Sigma^*$  with  $|w| \leq k + 2$ . Then,  $w \in L(T_0)$  iff  $w \in L(T_k)$ .

The class of languages acceptable by these particularly designed containment decision lists has the interesting property that some of the nodes listed have necessarily to occur in any case base used to specify the corresponding language via the given similarity measures in  $\mathcal{S}[\sigma_{\preceq}]$ . For the subsequent proof, it is sufficient to consider this property with respect to positive cases, only. More formally:

**Observation 2.** Let  $k \in \mathbb{N}^+$ , let  $\sigma \in \mathcal{S}[\sigma_{\preceq}]$ , and let  $CB^+ \subseteq L(T_k)$  and  $CB^- \subseteq \overline{L(T_k)}$ . Then,  $L(CB, \sigma) = L(T_k)$  implies  $ba^{k+1}b \in CB^+$ , where  $CB = CB^+ \times \{1\} \cup CB^- \times \{0\}$ .

In order to verify the latter observation, let  $k \in \mathbb{N}^+$ , let  $\sigma \in \mathcal{S}[\sigma_{\preceq}]$ , and let  $CB^+ \subseteq L(T_k)$  and  $CB^- \subseteq \overline{L(T_k)}$ . Suppose to the contrary that  $L(CB, \sigma) = L(T_k)$ , but  $ba^{k+1}b \notin CB^+$ . Now, consider the string  $w = ba^{k+1}b$ . Clearly,  $w \in L(T_k)$ . By assumption,  $L(CB, \sigma) = L(T_k)$  and, therefore, there must be some string  $u \in CB^+$  that satisfies both  $\sigma(u, w) > 0$  and  $\sigma(u, w) > \sigma(v', w)$  for all  $v' \in CB^-$ . By the definitions of both  $\sigma_{\preceq}$  and  $\mathcal{S}[\sigma_{\preceq}]$ , we may conclude that  $u \preceq ba^{k+1}b$ . Now since  $ba^{k+1}b \notin CB^+$ , we obtain  $u \neq ba^{k+1}b$  and, thus,  $u \prec ba^{k+1}b$ . There could be two subcases: ( $\alpha$ )  $u \preceq ba^{k+1}$  and ( $\beta$ )  $u \preceq a^{k+1}b$ . Without loss of generality, assume that ( $\alpha$ ) happens and set  $w' = ba^{k+1}$ . Clearly,  $w' \notin L(T_k)$ . Since by assumption  $L(CB, \sigma) = L(T_k)$ , there has to be a string  $v \in CB^-$  that, in particular, meets  $\sigma(v, w') \geq \sigma(u, w')$ . Hence,  $\sigma(v, w') > 0$  follows and, thus, we obtain  $v \preceq w'$  as before. Moreover,  $w' \preceq w$  implies  $v \preceq w$ . Putting this all together we may conclude that  $\sigma(v, w) = \sigma(v, w') \geq \sigma(u, w') = \sigma(u, w)$  which contradicts  $\sigma(u, w) > \sigma(v', w)$  for all  $v' \in CB^-$ . This contradiction finishes the proof of Observation 2.

The proof proceeds by *reductio ad absurdum*. Assume any case-based learner  $M$  that infers every language acceptable by a CDL  $T_k$ ,  $k \in \mathbb{N}$ , as required. Therefore, the CBL  $M$ ,

in particular, case-based learns  $L(T_0)$  by collecting cases and tuning parameters with respect to  $\sigma_{\leq}$ . It suffices to show that there is a  $k \in \mathbb{N}^+$  and an informant  $\hat{i}$  for  $L(T_k)$  such that  $M$  fails to learn  $L(T_k)$  when successively fed  $\hat{i}$ . This can be seen as follows.

Similarly as in the proof of Theorem 7, one easily verifies that there exists a locking sequence for  $L(T_0)$ . Thus, there exists an  $i \in \text{info}(L(T_0))$  and an  $x \in \mathbb{N}$  such that  $M(i_x) = M(i_x \cdot (u, 1))$  for all  $u \in L(T_0)$ . Given any locking sequence  $i_x$  for  $L(T_0)$ , choose any  $k \in \mathbb{N}^+$  exceeding the maximal word length in  $i_x$ . By Observation 1 we know that  $i_x^+ \subseteq L(T_k)$  and  $i_x^- \cap L(T_k) = \emptyset$ . Finally, select any informant  $\hat{i} = ((s_n, d_n))_{n \in \mathbb{N}}$  for  $L(T_k)$  that meets  $\hat{i}_x = i_x$ ,  $s_{x+1} = ba^{k+1}b$ , and  $s_y \neq ba^{k+1}b$  for all  $y > x + 1$ . Since  $\hat{i}_x$  is a locking sequence for  $L(T_0)$ , and since, by definition of  $T_0$ ,  $ba^{k+1}b \in L(T_0)$ , the case-based learner  $M$  does not add the positive case  $(ba^{k+1}b, 1)$  to its actual case base  $CB_x$ , i.e.,  $(ba^{k+1}b, 1) \notin CB_{x+1}$ , and therefore  $CB_{x+1} = CB_x$ . Since the positive case  $(ba^{k+1}b, 1)$  appears exactly once, and since  $M$  is supposed to case-based learn  $L(T_k)$  by collecting cases and tuning parameters, the CBL  $M$ , if ever, converges on  $\hat{i}$  to a pair  $(CB, \sigma)$  with  $(ba^{k+1}b, 1) \notin CB$  and  $\sigma \in \mathcal{S}[\sigma_{\leq}]$ . This completes the proof, since according to the contraposition of Observation 2, we know that  $L(CB, \sigma)$  does not equal  $L(T_k)$ .  $\square$

## 7 Conclusions

In the present paper we have been investigating the possibilities as well as the limitations of case-based learning of indexable classes. The characteristic feature of a case-based learner is the way it encodes its hypotheses. Each hypothesis of a case-based learner consists of a case base and a similarity measure. Such a pair is called a case-based representation. First, we have been studying which indexable classes are case-based representable. If the similarity measure is fixed for the class under consideration, case-based representability using positive cases only is quite limited. In contrast, every indexable class is case-based representable with respect to a fixed similarity measure, if positive and negative cases are admissible. Then, moreover, case bases that contain at most two cases are sufficient. These results heavily depend on the existence of infinitely many cases. For a study of case-based representability in finite domains the reader is referred to [GL96].

The case-based reasoning paradigm adopts the view that an AI system is dynamically changing during its life-cycle. A system's reaction in response to environmental changes may be reasonably interpreted as learning, in many cases. A case-based learning problem is the twofold task to select from the positive and negative cases received during a system's application appropriate ones and to choose a similarity measure from a pre-given class in order to produce a case-based representation that describes the target phenomenon. Since every indexable class can be case-based learned easily if the class of similarity measures contains an appropriate measure for each language, we analyzed two restrictions on the general model. On the one hand, the case-based learner was forced to use the same similarity measures for all languages of the class. On the other hand, we studied case-based learners that are required to collect cases only, i.e., case-based learners that can only decide whether or not

to store up the last case presented in the case base.

As it turns out all indexable classes remain learnable, if the case-based learner has to obey either of these restrictions. But there are indexable classes that are not case-based learnable, if both to collect cases only and to use a single similarity measure is demanded. Remarkably, the situation is not as worse as one might expect. It suffices to weight the importance of cases in the case base comparable to the weighted nearest neighbor algorithm proposed in [CS93] to gain case-based learnability even under these restrictions.

To illustrate the dependency between case-based representability and the choice of an appropriate similarity measure, we have been investigating an area which seems particularly tailored to case-based reasoning – learning of containment decision lists. Whereas a certain seemingly natural similarity measure does not admit case-based learnability of all containment decision lists by collecting cases and tuning parameters, certain non-standard concepts do. This nicely illustrates the limitations of common-sense reasoning in automating learning processes. The dependency on the choice of the similarity measure is under further investigations by experiments (cf. [BDF96], for a guided tour through our system prototype). The reader may consult [JL95] for a couple of similar results trading naturalness versus representability and learnability, respectively.

## Acknowledgement

The work of the first three authors has been partially supported by the DEUTSCHE FORSCHUNGSGEMEINSCHAFT (DFG) within the project IND-CBL.

Our students Udo Burghardt, Volker Dötsch, and Stephan Frind at Leipzig University of Technology have developed and implemented an experimental system TIC for investigations and comparisons of several classical inductive and case-based learning algorithms. Among others, the system provides a very helpful environment for studying phenomena of learning containment decision lists.

It is worth to be mentioned that the origin of the present investigations has been at the FUJITSU Laboratories in Numazu, Japan, where the last three authors had several opportunities to collaborate.

We gratefully acknowledge the reports of two anonymous referees. The critical hints and the constructive suggestions helped very much to improve the presentation of our ideas and results.

## References

- [AKA91] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

- [Ang80] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.
- [AS83] Dana Angluin and Carl H. Smith. Inductive inference: Theory and methods. *ACM Computing Surveys*, 15:237–269, 1983.
- [BB75] Leonore Blum and Manuel Blum. Toward a mathematical theory of inductive inference. *Information and Control*, 28:122–155, 1975.
- [BB+93] Shirin Bakhtari, Katy Börner, Brigitte Bartsch-Spörl, Carl-Helmut Coulon, Dietmar Janetzko, Markus Knauff, Ludger Hovestadt, and Christoph Schlieder. EWCBR 93: Contributions of FABEL. *FABEL-Report 17*, Gesellschaft für Mathematik und Datenverarbeitung, 1993.
- [BDF96] Udo Burghardt, Volker Dötsch, and Stephan Frind. TIC - ein Testrahmen für IND-CBL. *CALG 01/96*, HTWK Leipzig, Fachbereich IMN, Forschungsgruppe Algorithmisches Lernen, 1996.
- [CG94] Carl-Helmut Coulon, Friedrich Gebhardt. Evaluation of retrieval methods in case-based design. *FABEL-Report 24*, Gesellschaft für Mathematik und Datenverarbeitung, 1994.
- [CS93] Scott Cost and Steven Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.
- [GL96] Christoph Globig and Steffen Lange. Case-based representability of classes of boolean functions. In *Proceedings of 12th European Conference on Artificial Intelligence (ECAI 96)*. John Wiley & Sons, 1996, to appear.
- [Gol67] E Mark Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [Jan92] Klaus P. Jantke. Case-based learning in inductive inference. In *Proceedings of 5th Workshop on Computational Learning Theory (COLT'92)*, pages 218–223. ACM Press, 1992.
- [Jan93] Klaus P. Jantke. Types of incremental learning. In *Working Notes, AAAI Spring Symposium on Training Issues in Incremental Learning*, pages 26–32. Stanford University, 1993.
- [JL93] Klaus P. Jantke and Steffen Lange. Case-based representation and learning of pattern languages. In *Proceedings of 4th Workshop on Algorithmic Learning Theory (ALT'93)*, Lecture Notes in Artificial Intelligence 744, pages 87–100. Springer-Verlag, 1993.

- [JL95] Klaus P. Jantke and Steffen Lange. Case-based representation and learning of pattern languages. *Theoretical Computer Science*, 137:25–51, 1995.
- [Kol92] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6:3–34, 1992.
- [Kol93] Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
- [RS89] Christopher K. Riesbeck and Roger C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum Assoc., 1989.
- [Riv87] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.
- [SS92] Yasubumi Sakakibara and Rani Siromoney. A noise model on learning sets of strings. In *Proceedings of 5th Workshop on Computational Learning Theory (COLT'92)*, pages 295–302. ACM Press, 1992.
- [ZL95] Thomas Zeugmann and Steffen Lange. A guided tour across the boundaries of learning recursive languages. In Klaus P. Jantke and Steffen Lange (eds.) *Algorithmic Learning for Knowledge-Based Systems*, pages 190–258. Springer-Verlag, 1995.