Identification in the limit of systematic noisy languages

Frédéric Tantini, Colin de la Higuera and Jean-Christophe Janodet

Eurise, Université de Saint-Etienne, 23 rue du Docteur Paul Michelon,
42023 Saint-Etienne
{frederic.tantini,cdlh,janodet}@univ-st-etienne.fr

Abstract. To study the problem of learning from noisy data, the common approach is to use a statistical model of noise. The influence of the noise is then considered according to pragmatic or statistical criteria, by using a paradigm taking into account a distribution of the data. In this article, we study the noise as a nonstatistical phenomenon, by defining the concept of systematic noise. We establish various ways of learning (in the limit) from noisy data. The first is based on a technique of reduction between problems and consists in learning from the data which one knows noisy, then in denoising the learned function. The second consists in denoising on the fly the training examples, thus to identify *in the limit* good examples, and then to learn from noncorrupted data. We give in both cases sufficient conditions so that learning is possible and we show through various examples (coming in particular from the field of the grammatical inference) that our techniques are complementary.

Keywords: Identification in the limit, languages, noise, pretopology.

Submission to ICGI'06

1 Introduction

Grammatical inference [1, 2] is a field that provides a lot of algorithms to learn from sequential or structured data: words, trees,... Among the advantages of these techniques, we can underline the comprehensibility of the learned models, solid theories which allow in particular to avoid working with nonexplicit bias, the power of the functions which define the concepts (automata and grammars), the fact that the training data can be analyzed in their globality and not by taking into account only pieces of information, *etc.* But these qualities have a counterpart: they do not resist (or hardly) to noise [3, 4].

Noise appears in the data for several reasons. It can be due to the fact that the bias is inadapted: if we try to learn a regular language from data that comes from a context-free language, we can expect problems. It can also be due to poor experimental conditions or to the fact that cleaning the data is either too difficult or too expensive. This can occur in voice recognition, or when we wish to learn from manually-built HTML files .

The management of noise is a crucial and recurring problem in machine learning in general. Concerning grammatical inference, we can quote the following lines of research. Very theoretical work was undertaken, either within the framework of inductive inference [5, 6] and following the track of old results [7], or in that of approximate learning [8, 9]. Other works tried to use well-founded ideas in existing algorithms to make them more robust to noise [10, 11]. In addition, nondeterministic automata are probably more resistant to the noise than the deterministic ones [12]. More pragmatic works were undertaken to use techniques of grammatical inference on naturally noisy time series [13]. We can also note studies on approximated learning of languages, which are based on the *rough sets* theory and brings algorithms intrinsically more resistant to noisy data [14]. The paradigm of learning by analogy was also the subject of a study under the angle of its resistance to noisy data [15]. Lastly, a traditional approach is that of learning stochastic automata. This approach aims at avoiding the problem by imposing a different bias: the data come from a distribution, itself represented by a stochastic automaton [16]. The question is then not that of learning a language but a distribution. Results in this direction are both theoretical [17] and algorithmic [18].

Let us note that in most of the theoretical approachess, the treatment of noise is statistical. In this work, we explore the case of a *systematic* noise based on the edit distance; we study the properties of this kind of noise in the context of the identification in the limit [19, 20].

In this setting we propose various ways of learning (in the limit) from noisy data. The first one is based on a technique of reduction between problems and consists in learning from the data, knowing that it is noisy, then in denoising the learned function. The second one consists in denoising on the fly the learning examples, thus in identifying *in the limit* good examples, and in learning from noncorrupted examples. In this second approach, we show that it is possible (and sometimes recommended) to add additional noise to boost the training.

We give for these two outlines sufficient conditions for the learning and we show through various examples (and in particular examples coming from the field of Grammatical Inference) that the techniques are complementary. The definitions we tailor are general, and we use them within the framework of the systematically noisy texts, but we believe they might be used in a broader way.

2 Preliminaries

An *alphabet* Σ is a non-empty finite set of symbols called *letters*. A *word* w is a finite sequence $w = a_1 a_2 \dots a_n$ of letters. Let $|w|$ denote the length of w. In the following, letters will be indicated by $a, b, c, \dots$, words by $u, v, \dots, z$ and the empty word by λ. Let Σ^* be the set of al finite words over alphabet Σ. We call language any subset $L \subseteq \Sigma^*$.

The identification in the limit paradigm has been introduced by Gold [19]. We give it here in the formalism of [21] which allows to study reductions between identification problems (section 4). Let $\mathcal{L}$ be a class of languages and $\mathcal{R}(\mathcal{L})$ a class of representations for $\mathcal{L}$ (*e.g.* the class of regular languages and that of deterministic automata). Let $\mathbb{L}_{\mathcal{L}} : \mathcal{R}(\mathcal{L}) \to \mathcal{L}$ be the function that for every representation returns the corresponding language. This function is surjective: every language can be represented. But it does not have to be injective. Indeed, two different functions can represent the same language. We suppose that the following word problem is decidable: "given $w \in \Sigma^*$ and $G \in \mathcal{R}(\mathcal{L})$, $w \in \mathbb{L}_{\mathcal{L}}(G)$?".

Definition 1 (Presentation). *A presentation of $L \in \mathcal{L}$ is a function $f : \mathbb{N} \to X$ where X is a set. Let $\boldsymbol{Pres}(\mathcal{L})$ be the set of all presentations. Since a presentation denotes a language of $\mathcal{L}$, there exists a function yield : $\boldsymbol{Pres}(\mathcal{L}) \to \mathcal{L}$. I.e., if $L = yield(f)$ then f is a presentation of L, or $f \in \boldsymbol{Pres}(L)$. Let f_n denote the set $\{f(j) : j < n\}$.*

With this definition, the notion of presentations is very large: they are sequences of information of any type that inform us on the language. Indeed, X can be Σ^* in the case of positive examples only. If in addition, $yield(f) = f(\mathbb{N})$, then such presentations are called *texts*. In the case of an *informant*, which is a presentation with both negative and positive examples, $X = \Sigma^* \times \{0, 1\}$.

If two languages share one presentationt,then they cannot be distinguished, so $\mathcal{L}$ will not be learnable from $\mathbf{Pres}(\mathcal{L})$. Therefore, we will suppose that if two presentations f and g are such that $f(\mathbb{N}) = g(\mathbb{N})$, then $yield(f) = yield(g)$.

A *learning algorithm* **alg** is a program taking as input the first n elements of a presentation and returning a representation:

$$\mathbf{alg} : \bigcup_{f \in \mathbf{Pres}(\mathcal{L}), i \in \mathbb{N}} \{f_i\} \to \mathcal{R}(\mathcal{L})$$

The next definition is adapted from [20]:

Definition 2. *We say that $\mathcal{L}$ is learnable in the limit from $\boldsymbol{Pres}(\mathcal{L})$ in terms of $\mathcal{R}(\mathcal{L})$ if there exists a learning algorithm $\boldsymbol{alg}$ such that for any $L \in \mathcal{L}$ and for any presentation $f \in \boldsymbol{Pres}(L)$, there exists a rank n such that for all $m \geq n$, $\mathbb{L}_{\mathcal{L}}(\boldsymbol{alg}(f_m)) = L$.*

Usual classes of languages (defined by automata, grammars, ...) are not appropriate in the case of noise. The essential problem is that in a quasi systematic way, the modification of a symbol in a word swaps it from the language to its complementary. To use an image coming from a field in which the noise was better analysed, it is like if, by drawing on a screen the words of a language, no shape was perceptible: all the languages would look like uniform grey. We thus introduce distance and simple topological objects, the balls, which do not present this problem.

The edit distance between two words was defined by Levenshtein in 1965 [22]. It consists in counting the minimal number of symbol operations needed to rewrite the former into the latter, where the operations are the insertion, the substitution and the deletion. More formally, let w and w' be two words in Σ^*, we rewrite w into w' in one step if one of the following condition is true:

- $w = uav, w' = uv$ and $u, v \in \Sigma^*, a \in \Sigma$ (deletion),
- $w = uv, w' = uav$ and $u, v \in \Sigma^*, a \in \Sigma$ (insertion),
- $w = uav, w' = ubv$ and $u, v \in \Sigma^*, a, b \in \Sigma$ (substitution).

We consider the reflexive and transitive closure of this relation and we note $w \xrightarrow{k} w'$ iff w can be rewritten into w' by means of k operations. Then the Levenshtein distance between w and w', noted $d_{edit}(w, w')$, is the smallest k such that $w \xrightarrow{k} w'$. For instance, $d_{edit}(abaa, aab) = 2$ since $abaa \rightarrow aaa \rightarrow aab$.

Notice that the edit distance between two words is computed by dynamic programming [23]. Moreover several variants have been studied and the distance has been adapted to the case of circular words and trees. The weight of the edit operations can also differ from 1. We have chosen in this work to study only the standard case.

Definition 3 (Balls). *The ball of centre $u \in \Sigma^*$ and radius $r \in \mathbb{N}$ is defined by $B_r(u) = \{w \in \Sigma^* : d(w, u) \leq r\}$. A representation of the ball $B_r(u)$ will then be the couple (u, r). Let $\mathcal{B}_\Sigma$ denote the set of all the balls: $\mathcal{B}_\Sigma = \{B_k(u) : K \in \mathbb{N}, U \in \Sigma^*\}$.*

Note that if the alphabet Σ contains only one letter, the same ball can be represented in several ways ($B_2(a) = B_3(\lambda)$), but this characteristic is not a problem: many classes of representations have this property (automata, grammars).

3 Identification in the limit from noisy data

In this paper, we propose a model of noise that we call *systematic*: noise will be added to a data in all the possible ways up to a certain distance. This idea can be illustrated by considering spots of painting on a paper sheet: putting an object on the sheet makes the points become blurs.

Definition 4 (Noise of a language). *Let L be a language on Σ^*. The* k-noise *of L is $N_k(L) = \{w : \exists x \in L, d(x, w) \leq k\}$.*

Let us first notice that once noise is added, if two languages of the class are not distinguishable one from the other, then the class itself is not resistant to systematic noise. In particular, this is the case of the class of rational languages, and in a broader way for any class defined by rewriting systems. The possibility to represent in these classes *parity functions* is susceptible to convince us of the low resistance to the noise of these languages [4]. This justifies our interest in classes of languages defined differently than through grammars.

It is reasonable to study systematic noise in the paradigm of the identification in the limit, due to the absence of distribution on the data. To this purpose, we introduce the following new notion of presentation:

Definition 5 (Noisy presentation). *A noisy presentation is a presentation $f : \mathbb{N} \to X$ for which there exists an (unknown) function* isnoise $: X \to \{0, 1\}$ *that is able to distinguish noisy elements and pure ones.*

This definition allows to model a variety of situations, for instance:

Definition 6 (k-noisy presentation). *Let L be a language. A k-noisy presentation of L is a presentation of $N_k(L)$. The function* isnoise *is then equal to 0 on the elements of L and to 1 on those of $N_k(L) \setminus L$.*

We now tackle the problem of learning in presence of noisy data. Two solutions seem relevant as shown by the following diagram:

$$\begin{array}{ccc} \mathbf{Pres}(\mathcal{L}) & \longrightarrow & \mathcal{L}\text{'} \\ \downarrow & & \downarrow \\ \overline{\mathbf{Pres}}(\mathcal{L}) & \longrightarrow & \mathcal{L} \end{array}$$

In this diagram the problem is to learn a language $\mathcal{L}$ from a noisy presentation of $\mathbf{Pres}(\mathcal{L})$. First, we can try to learn instead a language from another class which would incorporate the noise (the class $\mathcal{L}$') and then try to deduce the original language. On the other hand, we can try to denoise the data in order to obtain a nonnoisy presentation in $\overline{\mathbf{Pres}}(\mathcal{L})$ and then learn from this one. In this second strategy, it is thus the function isnoise that we want to identify.

4 Reduction

A technique implemented in many fields of computer science and mathematics is that of *reductions*. They make are used to obtain negative results (such problem is at least as difficult as such other, known as being too hard) but also to use algorithms that are valid in a case for another case. Here, we consider the latter technique: following the arguments of [21], we show that the balls are identifiable from noisy data. Beyond this result, we think that the reductions are an effective way to learn from noisy data.

We recall that a situation of identification is defined by the class of languages, that of the representations and the type of allowed presentations. Let $\mathcal{L}$ and $\mathcal{L}$' be the two classes of languages represented respectively by $R(\mathcal{L})$ and $R(\mathcal{L}')$. We denote by $\mathbb{L}_{\mathcal{L}}$ (*resp.* $\mathbb{L}_{\mathcal{L}'}$) the surjective mapping $R(\mathcal{L}) \to \mathcal{L}$ (*resp.* $\mathbb{L}_{\mathcal{L}'} : R(\mathcal{L}') \to \mathcal{L}'$).

Given a surjective mapping $\phi : \mathcal{L} \to \mathcal{L}'$, we denote by ψ a surjective mapping $R(\mathcal{L}) \to R(\mathcal{L}')$ for which the diagram commutes ($\phi \circ \mathbb{L}_{\mathcal{L}} = \mathbb{L}_{\mathcal{L}'} \circ \psi$):

$$\begin{array}{ccc} R(\mathcal{L}) & \xrightarrow{\psi} & R(\mathcal{L}') \\ {\scriptstyle \mathbb{L}_{\mathcal{L}}}\downarrow & & \downarrow{\scriptstyle \mathbb{L}_{\mathcal{L}'}} \\ \mathcal{L} & \xrightarrow{\phi} & \mathcal{L}' \end{array}$$

Given a surjective mapping $\phi : \mathcal{L} \to \mathcal{L}'$, we denote ξ a surjective mapping $\mathbf{Pres}(\mathcal{L}) \to \mathbf{Pres}(\mathcal{L}')$ for which the following diagram commutes ($\phi \circ yield_{\mathcal{L}} = yield_{\mathcal{L}'} \circ \xi$):

$$\begin{array}{ccc} \mathcal{L} & \xrightarrow{\phi} & \mathcal{L}' \\ {\scriptstyle yield_{\mathcal{L}}}\uparrow & & \uparrow{\scriptstyle yield_{\mathcal{L}'}} \\ \mathbf{Pres}(\mathcal{L}) & \xrightarrow{\xi} & \mathbf{Pres}(\mathcal{L}') \end{array}$$

As a presentation may not be a computable function, describing the computation aspects of function ξ is as follows:

Definition 7. *Let $\mathcal{L}$ be a class of languages represented in $R(\mathcal{L})$ with presentations in $\boldsymbol{Pres}(\mathcal{L}) : \mathbb{N} \to X$ and let $\mathcal{L}$' be a class of languages represented in $R(\mathcal{L}')$ with presentations in $\boldsymbol{Pres}(\mathcal{L}') : \mathbb{N} \to Y$. A reduction between presentations $\xi : \boldsymbol{Pres}(\mathcal{L}) \to \boldsymbol{Pres}(\mathcal{L}')$ such that $\xi(\boldsymbol{f}) = \boldsymbol{g}$ is* computable *if and only if there exists a computable function $\overline{\xi} : X \to 2^Y$ such that $\bigcup_{i \in \mathbb{N}} \overline{\xi}(\boldsymbol{f}(i)) = \boldsymbol{g}(\mathbb{N})$.*

$\overline{\xi}$ is the description of the function ξ in all its points. We suppose here that $\forall i \in \mathbb{N}$, $\overline{\xi}(\boldsymbol{f}(i))$ is a finite set.

By combining the two previous diagrams we obtain:

$$\begin{array}{ccc} R(\mathcal{L}) & \xrightarrow{\psi} & R(\mathcal{L}') \\ {\scriptstyle \mathbb{L}_{\mathcal{L}}}\downarrow & & \downarrow{\scriptstyle \mathbb{L}_{\mathcal{L}'}} \\ \mathcal{L} & \xrightarrow{\phi} & \mathcal{L}' \\ {\scriptstyle yield_{\mathcal{L}}}\uparrow & & \uparrow{\scriptstyle yield_{\mathcal{L}'}} \\ \mathbf{Pres}(\mathcal{L}) & \xrightarrow{\xi,\overline{\xi}} & \mathbf{Pres}(\mathcal{L}') \end{array}$$

Theorem 1. *If (i) $\mathcal{L}$' is learnable in terms of $R(\mathcal{L}')$ from $\boldsymbol{Pres}(\mathcal{L}')$, (ii) there exists a computable function $\chi : R(\mathcal{L}') \to R(\mathcal{L})$ and a computable function*

$\psi : R(\mathcal{L}) \to R(\mathcal{L}')$ such that $\psi \circ \chi = Id$ and (iii) ξ is a computable reduction, then $\mathcal{L}$ is learnable in terms of $R(\mathcal{L})$ from $\boldsymbol{Pres}(\mathcal{L})$.

$$\begin{array}{ccc} R(\mathcal{L}) & \xleftarrow{\chi} & R(\mathcal{L}') \\ {\scriptstyle \mathbb{L}_{\mathcal{L}}}\downarrow & & \downarrow{\scriptstyle \mathbb{L}_{\mathcal{L}'}} \\ \mathcal{L} & \xrightarrow{\phi} & \mathcal{L}' \\ {\scriptstyle yield_{\mathcal{L}}}\uparrow & & \uparrow{\scriptstyle yield_{\mathcal{L}'}} \\ \boldsymbol{Pres}(\mathcal{L}) & \xrightarrow{\xi,\overline{\xi}} & \boldsymbol{Pres}(\mathcal{L}') \end{array}$$

Proof. Let **alg2** be a learning algorithm that identifies $\mathcal{L}'$. Consider algorithm **alg1** below, that takes a presentation $\boldsymbol{f}$ by its n first items $(\boldsymbol{f}_n)$ and then executes:

$\boldsymbol{g}_m \longleftarrow \overline{\xi}(\boldsymbol{f}_n)$
$G_{\mathcal{L}'} \longleftarrow \mathbf{alg2}(\boldsymbol{g}_m)$
$G_{\mathcal{L}} \longleftarrow \chi(G_{\mathcal{L}'})$
return $G_{\mathcal{L}}$

$G_{\mathcal{L}}$ and $G_{\mathcal{L}'}$ are grammars of $R(\mathcal{L})$ and $R(\mathcal{L}')$. As ξ is computable, $\boldsymbol{g}_m$ can effectively be built.

As a consequence of Theo. 1, we can prove known results like the identification of even linear grammars [24], by reduction from deterministic finite automata. In the context of noisy data, we get:

Theorem 2. *$\mathcal{B}_\Sigma$ is learnable in the limit from k-noisy text.*

We first establish that the k-noise of a ball is a ball:

Lemma 1. $N_k(B_{k'}(u)) = B_{k+k'}(u)$

Proof. ($\subseteq$) Let $x \in N_k(B_{k'}(u))$. Then $\exists y \in B_{k'}(u) : d(y,x) \leq k$, so $d(u,y) \leq k' \wedge d(y,x) \leq k$, therefore $d(u,x) \leq k' + k$.
($\supseteq$) Let $x \in B_{k+k'}(u)$ so $d(u,x) \leq k + k'$. Let $d(u,x) > k'$. The fact that $k' < d(u,x) \leq k + k'$ means that u can be changed in x by the mean of $k + k'$ operations of edition. Let y be the word obtained after the first k' operations. Then $d(u,y) = k'$ and $d(y,x) \leq k$; thus $y \in B_{k'}(u)$ and $x \in N_k(B_{k'}(u))$.

We also get the following result:

Lemma 2. *$\mathcal{B}_\Sigma$ is identifiable in the limit from text.*

Proof. By saturation, when all the points have appeared, the ball can be computed. If only some points are given, the problem is NP-hard [25], but, with all the points, it is easy: let B_{max} be the set of the longuest words. The centre u is the only word such that $a^k u$ and $b^k u$ are in B_{max}, where k is the greatest integer such that a^k and b^k are left factors of B_{max}. The ball is then $B_k(u)$.

From Lemmas 1 and 2 we deduce:

Proof (of Theo. 2). Taking χ=*if the radius of the ball is at least* k*, then deduct* k *from the radius, if not identity*, we obtain the following diagram:

$$\begin{array}{ccc} \mathcal{B}_\Sigma & \xleftarrow{\chi} & \mathcal{B}_\Sigma \\ {\scriptstyle \mathbb{L}_\mathcal{L}}\downarrow & & \downarrow{\scriptstyle \mathbb{L}_{\mathcal{L}'}} \\ \mathcal{B}_\Sigma & \xrightarrow{Id} & \mathcal{B}_\Sigma \\ {\scriptstyle yield_\mathcal{L}}\uparrow & & \uparrow{\scriptstyle yield_{\mathcal{L}'}} \\ k\text{-noisy text} & \xrightarrow{Id,\overline{Id}} & \text{Text} \end{array}$$

So we deduce the result from Theo. 1.

5 Denoising in the limit

Another way to learn from noisy data is to denoise the data on the fly, then to learn the language from these pure data. In order to denoise the data, we will see that it can even be useful to add more noise as a preliminary. The data processing sequence is then the following one:

$$\mathbf{Pres}(\mathcal{L}) \xrightarrow{\text{add noise}} \overline{\mathbf{Pres}}(\mathcal{L}) \xrightarrow{\text{remove noise}} \overline{\overline{\mathbf{Pres}}}(\mathcal{L})$$

where $\mathbf{Pres}(\mathcal{L})$ and $\overline{\mathbf{Pres}}(\mathcal{L})$ are noisy presentations and $\overline{\overline{\mathbf{Pres}}}(\mathcal{L})$ is a presentation of pure data. Once the presentation is denoised, we can then learn in the limit a language L' and use it to deduce the language L which interests us. Note that if we *strictly* denoise a presentation, *i.e.* if we remove all the noise and only the noise, we will then obtain directly $L' = L$.

Definition 8 (Denoisable in the limit). *Let* $\boldsymbol{Pres}(\mathcal{L})$ *be a class of* k*-noisy presentations. If there exists an algorithm* $\theta : X \times \bigcup_{f \in \boldsymbol{Pres}\,\mathcal{L}, i \in \mathbb{N}} \{f_i\} \to \{0,1\}$ *such that :* $\forall x \in X, \forall f \in \boldsymbol{Pres}(L), \exists n_x$ *such that* $\forall m \geq n_x\ \theta(x, f_m) = \theta(x, f_{n_x}) =$ isnoise$(x) =$ *if* $x \in N_k(L) \setminus L$ *then* 1 *else* 0*, then we say that the presentations of* $\boldsymbol{Pres}(\mathcal{L})$ *are denoisable in the limit.*

Note that the identification of the noise is not monotonic: we can have identified some data as pure and cannot guess for others. Moreover, denoising in the limit is not identification in the limit: the function isnoise is learned point-to-point but never in its globality.

In the following, we consider only learning from k*-noisy text*. In this case, $\theta_k(x, f_m) = 1$ indicates the fact that at the rank m the algorithm estimates that x is a noisy piece of data and therefore is not in L.

To denoise the data, we must thus know if the data belong to the target language or not, *i.e. we must be able to decide if a data is noise. For that, we will need to know the relations of proximity of the data compared one to the*

other, and in particular compared to those which belong indeed to the language. This concept of "neighbourhood" naturally leads to topology.

However, for our problem, traditional topology with its numerous axioms is too constraining. We thus will use pretopologic spaces which aim at defining "topologies with less axioms". For sake of clarity, we point out the definitions of the pretopologies and their properties in appendix.

Let I_k and E_k define the function allowing the deletion and the addition of noise: $I_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \subseteq L\}$ and $E_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \cap L \neq \emptyset\}$

Definition 9. *A language L is said to be* closed *for the pretopologic space $\mathbb{E}_j = (\Sigma^*, E_k \circ I_k, I_k \circ E_k)$ if and only if $I_j(E_j(L)) = L$ and a language class is* closed *if all its elements are closed.*

We can show that:

$$L \text{ closed} \Rightarrow (\forall x \in \Sigma^* N_k(x) \subseteq E_k(L) \Rightarrow x \in L) \tag{1}$$

The function I_k enables us to implement a way to denoise data:

Theorem 3. *Let k be the level of noise and $\mathbb{E}_k$ be a pretopologic space. If $\mathcal{L}$ is closed (for $\mathbb{E}_k$) then $\mathbf{Pres}(\mathcal{L})$ is k-denoisable in the limit.*

Proof. We consider the following algorithm θ_k: let f be a k-noisy presentation of a language L and $x \in N_k(L)$; $\theta_k(x, f_p) = 0$ if $x \in I_k(f_p)$ and 1 if $B_k(x) \not\subseteq f_p$.

Let f be a k-noisy presentation of a language L and $x \in N_k(L)$. If $\text{isnoise}(x) = 0$ *then $x \in L$ thus $B_k(x) \subseteq N_k(L)$ and as $f(\mathbb{N}) = N_k(L)$, there is a rank n_x such that $B_k(x) \subseteq f_{n_x}$ and thus $x \in I_k(B_k(x)) \subseteq I_k(f_{n_x})$. Consequently $\theta_k(X, f_{n_x}) = 0 = \text{isnoise}(x)$. Conversely, if* $\text{isnoise}(x) = 1$*, then $x \notin L$ and as L is closed for $\mathbb{E}_k$ then $B_k(x) \not\subseteq N_k(L)$ (cf equation 1) and thus $\forall p \in \mathbb{N}, B_k(x) \not\subseteq f_p$. Consequently, $\forall p \in \mathbb{N}, \theta(x, f_p) = 1 = \text{isnoise}(x)$.*

Example 1. Let $\overline{\mathcal{B}_\Sigma}$ be the set defined by $\overline{\mathcal{B}_\Sigma} = \left\{\overline{B_k(u)} : k \in \mathbb{N}, u \in \Sigma^*\right\}$ where $\overline{B_k(u)} = \Sigma^* \setminus B_k(u)$. Presentations of $\overline{\mathcal{B}_\Sigma}$ are denoisable in the limit. Indeed, we can show that balls are open and that the complementary of an open set is a closed set, thus the class $\overline{\mathcal{B}_\Sigma}$ is closed.

To add noise, however, can seem strange; nevertheless, it makes it possible to obtain the following result:

Theorem 4. *Let k be the level of noise and $\mathbb{E}_j$ be a pretopologic space. If $\mathcal{L}$ is closed and if $j \geq k$ then $\mathcal{L}$ is k-denoisable in the limit.*

Proof. Consider the algorithm $\theta_k(x, f_p) = 0$ if $x \in I_k(E_{j-k}(f_p))$ and 1 if not. Then take again the proof of Theo. 3. More intuitively, let f be a k-noisy presentation of L. For all p we define $g_p = E_{j-k}(f_p)$. As f is a presentation of $N_k(L)$, g is a presentation of $N_j(L)$. Moreover L is closed for $\mathbb{E}_j$ thus according to Theo. 3, g is j-denoisable in the limit and thus f is k-denoisable in the limit

		k=1					k=2				
r	$\lvert B_r(\lambda)\rvert$	$\lvert \boldsymbol{f}\rvert$	j=0	j=1	j=2	j=3	$\lvert \boldsymbol{f}\rvert$	j=0	j=1	j=2	j=3
1	3	6	0.183	1.774	1.876	1.876	12	0.004	1.338	1.761	1.765
2	7	14	0.278	4.711	5.462	5.473	28	0.026	3.818	5.126	5.330
3	15	30	0.420	11.017	12.929	13.157	60	0.030	9.112	12.453	13.000
4	31	62	0.422	21.755	29.213	29.592	124	0.041	22.831	27.690	29.244

Table 1. Addition of noise can be useful

However, the addition of noise will not allow the identification of new classes which were not learnable without noise. On the other hand, it can allow to learn more quickly.

Table 1 shows for different balls centred on λ, the size of the ball (the target) and for two level of noise (k), the size of the randomly generated k-noisy presentation of $B_r(\lambda)$ ($\lvert \boldsymbol{f}\rvert$), and the number of remaining elements after adding a level of noise of j and $(k+j)$-denoising the noisy presentation. We can easily see that once noise is added, we can validate data faster.

Lastly, the majority of the languages are naturally not totally denoisable in the limit. Nevertheless, it is possible to deduce *the class $\mathcal{L}$ from a class of language $\mathcal{L}$' by combining addition and deletion of noise.*

Example 2. Let $\mathcal{B}_\Sigma$ be the class of balls. We recall that this class is not closed. Let $L = B_r(u)$. Then $I_{j+k}(E_j(N_k(L))) = I_{j+k}(E_{j+k}(L))$ which contains an approximation of L, i.e., *L plus possibly some words (for example $bbbaaa \in I_1(E_1(B_4(aabb)))$ but $bbbaaa \notin B_4(aabb)$). However in $I_{j+k}(E_{j+k}(L))$, there exists a couple $(a^n v, b^n v)$ which are respectively the smallest and the greatest word of the longest words of L. These words enable us to deduce $r = n$ and $u = v$, thus to identify $L = B_r(u)$. Consequently, there is an algorithm allowing to identify indirectly $\mathcal{B}_\Sigma$ after an approximate denoising of the data.*

6 Conclusion

We introduced two techniques allowing to learn languages in presence of systematic noise. One of them is based on a theorem of reduction. The other uses the idea of the on the fly denoising of the data (denoising whose correction is obtained only in the limit). We also established the fact that this process could advantageously be accompanied by an over-noising of the data in order to accelerate the identification.

Several problems remain open: we did not tackle the questions of complexity. It is obvious, for example, that the over-noising should not be explicit since too expensive. Techniques simulating it must be introduced. The systematic noise is also a strong assumption: a more realistic model could be based on the fact that only a part (the majority?) of the noisy examples appears in the presentation. In the same way, we chose here to use a strict denoising: as long as all the elements of the noise of x did not appear, x is regarded as noise. Other strategies are

possible and deserve to be analyzed. Finally, balls are a first candidate of topologically robust languages. But other classes of languages, defined by topological properties, can be richer and maintain the necessary robustness.

Acknowledgment

Several ideas related in particular to the systematic noise and the balls were discussed in June 2004 with Remi Eyraud and Jose Oncina during its stay in Saint-Etienne as an invited professor.

References

1. *Sakakibara, Y.: Recent advances of grammatical inference. Theoretical Computer Science* ***185*** *(1997) 15–45*
2. *de la Higuera, C.: A bibliographical study of grammatical inference. Pattern Recognition* ***38*** *(2005) 1332–1348*
3. *Lang, K.J., Pearlmutter, B.A., Price, R.A.: Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. LNCS* ***1433*** *(1998) 1–12*
4. *de la Higuera, C.: Data complexity in Grammatical Inference. Number ISBN: 1-84628-171-7 in Advanced Information and Knowledge Processing. In: Data complexity in Pattern Recognition. Springer Verlag (2006)*
5. *Case, J., Jain, S., Sharma, A.: Synthesizing noise-tolerant language learners. Theoretical Computer Science* ***261*** *(2001) 31–56*
6. *Stephan, F.: Noisy inference and oracles. Theoretical Computer Science* ***185*** *(1997) 129–157*
7. *Wharton, R.M.: Approximate language identification. Information and Control* ***26*** *(1974) 236–255*
8. *Kearns, M., Valiant, L.: Cryptographic limitations on learning boolean formulae and finite automata. In: 21st ACM Symposium on Theory of Computing. (1989) 433–444*
9. *Kearns, M.: Efficient noise-tolerant learning from statistical queries. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing. (1993) 392–401*
10. *Sebban, M., Janodet, J.C.: On state merging in grammatical inference: a statistical approach for dealing with noisy data. In: Proceedings of ICML. (2003)*
11. *Habrard, A., Bernard, M., Sebban, M.: Improvement of the state merging rule on noisy data in probabilistic grammatical inference. In Lavrac, N., Gramberger, D., Blockeel, H., Todorovski, L., eds.: 10th European Conference on Machine Learning. Number 2837 in LNAI, Springer-Verlag (2003) 169–1180*
12. *Coste, F., Fredouille, D.: Unambiguous automata inference by means of state-merging methods. In: Proceedings of ECML (LNAI 2837). (2003) 60–71*
13. *Giles, C.L., Lawrence, S., Tsoi, A.: Noisy time series prediction using recurrent neural networks and grammatical inference. Machine Learning Journal* ***44*** *(2001) 161–183*
14. *Yokomori, T., Kobayashi, S.: Inductive learning of regular sets from examples: a rough set approach. In: Proc. of International Workshop on Rough Sets and Soft Computing. (1994)*

15. *Miclet, L., Bayoudh, S., Delhay, A.: Définitions et premières expériences en apprentissage par analogie dans les séquences. In Denis, F., ed.: CAP, PUG (2005) 31–48*
16. *Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite state automata – part I and II. Pattern Analysis and Machine Intelligence* ***27*** *(2005) 1013–1039*
17. *Abe, N., Warmuth, M.: On the computational complexity of approximating distributions by probabilistic automata. Machine Learning Journal* ***9*** *(1992) 205–260*
18. *Carrasco, R.C., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: Proceedings of ICGI (LNAI 862). (1994) 139–150*
19. *Gold, M.: Language identification in the limit. Information and Control* ***10*** *(1967) 447–474*
20. *Gold, E.M.: Complexity of automaton identification from given data. Information and Control* ***37*** *(1978) 302–320*
21. *de la Higuera, C.: Complexity and reduction issues in grammatical inference. Technical Report ISSN 0946-3852, Universität Tübingen (2005)*
22. *Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory* ***10*** *(1965) 707–710 Original in* Doklady Akademii Nauk SSSR *163(4): 845–848 (1965).*
23. *Wagner, R., Fisher, M.: The string-to-string correction problem. Journal of the ACM* ***21*** *(1974) 168–178*
24. *Takada, Y.: Grammatical inference for even linear languages based on control sets. Information Processing Letters* ***28*** *(1988) 193–199*
25. *de la Higuera, C., Casacuberta, F.: Topology of strings: median string is NP-complete. Theoretical Computer Science* ***230*** *(2000) 39–48*
26. *Belmandt, Z.: Manuel de prétopologie et ses applications. Hermès (1993)*
27. *Pawlak, Z.: Theory of rough sets: A new methodology for knowledge discovery (abstract). In: ICCI. (1990) 11*
28. *Kobayashi, S., Yokomori, T.: On approximately identifying concept classes in the limit. In: ALT. (1995) 298–312*

Appendix

We recall here some definitions of pretopology [26], then we define a pretopologic space adapted to the study of Σ^* and we study its properties within the framework of denoising in the limit.

Definition 10 (c-duality). *We note c the* complementary*: let U be a set, $\forall A \in \mathcal{P}(U), c(A) = U \setminus A = \bar{A}$. Two applications e and i from $\mathcal{P}(U)$ to $\mathcal{P}(U)$ are* c-duals *if and only if $i = c \circ e \circ c$ or $e = c \circ i \circ c$.*

Definition 11 (Pretopologic space). *(U, i, e) defines a* pretopologic space*, if and only if:(1.) i are e c-duals, (2.) $i(U) = U$, (3.) $\forall L \in \mathcal{P}(U), i(L) \subset L$.*

The concept of topology is thus a particular case of pretopology. It is a pretopologic space such as $\forall A, B \in \mathcal{P}(U), e(A \cup B) = e(A) \cup e(B)$ and $e(e(A)) = e(A)$. With the tools of the pretopology, we can model processes of extension $L = e^0(L) \subset e(L) \subset e[e(L)] \subset \ldots \subset e^n(L) \subset \ldots \subset U$ and erosion $L = i^0(L) \supset i(L) \supset i[i(L)] \supset \ldots \supset i^n(L) \supset \ldots \supset \emptyset$, what is not the case in topology because of the idempotence of the applications e and i.

Definition 12 (Closed and open sets). *Let (U, i, e) be a pretopologic space. K is a* closed set *of U if and only if $e(K) = K$ and L is an* open set *of U if and only if $i(L) = L$. A class of languages $\mathcal{L}$ is* closed *if and only if $\forall L \in \mathcal{L}$, L is a closed set and is* open *if and only if $\forall L \in \mathcal{L}$, L is an open set.*

Below, we define functions i and e thanks to which we will build the pretopologic spaces adapted to our study. We recall that the distance used (and in particular for the function of noise N) is the edit distance.

Definition 13 (Interior et exterior). *We call the* k-interior of L *the function defined by $I_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \subseteq L\}$ and the* k-exterior of L *the function defined by $E_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \cap L \neq \emptyset\}$.*

These concepts are similar to those of lower and upper approximation *of a set in the framework of* Rough Sets *[27, 28]*

A first naive idea would consist in choosing $i = I_k$ and $e = E_k$ as functions of interior and exterior. However, defined as this, the extension and erosion are too important to find interesting closed and open sets. We will then take $i = E_k \circ I_k$ and $e = I_k \circ E_k$. We now showthat these two functions fulfil the properties.

Lemma 3. *$I_k \circ E_k$ and $E_k \circ I_k$ are c-duals in Σ^*, i.e., $\forall L \in \mathcal{P}(\Sigma^*), I_k(E_k(L)) = \overline{E_k(I_k(\overline{L}))}$.*

Proof. *I_k and E_k are c-duals: $I_k(\overline{L}) = \{w \in \overline{\Sigma^* : N_k(\{w\}) \subseteq \overline{L}}\} = \{w \in \Sigma^* : N_k(\{w\}) \cap L = \emptyset\} = \overline{E_k(L)}$. So $E_k(I_k(\overline{L})) = I_k(\overline{I_k(\overline{L})}) = \overline{I_k(E_k(L))}$.*

Theorem 5. *$\mathbb{E}_k = (\Sigma^*, E_k \circ I_k, I_k \circ E_k)$ defines a pretopologic space, and then verifies: (1.) $I_k \circ E_k$ and $E_k \circ I_k$ are c-duals, (2.) $E_k(I_k(U)) = U$ and (3.) $\forall L \in \mathcal{P}(U), I_k(E_k(L)) \subset L$*

Proof. *(1.) By Lemma 3. (2.) straightforward. (3.) If $x \in E_k(I_k(L))$ then $N_k(\{x\}) \cap I_k(L) \neq \emptyset$, so $\exists y \in I_k(L) : d(x, y) \leq k$. Since $(d(x, y) \leq k \Rightarrow x \in N_k(\{y\}))$ and $(y \in I_k(L) \Rightarrow N_k(\{y\}) \subseteq L)$,we deduce $x \in L$ and $E_k(I_k(L)) \subseteq L$.*

The function E_k, respectively *I_k, allows to add noise to L,* respectively *to remove some noise. We can then use them within our framework of denoising in the limit. Note that $E_k \neq I_k^{-1}$ since $E_1(B_1(aa)) = B_2(aa)$ but $I_1(B_2(aa)) = B_1(aa) \cup \{\lambda, b\}$*