Identification à la limite de langages dans le cadre d'un bruit systématique

Frédéric Tantini, Colin de la Higuera et 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 : Pour étudier l'apprentissage à partir de données bruitées, il est usuel de se baser sur un modèle de bruit statistique. L'influence du bruit est alors envisagée selon des critères pragmatiques ou eux-mêmes statistiques, en se basant donc sur un paradigme prenant en compte une distribution des données. Dans cet article, nous étudions le bruit comme un phénomène non statistique, en définissant la notion de bruit systématique. Nous établissons différentes manières d'apprendre (à la limite) à partir de données bruitées. La première se base sur une technique de réduction entre problèmes et consiste à apprendre à partir des données qu'on sait bruitées, puis à débruiter la fonction apprise. La seconde consiste à débruiter à la volée les exemples d'apprentissage, à identifier ainsi *à la limite* les bons exemples, et à apprendre alors à partir d'exemples non corrompus. Nous donnons dans les deux cas des conditions suffisantes pour que l'apprentissage soit possible et montrons à travers différents exemples (provenant en particulier du domaine de l'inférence grammaticale) que nos techniques sont complémentaires.

Mots-clés : Identification à la limite, langages, bruit, prétopologie.

1 Introduction

L'inférence grammaticale (Sakakibara, 1997; de la Higuera, 2005a) est un domaine riche en algorithmes que l'on peut utiliser pour apprendre à partir de données séquentielles ou structurées : chaînes, mots, arbres ou termes. Parmi les nombreux avantages de ces techniques, on peut souligner la compréhensibilité des modèles appris, des théories solides qui permettent en particulier d'éviter de travailler avec des biais non explicites, la puissance des fonctions qui définissent les concepts (automates et grammaires), le fait que les données d'apprentissage peuvent être analysées dans leur globalité et non en n'en prenant en compte que des morceaux, *etc*. Mais ces qualités ont naturellement une contrepartie, et en particulier le fait que ces techniques résistent très faiblement (pour ne pas dire pas du tout) au bruit.

Le bruit apparaît dans les données pour de nombreuses raisons :

- Il peut être dû au fait que le biais utilisé est inadapté : si on essaye d'apprendre un

langage régulier à partir de données qui ne proviennent pas d'un langage régulier, on peut s'attendre à un problème ;

- Le bruit peut aussi être dû à des conditions expérimentales médiocres ou bien à un "nettoyage" des données qui est soit trop difficile à réaliser, soit encore trop couteux. Ceci peut se produire en reconnaissance de la parole, ou lorsque l'on désire apprendre à partir de fichiers HTML construits manuellement.

La gestion du bruit est un problème crucial et récurrent en apprentissage automatique en général. Pour ce qui concerne l'inférence grammaticale et le traitement des données séquentielles, on peut citer les lignes de recherche suivantes :

- Des travaux très théoriques ont été menés, soit dans le cadre de l'inférence inductive (Case *et al.*, 2001; Stephan, 1997) et en suivant la piste de résultats déjà anciens (Wharton, 1974), soit dans celui de l'apprentissage approximatif (Kearns & Valiant, 1989; Kearns, 1993);

- D'autres travaux ont essayé d'utiliser des idées bien fondées dans des algorithmes existants pour rendre ceux-ci plus robustes au bruit (Sebban & Janodet, 2003; Habrard *et al.*, 2003). Par ailleurs, les automates non déterministes sont probablement plus résistants au bruit que les déterministes (Coste & Fredouille, 2003);

- Des travaux plus pragmatiques ont été menés pour utiliser des techniques d'inférence grammaticale sur des séries chronologiques naturellement bruitées (Giles *et al.*, 2001) ;

- On peut aussi noter les travaux sur l'apprentissage de langages par approximation, qui se fondent sur la théorie des ensembles grossiers (*rough sets*) et qui donnent lieu à des algorithmes intrinsèquement plus résistants aux données bruitées (Yokomori & Kobayashi, 1994) ;

- Le paradigme de l'apprentissage par analogie a également fait l'objet d'une étude sous l'angle de sa résistance aux données bruitées (Miclet *et al.*, 2005) ;

- Enfin, une approche classique est celle qui concerne l'apprentissage d'automates stochastiques. Cette approche prend le parti d'éviter le problème en imposant un biais différent : les données proviennent d'une distribution elle-même représentée par un automate stochastique (Vidal *et al.*, 2005). Il ne s'agit alors plus d'apprendre un langage mais une distribution. Les travaux sont théoriques (Abe & Warmuth, 1992) et algorithmiques (Carrasco & Oncina, 1994).

On note que dans la plupart des travaux théoriques, le traitement est essentiellement statistique. Nous nous éloignons dans ce travail de cette veine majoritaire et explorons le cas d'un bruit *systématique* basé sur l'utilisation de la distance d'édition; nous étudions les propriétés de ce type de bruit dans le contexte de l'identification à la limite (Gold, 1967; Gold, 1978).

Dans ce contexte nous proposons différentes manières d'apprendre (à la limite) à partir de données bruitées. La première se base sur une technique de réduction entre problèmes et consiste à apprendre à partir des données qu'on sait bruitées, puis

à débruiter la fonction apprise. La seconde consiste à débruiter à la volée les exemples d'apprentissage, à identifier ainsi *à la limite* les bons exemples, et à apprendre à partir d'exemples non corrompus. Dans cette seconde approche nous démontrons qu'il est même possible (et parfois recommandé) d'ajouter du bruit supplémentaire pour accélérer l'apprentissage.

Nous donnons pour ces deux schémas des conditions suffisantes pour que l'apprentissage soit possible et montrons à travers différents exemples (et en particulier des exemples provenant du domaine de l'inférence grammaticale) que les techniques sont complémentaires. Les définitions que nous allons utiliser sont générales, nous les exploiterons dans le cadre des textes bruités systématiquement, mais elles nous semblent avoir vocation à être utilisées plus largement.

2 Préliminaires

En inférence grammaticale les concepts utilisés pour définir les fonctions (décrivant les langages) proviennent de la théorie des langages formels : les formalismes issus de la théorie des systèmes de réécriture (grammaires et automates) permettent de définir puissamment des ensembles de mots. Ces formalismes sont cependant peu robustes au bruit, ainsi qu'il a été noté par différents auteurs (Lang *et al.*, 1998; de la Higuera, 2006). Un des objectifs de ce travail est d'explorer des définitions de langages un peu plus adaptées à des situations où les données sont bruitées.

2.1 Langages

Un *alphabet* Σ est un ensemble fini non vide de symboles appelés *lettres*. Un *mot* w est une séquence finie $w = a_1 a_2 \dots a_n$ de lettres. $|w|$ dénote la longueur de w. Par la suite, les lettres seront désignées par $a, b, c, \dots$, les mots par $u, v, \dots, z$ et le mot vide par λ. L'ensemble de tous les mots est noté Σ^*. On appelle langage toute partie $L \subseteq \Sigma^*$.

Soient $\mathcal{L}$ une classe de langages et $\mathcal{R}(\mathcal{L})$ une classe de représentations des langages de $\mathcal{L}$. Les classes de langages qu'on considèrera peuvent être les langages réguliers ou algébriques, et les classes de représentations les grammaires algébriques, les automates finis déterministes ou encore les expressions régulières. On notera $\mathbb{L}_{\mathcal{L}} : \mathcal{R}(\mathcal{L}) \to \mathcal{L}$ la fonction qui pour n'importe quelle représentation retourne le langage correspondant. Cette fonction est surjective (chaque langage peut être représenté), mais pas nécessairement injective : deux représentations peuvent correspondre au même langage.

Nous supposerons enfin que le problème suivant est décidable : soient $w \in \Sigma^*$ et $G \in \mathcal{R}(\mathcal{L})$, $w \in \mathbb{L}_{\mathcal{L}}(G)$?

2.2 Identification à la limite

Le paradigme de l'identification à la limite a été introduit par Gold (Gold, 1967). Nous le donnons ici dans le formalisme de (de la Higuera, 2005b) qui permet d'étudier des réductions entre problèmes d'identification (voir section 4).

Définition 1 (Présentation)
*Soit $\mathcal{L}$ une classe de langages, une présentation de $L \in \mathcal{L}$ est une fonction $\mathbb{N} \to X$ où X est un ensemble. On note **Pres**$(\mathcal{L})$ un ensemble de présentations. Une présentation dénote un langage de $\mathcal{L}$, i.e. il existe une fonction $yield$: **Pres**$(\mathcal{L}) \to \mathcal{L}$. Si $L = yield(f)$ alors nous dirons que f est une présentation de L, ou $f \in$ **Pres**(L).*

Avec cette définition, les présentations sont à prendre au sens large : ce sont des séquences de n'importe quel type d'informations pouvant aider à l'apprentissage du langage.

Exemple 1
X peut être Σ^* dans le cas d'exemples positifs seulement. Si en outre, $yield(f) = f(\mathbb{N})$, alors la présentation est appelée un texte. Remarquons encore que dans le cas d'un informateur, c'est-à-dire de présentations composées d'exemples à la fois positifs et négatifs, on aurait $X = \Sigma^* \times \{0, 1\}$.

Étant données deux présentations f et g, si $f(\mathbb{N}) = g(\mathbb{N})$ alors $yield(f) = yield(g)$. Sinon, $\mathcal{L}$ n'est pas apprenable à partir de **Pres**$(\mathcal{L})$. En effet, deux langages partageant une même présentation ne peuvent pas être distingués l'un de l'autre. Si $f \in$ **Pres**$(\mathcal{L})$ et $g : \mathbb{N} \to X$ telle que $g(\mathbb{N}) = f(\mathbb{N})$ alors $g \in$ **Pres**$(\mathcal{L})$.

Soit f une présentation, on note f_n l'ensemble $\{f(j) : j < n\}$. Un *algorithme d'apprentissage* **alg** est un programme prenant les n premiers éléments d'une présentation et retournant une représentation :

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

La définition suivante est adaptée de (Gold, 1978) :

Définition 2 (Identifiable à la limite)
*On dit que $\mathcal{L}$ est identifiable à la limite à partir de **Pres**$(\mathcal{L})$ en terme de $\mathcal{R}(\mathcal{L})$ si et seulement si il existe un algorithme **alg** tel que pour tout $L \in \mathcal{L}$ et pour n'importe quelle présentation $f \in$ **Pres**(L), il existe un rang n tel que pour tout $m \geq n, \mathbb{L}_{\mathcal{L}}(\textbf{alg}(f_m)) = L$.*

2.3 Distances

Poser une distance sur Σ^* consiste à introduire une fonction $d : \Sigma^* \times \Sigma^* \to \mathbb{R}$ telle que: (i) $d(x, x) = 0$, (ii) $d(x, y) = d(y, x)$ et (iii) $d(x, y) \geq 0$. Les conditions suivantes, sans être obligatoires, sont également utiles:

$$d(x, y) = 0 \implies x = y$$

$$d(x, y) + d(y, z) \geq d(x, z)$$

La distance d'édition a été définie par Levenshtein en 1965 (Levenshtein, 1965). Il s'agit de compter le nombre minimal d'opérations nécessaires pour passer d'une chaîne à une autre. Les opérations de base sont l'insertion, la substitution et la suppression (d'un symbole).

Techniquement, étant données deux chaînes w et w' dans Σ^*, w se réécrit en w' en une étape si l'une des conditions suivantes est vraie :

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

On considère la fermeture réflexive et transitive de cette dérivation et on note $w \xrightarrow{k} w'$ si et seulement si w se réécrit en w' par k opérations.

Étant données deux chaînes w et w', la distance de Levenshtein entre w et w' notée $d_{edit}(w, w')$ est la plus petite valeur de k telle que $w \xrightarrow{k} w'$.

Exemple 2
$d_{edit}(abaa, aab) = 2$. $abaa$ se réécrit en aab via (par exemple) la suppression du b et la substitution du dernier a par un b.

La distance d'édition entre deux chaînes se calcule par programmation dynamique (Wagner & Fisher, 1974). De nombreuses variantes ont été étudiées et la distance a été adaptée au cas des chaînes circulaires ou des arbres. Les poids des différentes opérations d'édition peuvent également ne pas toujours valoir 1. Nous avons choisi dans ce travail de n'étudier que le cas standard. Voir aussi (Crochemore *et al.*, 2001).

Les classes de langages usuelles (définies par automates, grammaires,...) ne conviennent pas en cas de bruit. Le problème essentiel est que de façon quasi systématique, le changement d'un symbole dans un mot fait basculer celui-ci du langage à son complémentaire. Pour utiliser une image provenant d'un domaine dans lequel le bruit a été bien mieux analysé, c'est comme si, en dessinant sur un écran les mots d'un langage, aucune forme n'était perceptible : tous les langages ressembleraient à du gris uniforme. Nous introduisons donc des objets topologiques simples, les boules, qui intuitivement ne présentent pas ce problème.

Définition 3 (Boules)
Étant donnée $d(\cdot, \cdot)$ une distance sur Σ^*, la boule de centre $u \in \Sigma^*$ et de rayon $r \in \mathbb{N}$ est définie par $B_r(u) = \{w \in \Sigma^* : d(w, u) \leq r\}$. Une représentation de la boule $B_r(u)$ sera alors le couple (u, r). On note $\mathcal{B}_\Sigma$ l'ensemble de toutes les boules : $\mathcal{B}_\Sigma = \{B_k(u) : k \in \mathbb{N}, u \in \Sigma^*\}$.

La même distance donne lieu à la définition de bruité d'un langage :

Définition 4 (Bruité d'un langage)
Soit L un langage sur Σ^* et $d(.,.)$ une distance sur Σ^*, le bruité d'ordre k (ou le k-bruité*) de L est $N_k(L) = \{w : \exists x \in L, d(x, w) \leq k\}$.*

Comme nous l'avons dit, les boules se comportent bien en présence de bruit. En particulier, le bruité d'une boule est une boule :

Théorème 1

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

Démonstration:

($\subseteq$)

$$\begin{aligned} x \in N_k(B_{k'}(u)) &\Rightarrow \exists y \in B_{k'}(u) : d(y,x) \leq k \\ &\Rightarrow \exists y : d(u,y) \leq k' \wedge d(y,x) \leq k \\ &\Rightarrow d(u,x) \leq k' + k \end{aligned}$$

($\supseteq$) Soit $x \in B_{k+k'}(u)$ alors $d(u,x) \leq k + k'$. Supposons que $d(u,x) > k'$ (sinon c'est trivial). Le fait que $k' < d(u,x) \leq k+k'$ signifie que u peut être transformé en x par au plus $k + k'$ opérations d'édition. Soit y la chaîne obtenue après les k' premières opérations. Alors $d(u,y) = k'$ et $d(y,x) \leq k$; il s'en suit que $y \in B_{k'}(u)$ et $x \in N_k(B_{k'}(u))$. □

Par ailleurs, on peut facilement apprendre les boules dans un contexte non bruité :

Théorème 2
$\mathcal{B}_\Sigma$ est identifiable à la limite à partir de texte.

Démonstration: Par saturation, lorsque tous les points sont apparus, la boule peut être calculée. Il est à noter que si seuls certains points sont donnés, le problème est NP-difficile (de la Higuera & Casacuberta, 2000), mais si tous les points sont présents, c'est simple : soit B_{max} l'ensemble des plus longs mots apparus. Le centre de la boule u est le seul mot tel que $a^k u$ et $b^k u$ sont dans B_{max} où k est le plus grand entier tel que a^k et b^k sont des facteurs gauches de B_{max}. La boule est alors $B_k(u)$. □

On notera que si l'alphabet Σ ne contient qu'une seule lettre, une même boule peut être représentée de plusieurs façons ($B_2(a) = B_3(\lambda)$), mais cette caractéristique n'est pas gênante : de nombreuses classes de représentations ont cette propriété (automates, grammaires).

3 Identification à la limite à partir de données bruitées

Nous proposons ici un modèle de bruit que nous appelons *systématique* : une même donnée sera bruitée de toutes les façons possibles jusqu'à une distance prévue. On peut illustrer cette idée par des points de peinture sur une feuille de papier : en posant un objet sur la feuille les points deviennent des tâches.

Il est raisonnable d'étudier ce type de bruit dans le paradigme de l'identification à la limite. En effet, l'absence de distribution nous permet de penser à une convergence à la limite de l'algorithme d'apprentissage. L'identification à la limite a pour but de nous assurer de l'absence de biais caché. La distance utilisée sera la distance d'édition.

Une première remarque particulièrement adaptée au cas de l'inférence grammaticale (et qui va justifier la recherche d'autres modèles de langages) est que si une fois bruités deux langages ne sont pas distinguables l'un de l'autre, alors la classe de langages n'est pas résistante au bruit systématique.

Il est aisé de noter que c'est le cas pour la classe des langages rationnels, et de façon plus élargie pour toute classe définie par systèmes de réécriture. La possibilité de représenter dans ces classes *les fonctions de parité* est de nature à nous convaincre de la faible résistance au bruit de ces langages. Cela justifie de s'intéresser à des classes de langages définis autrement que par des grammaires.

Définition 5 (Présentation bruitée)
Une présentation bruitée est une présentation $f : \mathbb{N} \to X$ *à laquelle est associée une fonction* $\mathrm{isnoise} : X \to \{0, 1\}$ *indiquant si un élément particulier de la présentation est du bruit ou pas.*

La définition précédente est posée afin de répondre à une variété de situations, dont :

Exemple 3
Apprendre en présence de texte k*-bruité systématique (ou à partir d'une présentation* k*-bruitée) signifie donc apprendre* L *à partir de mots de* $N_k(L)$*, c'est-à-dire d'une présentation* f *telle que* $f(\mathbb{N}) = N_k(L)$*. La fonction* $\mathrm{isnoise}$ *vaut alors 0 sur les éléments de* L *et 1 sur ceux de* $N_k(L) \setminus L$*.*

On cherche donc à apprendre en présence de données bruitées, et on a deux manières d'envisager les choses que l'on peut expliciter par le diagramme suivant :

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

Dans ce diagramme la situation est celle où l'on cherche à apprendre un langage L de la classe $\mathcal{L}$ à partir d'une présentation bruitée de $\mathbf{Pres}(L)$. On peut essayer d'apprendre à la place un langage d'une autre classe qui incorporerait le bruit (la classe $\mathcal{L}$') ou débruiter les données pour se ramener à une présentation non bruitée dans $\overline{\mathbf{Pres}}(\mathcal{L})$ et apprendre à partir de celle-ci. Dans cette seconde stratégie, c'est de fait la fonction $\mathrm{isnoise}$ qu'on cherche à identifier.

4 Réduction

Une technique mise en œuvre dans de nombreux domaines de l'informatique est celle des *réductions*. Celles-ci permettent d'obtenir des résultats négatifs (tel problème est au moins aussi difficile que tel autre, connu comme étant trop dur) mais aussi d'utiliser les algorithmes valides dans un cas sur un autre.

C'est dans ce contexte que nous allons utiliser les réductions. En utilisant les arguments de (de la Higuera, 2005b), nous montrons que les boules sont identifiables à partir de données bruitées. Au delà de ce résultat, nous suggérons que les réductions sont une façon efficace d'apprendre à partir de données bruitées.

Rappelons qu'une situation d'identification est définie par la classe de langages, celle des représentations et le type de présentations admises.

Soient maintenant $\mathcal{L}$ et $\mathcal{L}$' les 2 classes de langages représentées respectivement par $R(\mathcal{L})$ and $R(\mathcal{L}')$.

On note par $\mathbb{L}_{\mathcal{L}}$ (respectivement $\mathbb{L}_{\mathcal{L}'}$) l'application surjective $R(\mathcal{L}) \rightarrow \mathcal{L}$ (respectivement $\mathbb{L}_{\mathcal{L}'} : R(\mathcal{L}') \rightarrow \mathcal{L}'$).

Etant donnée une application surjective $\phi : \mathcal{L} \rightarrow \mathcal{L}'$, notons ψ une application surjective $R(\mathcal{L}) \rightarrow R(\mathcal{L}')$ pour laquelle le diagramme suivant commute :

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

Donc :

$$\phi \circ \mathbb{L}_{\mathcal{L}} = \mathbb{L}_{\mathcal{L}'} \circ \psi$$

Plaçons nous maintenant du point de vue des présentations. Étant donnée une application surjective $\phi : \mathcal{L} \rightarrow \mathcal{L}'$, notons ξ une application (surjective) **Pres**$(\mathcal{L}) \rightarrow$ **Pres**$(\mathcal{L}')$ pour laquelle le diagramme suivant commute :

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

Il en résulte :

$$\phi \circ yield_{\mathcal{L}} = yield_{\mathcal{L}'} \circ \xi$$

Comme une présentation peut ne pas être une fonction calculable, l'application calculable associée à ξ se fait par morceaux :

Définition 6
Soient $\mathcal{L}$ une classe de langages représentés dans $R(\mathcal{L})$ avec des présentations dans ***Pres***$(\mathcal{L}) : \mathbb{N} \rightarrow X$ *et $\mathcal{L}$' une classe de langages représentés dans $R(\mathcal{L}')$ avec des présentations dans* ***Pres***$(\mathcal{L}') : \mathbb{N} \rightarrow Y$*. Une réduction de présentations $\xi :$* ***Pres***$(\mathcal{L}) \rightarrow$ ***Pres***$(\mathcal{L}')$ *telle que $\xi(\boldsymbol{f}) = \boldsymbol{g}$ est* calculable *si et seulement si il existe une fonction calculable $\overline{\xi} : X \rightarrow 2^Y$ avec $\bigcup_{i \in \mathbb{N}} \overline{\xi}(\boldsymbol{f}(i)) = \boldsymbol{g}(\mathbb{N})$.*

$\overline{\xi}$ est la description en chaque point de la fonction ξ. Nous supposerons que $\forall i \in \mathbb{N}$, $\overline{\xi}(\boldsymbol{f}(i))$ est un ensemble fini.

En combinant les deux diagrammes précédents on obtient :

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

Théorème 3
Si

1 $\mathcal{L}$' *est apprenable en termes de* $R(\mathcal{L}')$ *à partir de* ***Pres***$(\mathcal{L}')$*,*

2 *il existe une fonction calculable* $\chi : R(\mathcal{L}') \to R(\mathcal{L})$ *et une fonction calculable* $\psi : R(\mathcal{L}) \to R(\mathcal{L}')$ *telle que* $\psi \circ \chi =$*Id et*

3 ξ *est une réduction calculable,*

alors $\mathcal{L}$ *est identifiable par* $R(\mathcal{L})$ *à partir de* ***Pres***$(\mathcal{L})$*.*

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

Démonstration:
Soit **alg2** un algorithme d'apprentissage qui identifie $\mathcal{L}$'. Considerons l'algorithme **alg1** ci-dessous, qui prend les n premiers éléments ($\boldsymbol{f}_n$) d'une présentation $\boldsymbol{f}$ et exécute :

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

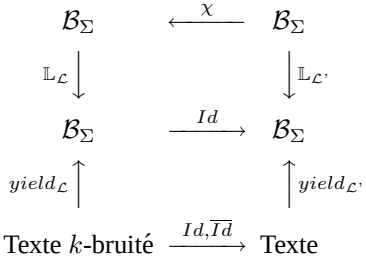
$G_{\mathcal{L}}$ et $G_{\mathcal{L}'}$ sont des grammaires de $R(\mathcal{L})$ et $R(\mathcal{L}')$. Comme ξ est calculable, $\boldsymbol{g}_m$ peut effectivement être construit. □

Comme conséquence du théorème 3, on retrouve des résultats connus comme l'apprentissage des grammaires linéaires équilibrées (Takada, 1988), par réduction à partir des automates finis déterministes. Dans le contexte des données bruitées, on a :

Théorème 4
$\mathcal{B}_\Sigma$ *est identifiable à la limite à partir de texte* k*-bruité.*

Démonstration:
D'après le théorème 1, le bruité d'une boule est une boule. De plus, par le théorème 2, $\mathcal{B}_\Sigma$ est identifiable à la limite à partir de texte. Enfin, en prenant χ=*si le rayon de la boule est au moins* k*, déduire* k *du rayon, sinon retourner identité*, on a le diagramme

suivant :



On déduit donc le résultat à partir du théorème 3. □

5 Débruitage à la limite

Une autre façon d'apprendre à partir de données bruitées est de débruiter les données à la volée, puis d'apprendre le langage à partir de ces données non bruitées. Afin de débruiter les données, nous verrons qu'il peut même être utile d'ajouter au préalable plus de bruit. La chaîne de traitement est alors la suivante :

$$\textbf{Pres}(\mathcal{L}) \xrightarrow{\text{ajouter du bruit}} \overline{\textbf{Pres}}(\mathcal{L}) \xrightarrow{\text{enlever du bruit}} \overline{\overline{\textbf{Pres}}}(\mathcal{L}) \quad (1)$$

où $\textbf{Pres}(\mathcal{L})$ et $\overline{\textbf{Pres}}(\mathcal{L})$ sont des présentations bruitées et $\overline{\overline{\textbf{Pres}}}(\mathcal{L})$ une présentation de données non (ou peu) bruitée. Une fois la présentation débruitée, on peut alors apprendre à la limite un langage L' puis s'en servir pour déduire le langage L qui nous intéresse. Notons que si l'on débruite *strictement* la présentation, c'est-à-dire si l'on supprime tout le bruit et uniquement le bruit, on aura alors directement $L' = L$.

Définition 7 (Débruitable à la limite)
Soit $\textbf{Pres}(\mathcal{L})$ *une classe de présentations* k*-bruitées. S'il existe un algorithme* $\theta : X \times \bigcup_{f \in \textbf{Pres}(\mathcal{L}), i \in \mathbb{N}} \{f_i\} \to \{0,1\}$ *telle que :* $\forall x \in X, \forall f \in \textbf{Pres}(L), \exists n_x$ *tel que* $\forall m \geq n_x$ $\theta(x, f_m) = \theta(x, f_{n_x}) = \text{isnoise}(x) =$ *si* $x \in N_k(L) \setminus L$ *alors* 1 *sinon* 0*, alors les présentations de* $\textbf{Pres}(\mathcal{L})$ *sont débruitables à la limite.*

Notons que l'identification du bruit n'est pas monotone : on peut avoir identifié certaines données comme étant bruitées et ne pas pouvoir (encore) le faire pour d'autres. Par ailleurs, le débruitage à la limite n'est pas de l'identification à la limite dans la mesure où la fonction isnoise sera apprise point à point mais jamais dans son ensemble.

Dans la suite, nous ne considérons que l'apprentissage à partir de *texte* k*-bruité*. Dans ce cas, $\theta_k(x, f_m) = 1$ indique le fait qu'au rang m l'algorithme estime que x est une donnée bruitée et donc ne fait pas partie de L.

Pour débruiter les données, nous devons donc savoir si elles appartiennent au langage cible ou non, c'est-à-dire pouvoir décider si une donnée est du bruit. Pour cela, nous allons avoir besoin de connaître les relations de proximité des données les unes par rapport aux autres, et notamment par rapport à celles qui appartiennent effectivement au langage. Cette notion de "voisinage" fait naturellement appel à de la topologie.

Cependant, pour notre problème, la topologie classique et son grand nombre d'axiomes sont trop contraignants. Nous allons donc utiliser les espaces prétopologiques qui visent à définir des "topologies possédant moins d'axiomes". Dans un souci de clarté, nous rappelons les définitions des prétopologies et leurs propriétés en annexe.

Définissons maintenant les fonctions I_k et E_k permettant de supprimer et d'ajouter du bruit :

$$I_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \subseteq L\}$$
$$E_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \cap L \neq \emptyset\}$$

Définition 8 (Langage fermé)
On dit qu'un langage L est fermé *pour l'espace prétopologique $\mathbb{E}_j = (\Sigma^*, E_k \circ I_k, I_k \circ E_k))$ (cf. annexe) si et seulement si $I_j(E_j(L)) = L$ et qu'une classe de langage est* fermée *si tous ses éléments sont fermés.*

On peut montrer que :

$$L \text{ fermé} \Rightarrow \forall x \in \Sigma^* N_j(x) \subseteq E_j(L) \Rightarrow x \in L \tag{2}$$

La fonction I_k nous permet de mettre en œuvre un débruitage des données :

Théorème 5
Soient $\mathbb{E}_k$ un espace prétopologique avec k donné et N_k la fonction de bruit. Si $\mathcal{L}$ est fermée (pour $\mathbb{E}_k$) alors ***Pres***$(\mathcal{L})$ *est k-débruitable à la limite.*

Démonstration:
On considère l'algorithme θ_k suivant : soient f une présentation k-bruitée d'un langage L et $x \in N_k(L)$; on pose $\theta_k(x, f_p) = 0$ si $x \in I_k(f_p)$ et 1 si $B_k(x) \not\subseteq f_p$.

Soient f une présentation k-bruitée d'un langage L et $x \in N_k(L)$. Si $\text{isnoise}(x) = 0$ alors $x \in L$ donc $B_k(x) \subseteq N_k(L)$ et comme $f(\mathbb{N}) = N_k(L)$, il existe un rang n_x tel que $B_k(x) \subseteq f_{n_x}$ et donc $x \in I_k(B_k(x)) \subseteq I_k(f_{n_x})$. Par conséquent $\theta_k(x, f_{n_x}) = 0 = \text{isnoise}(x)$. Inversement, si $\text{isnoise}(x) = 1$, alors $x \notin L$ et comme L est un fermé pour $\mathbb{E}_k$ alors $B_k(x) \not\subseteq N_k(L)$ (cf équation 2) et donc $\forall p \in \mathbb{N}, B_k(x) \not\subseteq f_p$. Par conséquent, $\forall p \in \mathbb{N}, \theta(x, f_p) = 1 = isnoise(x)$. □

Exemple 4
Soit $\overline{\mathcal{B}_\Sigma}$ l'ensemble défini par $\overline{\mathcal{B}_\Sigma} = \left\{\overline{B_k(u)} : k \in \mathbb{N}, u \in \Sigma^*\right\}$ où $\overline{B_k(u)} = \Sigma^* \setminus B_k(u)$. Les présentations de $\overline{\mathcal{B}_\Sigma}$ sont débruitables à la limite. En effet, on peut montrer que les boules sont des ouverts et que le complémentaire d'un ouvert est un fermé, donc la classe $\overline{\mathcal{B}_\Sigma}$ est fermée. D'après le théorème précédent, ses présentations sont alors débruitables à la limite.

Ajouter du bruit, par contre, peut paraître étrange, néanmoins, il permet d'obtenir le résultat suivant :

Théorème 6
Soit $\mathbb{E}_j$ un espace prétopologique avec j donné, soit N_k la fonction de bruit. Si $\mathcal{L}$ est fermée et si $j \geq k$ alors $\mathcal{L}$ est k-débruitable à la limite.

Démonstration:
Il suffit de considérer l'algorithme $\theta_k(x, f_p) = 0$ si $x \in I_k(E_{j-k}(f_p))$ et 1 sinon puis de reprendre la preuve du théorème 5. Plus intuitivement, soit f une présentation k-bruitée de L. Pour tout p on définit $g_p = E_{j-k}(f_p)$. Comme f est une présentation de $N_k(L)$, g est une présentation de $N_j(L)$. De plus L est un fermé pour $\mathbb{E}_j$ donc d'après le théorème 5, g est j-débruitable à la limite et donc f est k-débruitable à la limite. □

En outre, ajouter du bruit ne permet clairement pas d'apprendre à la limite des classes qui ne sont pas apprenables à partir de présentations non bruitées. En revanche, il peut permettre d'apprendre plus vite :

Exemple 5
Soit la classe des boules centrées sur λ, considérons la boule $B_2(\lambda)$. Soit f une présentation 1-bruitée de cette boule commençant par $f_4 = \{b, abb, aaa, baa\}$. Si l'on veut débruiter directement, on obtient $I_1(f_4) = \emptyset$, et il faudra encore d'autres exemples pour arriver à dégager des éléments non bruités. En revanche, si l'on ajoute du bruit de niveau 1 et que l'on débruite, on obtient $I_2(E_1(f_4)) = \{\lambda, a, b, aa\}$ ce qui suffit à retrouver $B_2(\lambda)$, la plus petite boule contenant $I_2(E_1(f_4))$. De plus, si le prochain élément qui apparaît est aab, alors on obtient encore $I_1(f_5) = \emptyset$ lorsque l'on n'ajoute aucun bruit et $I_2(E_1(f_5)) = B_2(\lambda)$ en ajoutant un peu de bruit.

Enfin, la plupart des langages ne sont naturellement pas totalement débruitables à la limite. Néanmoins, en combinant ajout et suppression de bruit, on peut se ramener à une classe de langages $\mathcal{L}$' à partir de laquelle il est possible de *déduire* la classe $\mathcal{L}$.

Exemple 6
Soit la classe des boules $\mathcal{B}_\Sigma$. Rappelons que cette classe n'est pas fermée. Soit $L = B_r(u)$. On a alors $I_{j+k}(E_j(N_k(L))) = I_{j+k}(E_{j+k}(L))$ qui contient une approximation de L, i.e.*, L plus éventuellement quelques mots (*$bbbaaa \in I_1(E_1(B_4(aabb)))$ *mais $bbbaaa \notin B_4(aabb)$). Or dans $I_{j+k}(E_{j+k}(L))$, il existe un couple ($a^n v$, $b^n v$) qui sont respectivement le plus petit et le plus grand mot des mots les plus longs de L. Ces mots nous permettent de déduire $r = n$ et $u = v$, donc d'identifier $L = B_r(u)$. Par conséquent, on a un algorithme permettant d'identifier indirectement $\mathcal{B}_\Sigma$ après un débruitage approximatif des données.*

6 Conclusion

Nous avons introduit deux techniques permettant d'apprendre des langages en présence de bruit systématique. L'une d'elles se base sur un théorème de réduction. L'autre utilise l'idée du débruitage à la volée des données (débruitage dont la correction n'est obtenue qu'à la limite).

Nous avons aussi établi le fait que ce processus pouvait être avantageusement accompagné d'un sur-bruitage des données afin d'accélérer l'identification.

Il reste de nombreuses questions ouvertes :

- les questions de complexité n'ont pas été abordées : il est évident, par exemple, que le sur-bruitage ne doit pas être explicite car trop coûteux. Des techniques permettant de le simuler doivent être introduites.
- le bruit systématique est une hypothèse forte : un modèle plus réaliste pourrait se baser sur le fait que seule une partie (la majorité ?) des exemples bruités apparaisse dans la présentation.
- de même, nous avons choisi ici d'utiliser un débruitage strict : tant que tous les éléments du bruité de x ne sont pas apparus, x est considéré comme du bruit. D'autres stratégies sont envisageables et méritent d'être analysées.
- les boules sont un premier candidat de langages topologiquement robustes. Mais d'autres classes de langages, définies par des propriétés topologiques, peuvent être autrement plus riches tout en maintenant la robustesse nécessaire.

Remerciements

Plusieurs idées concernant en particulier le bruit systématique et les boules ont été discutées en juin 2004 avec Rémi Eyraud et Jose Oncina lors de son séjour à Saint-Etienne en tant que professeur invité.

References

ABE N. & WARMUTH M. (1992). On the computational complexity of approximating distributions by probabilistic automata. *Machine Learning Journal*, **9**, 205–260.

BELMANDT Z. (1993). *Manuel de prétopologie et ses applications*. Hermès.

CARRASCO R. C. & ONCINA J. (1994). Learning stochastic regular grammars by means of a state merging method. In R. C. CARRASCO & J. ONCINA, Eds., *Grammatical Inference and Applications, Proceedings of ICGI '94*, number 862 in LNAI, p. 139–150, Berlin, Heidelberg: Springer-Verlag.

CASE J., JAIN S. & SHARMA A. (2001). Synthesizing noise-tolerant language learners. *Theoretical Computer Science*, **261**(1), 31–56.

COSTE F. & FREDOUILLE D. (2003). Unambiguous automata inference by means of state-merging methods. In (Lavrac *et al.*, 2003), p. 60–71.

CROCHEMORE M., HANCART C. & LECROQ T. (2001). *Algorithmique du texte*. Vuibert.

DE LA HIGUERA C. (2005a). A bibliographical study of grammatical inference. *Pattern Recognition*, **38**(9), 1332–1348.

DE LA HIGUERA C. (2005b). *Complexity and reduction issues in grammatical inference*. Rapport interne ISSN 0946-3852, Universität Tübingen.

DE LA HIGUERA C. (2006). *Data complexity in Pattern Recognition*, chapter Data complexity in Grammatical Inference. Number ISBN: 1-84628-171-7 in Advanced Information and Knowledge Processing. Springer Verlag.

DE LA HIGUERA C. & CASACUBERTA F. (2000). Topology of strings: median string is NP-complete. *Theoretical Computer Science*, **230**, 39–48.

GILES C. L., LAWRENCE S. & TSOI A. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning Journal*, **44**(1), 161–183.

GOLD E. M. (1978). Complexity of automaton identification from given data. *Information and Control*, **37**, 302–320.

GOLD M. (1967). Language identification in the limit. *Information and Control*, **10**(5), 447–474.

HABRARD A., BERNARD M. & SEBBAN M. (2003). Improvement of the state merging rule on noisy data in probabilistic grammatical inference. In (Lavrac *et al.*, 2003), p. 169–1180.

KEARNS M. (1993). Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, p. 392–401.

KEARNS M. & VALIANT L. (1989). Cryptographic limitations on learning boolean formulae and finite automata. In *21st ACM Symposium on Theory of Computing*, p. 433–444.

KOBAYASHI S. & YOKOMORI T. (1995). On approximately identifying concept classes in the limit. In *ALT*, p. 298–312.

LANG K. J., PEARLMUTTER B. A. & PRICE R. A. (1998). Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. *LNCS*, **1433**, 1+.

N. LAVRAC, D. GRAMBERGER, H. BLOCKEEL & L. TODOROVSKI, Eds. (2003). *14th European Conference on Machine Learning*, number 2837 in LNAI. Springer-Verlag.

LEVENSHTEIN V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, **10**(8), 707–710. Original in *Doklady Akademii Nauk SSSR* 163(4): 845–848 (1965).

MICLET L., BAYOUDH S. & DELHAY A. (2005). Définitions et premières expériences en apprentissage par analogie dans les séquences. In F. DENIS, Ed., *CAP*, p. 31–48: PUG.

PAWLAK Z. (1990). Theory of rough sets: A new methodology for knowledge discovery (abstract). In *ICCI*, p.1̃1.

SAKAKIBARA Y. (1997). Recent advances of grammatical inference. *Theoretical Computer Science*, **185**, 15–45.

SEBBAN M. & JANODET J.-C. (2003). On state merging in grammatical inference: a statistical approach for dealing with noisy data. In *Proceedings of ICML*.

STEPHAN F. (1997). Noisy inference and oracles. *Theoretical Computer Science*, **185**, 129–157.

TAKADA Y. (1988). Grammatical inference for even linear languages based on control sets. *Information Processing Letters*, **28**(4), 193–199.

VIDAL E., THOLLARD F., DE LA HIGUERA C., CASACUBERTA F. & CARRASCO R. C. (2005). Probabilistic finite state automata – part I and II. *Pattern Analysis and Machine Intelligence*, **27**(7), 1013–1039.

WAGNER R. & FISHER M. (1974). The string-to-string correction problem. *Journal of the ACM*, **21**, 168–178.

WHARTON R. M. (1974). Approximate language identification. *Information and Control*, **26**, 236–255.

YOKOMORI T. & KOBAYASHI S. (1994). Inductive learning of regular sets from examples: a rough set approach. In *Proc. of International Workshop on Rough Sets and Soft Computing*.

Annexe

Nous reprenons ici quelques définitions du Manuel de prétopologie et ses applications (Belmandt, 1993), puis nous définissons un espace prétopologique adapté à l'étude de Σ^* et nous en étudions les propriétés dans le cadre du débruitage à la limite.

Définition 9 (c-dualité)
Notons c le complémentaire *: soit U un ensemble, $\forall A \in \mathcal{P}(U), c(A) = U \setminus A = \bar{A}$. Deux applications e et i de $\mathcal{P}(U)$ dans $\mathcal{P}(U)$ sont* c-duales *si et seulement si elles vérifient $i = c \circ e \circ c$ ou $e = c \circ i \circ c$.*

Définition 10 (Espace prétopologique)
(U, i, e) *définit un* espace prétopologique*, si et seulement si :*

1. *i et e sont c-duales,*
2. *$i(U) = U$, (ou $e(\emptyset) = \emptyset$)*
3. *$\forall L \in \mathcal{P}(U), i(L) \subset L$ (ou $L \subset e(L)$).*

La notion de topologie est donc un cas particulier de prétopologie. C'est un espace prétopologique tel que $\forall A, B \in \mathcal{P}(U), e(A \cup B) = e(A) \cup e(B)$ et $e(e(A)) = e(A)$. Avec les outils de la prétopologie, nous pouvons donc modéliser des processus d'extension $L = e^0(L) \subset e(L) \subset e[e(L)] \subset \ldots \subset e^n(L) \subset \ldots \subset U$ et d'érosion $L = i^0(L) \supset i(L) \supset i[i(L)] \supset \ldots \supset i^n(L) \supset \ldots \supset \emptyset$, ce qui n'est pas le cas en topologie à cause de l'idempotence des applications e et i.

Définition 11 (Ensembles fermés et ouverts)
Soit (U, i, e) un espace prétopologique. K est un ensemble fermé *de U si et seulement si $e(K) = K$ et L est un* ensemble ouvert *de U si et seulement si $i(L) = L$. Une classe de langages $\mathcal{L}$ est* fermée *si et seulement si $\forall L \in \mathcal{L}$, L est un ensemble fermé et est* ouverte *si et seulement si $\forall L \in \mathcal{L}$, L est un ensemble ouvert.*

Après ces rappels sur les prétopologies, nous allons pouvoir définir des fonctions i et e grâce auxquelles nous construirons des espaces prétopologiques adaptés à notre étude. Nous rappelons pour cela que la distance utilisée (et notamment pour la fonction de bruit N) est la distance d'édition.

Définition 12 (Intérieur et extérieur)
Notons $I(L)$ l'intérieur *de L : $I(L) = \{w \in \Sigma^* : N_1(\{w\}) \subseteq L\}$ et $E(L)$* l'extérieur *de L : $E(L) = \{w \in \Sigma^* : N_1(\{w\}) \cap L \neq \emptyset\}$.*
Nous appelons le k-intérieur de L *la fonction définie par*

$$I_0(L) = L, \forall k \in \mathbb{N}\ I_k(L) = I[I_{k-1}(L)]$$

et le k-extérieur de L *la fonction définie par*

$$E_0(L) = L, \forall k \in \mathbb{N}\ E_k(L) = E[E_{k-1}(L)].$$

On peut alors montrer par récurrence que $I_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \subseteq L\}$ et $E_k(L) = \{w \in \Sigma^* : N_k(\{w\}) \cap L \neq \emptyset\}$.

Ces notions ne sont pas sans rappeller celles des Rough Sets (Pawlak, 1990) utilisés notamment par Satoshi Kobayashi et Takashi Yokomori (Kobayashi & Yokomori, 1995), les *lower approximation* et *upper approximation* d'un ensemble, avec lesquelles elles partagent un grand nombre de propriétés sur les intersections et les unions.

L'intuition voudrait que l'on prenne $i = I_k$ et $e = E_k$ comme fonction d'intérieur et d'extérieur. Cependant, définies comme ceci, l'extension et l'érosion sont trop importantes pour dégager des fermés et ouverts intéressants et non triviaux. Nous allons alors prendre $e = I_k \circ E_k$ et $i = E_k \circ I_k$. Nous montrons maintenant que ces deux fonctions remplissent bien les propriétés attendues.

Lemme 1
$I_k \circ E_k$ et $E_k \circ I_k$ sont c-duales dans Σ^*, i.e., $\forall L \in \mathcal{P}(\Sigma^*), I_k(E_k(L)) = \overline{E_k(I_k(\overline{L}))}$.

Démonstration:
I_k and E_k sont c-duales : $I_k(\overline{L}) = \{w \in \Sigma^* : \underline{N_k(\{w\}) \subseteq \overline{L}}\} = \{w \in \Sigma^* : N_k(\{w\}) \cap L = \emptyset\} = \overline{E_k(L)}$. Donc $E_k(I_k(\overline{L})) = I_k(\overline{\overline{I_k(\overline{L})}}) = \overline{I_k(E_k(L))}$. □

Théorème 7
$\mathbb{E}_k = (\Sigma^*, E_k \circ I_k, I_k \circ E_k)$ définit un espace prétopologique, c'est-à-dire vérifie :

1. *$I_k \circ E_k$ et $E_k \circ I_k$ sont c-duales,*
2. *$E_k(I_k(\Sigma^*)) = \Sigma^*$,*
3. *$\forall L \in \mathcal{P}(\Sigma^*), E_k(I_k(L)) \subset L$*

Démonstration:

1. lemme 1.
2. trivial.
3. $x \in E_k(I_k(L)) \Rightarrow N_k(\{x\}) \cap I_k(L) \neq \emptyset \Rightarrow \exists y \in I_k(L) : d(x, y) \leq k$.
   $d(x, y) \leq k \Rightarrow x \in N_k(\{y\})$ et $y \in I_k(L) \Rightarrow N_k(\{y\}) \subseteq L$ donc $x \in L$ et $E_k(I_k(L)) \subseteq L$. □

La fonction E_k, *respectivement* I_k, ajoute du bruit à L, *respectivement* enlève du bruit. Nous pouvons alors les utiliser dans notre cadre de débruitage à la limite. Notons cependant que $E_k \neq I_k^{-1}$.