Improvement of the State Merging Rule on Noisy Data in Probabilistic Grammatical Inference

Amaury Habrard, Marc Bernard, and Marc Sebban

EURISE – Université Jean Monnet de Saint-Etienne
23, rue du Dr Paul Michelon – 42023 Saint-Etienne cedex 2 – France
{amaury.habrard,marc.bernard,marc.sebban}@univ-st-etienne.fr

Abstract. In this paper we study the influence of noise in probabilistic grammatical inference. We paradoxically bring out the idea that specialized automata deal better with noisy data than more general ones. We propose then to replace the statistical test of the ALERGIA algorithm by a more restrictive merging rule based on a test of proportion comparison. We experimentally show that this way to proceed allows us to produce larger automata that better treat noisy data, according to two different performance criteria (perplexity and distance to the target model).

Keywords: probabilistic grammatical inference, noisy data, statistical approaches

1 Introduction

Nowadays the quantity of data stored in databases becomes more and more important. Beyond the fact that the amount of information is hard (in terms of complexity) to process by machine learning algorithms, these data often contain a high level of noise. To deal with this problem, many data reduction techniques aim at either removing irrelevant instances (prototype selection [1]) or deleting irrelevant features (feature selection [2]). These techniques always need positive examples and negative examples of the concept to learn. An outlier is seen as a positive (resp. negative) instance which should be negatively (resp. positively) labeled in absence of noise. However, in some real applications, it is difficult, even impossible, to have negative examples, that is for example the case in natural language processing. In such a context, learning algorithms exploit statistical information to infer a model allowing to define a probability distribution on positive data. Because of the absence of negative examples, standard data reduction techniques are not adapted for removing outliers, which require in fact specific processes. In the context of probabilistic models, an outlier can be seen as a weakly relevant instance, *i.e.* weakly probable because of noise. While such models are *a priori* known to be more efficient for dealing with noisy data, no study, as far as we know, has been devoted to analyze the impact of noise in the specific field of probabilistic grammatical inference.

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 169–180, 2003.

Grammatical inference [3] is a subtopic of machine learning which aims at learning models from a set of sequences (or trees). Probabilistic grammatical inference allows to learn probabilistic automata defining a distribution on the language recognized by the automaton. In this framework, the data (always considered as positive) are supposed to be generated from a probability distribution, and the objective is to learn the automaton which generated the data. A successful learning task produces a probabilistic automaton which gives a good estimation of the initial distribution.

In this paper we are interested in probabilistic grammatical inference algorithms based on state merging techniques. In particular, we study the behavior of the ALERGIA algorithm [4, 5] in the context of noisy data. Our thought concerns the generalization process: we think that a generalization issued from the merging of noisy and correct data in the same state is particularly irrelevant. This can be dramatic, especially in cyclic automata, because this kind of generalization could increase the deviation from the initial distribution. Then we need to restrict the state merging rule for avoiding such situations. In ALERGIA, the generalization process consists in merging states that are considered statistically close according to a test based on the Hœffding bound [6]. However this bound is an asymptotic one and is then only relevant for large samples. To deal with small sets, [7] proposed a more general approach (called MALERGIA) using multinomial statistical tests in the merging decision. Despite its good performances with small dataset sizes, MALERGIA has a major disadvantage: a high complexity on very small datasets for which the calculation of a costly statistic is needed. In this paper we overcome both the ALERGIA and MALERGIA drawbacks. We replace the original test of ALERGIA by a more restrictive one based on a test of proportion comparison. This test can deal with both large and small datasets and we show experimentally that it better performs in the context of noisy data.

After a brief recall about probabilistic finite state automata and their learning algorithms, we describe in Section 2 the state merging rule of the algorithm ALERGIA and its extension with a multinomial approach in MALERGIA. In Section 3, we propose a new approach based on a test of proportion comparison. We theoretically prove that the bound of our test is always smaller than the Hœffding's one, expressing the fact that the merge will be always more difficult to be accepted in presence of noise. We also relate our work in comparison to the multinomial approach. Section 4 deals with experiments comparing the three approaches with different levels of noise.

2 Learning of Probabilistic Finite State Automata

Probabilistic Finite State Automata (PFSA) are a probabilistic extension of finite state automata and define a probability distribution on the strings recognized by the automata.

2.1 Definitions and Notations

Definition 1 *A PFSA A is a 6-tuple $(Q, \Sigma, \delta, p, q_0, F)$. Q is a finite set of states. Σ is the alphabet. $\delta : Q \times \Sigma \to Q$ is the transition function. $p : Q \times \Sigma \to [0, 1]$ is the probability of a transition. q_0 is the initial state. $F : Q \to [0, 1]$ is the probability for a state to be a final state.*

In this article, we only consider deterministic PFSA (called PDFA), *i.e.* where δ is injective. This means that given a state q and a symbol s, the state reached from the state q by the symbol s is unique if it exists. In order to define a probability distribution on Σ^* (the set of all strings built on Σ), p and F must satisfy the following consistency constraint: $\forall q \in Q, F(q) + \sum_{a \in \Sigma} p(q, a) = 1$.

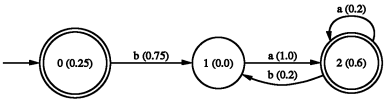

Fig. 1. A PDFA with $q_0 = 0$ and its probabilities

A string $s_0 \dots s_{n-1}$ is recognized by an automaton A iff there exists a sequence of states $e_0 \dots e_n$ such that: (i) $e_0 = q_0$, (ii) $\forall i \in [0, l-1]$, $\delta(e_i, s_i) = e_{i+1}$, (iii) $F(e_n) \neq 0$. Then the automaton assigns to the string the following probability:

$$P_A(s_0 \dots s_{n-1}) = \left(\Pi_{i=0}^{n-1} p(e_i, s_i)\right) * F(e_n)$$

For example the automaton represented in Figure 1 recognizes the string *baaa* with probability $0.75 \times 1.0 \times 0.2 \times 0.2 \times 0.6 = 0.018$.

2.2 Learning Algorithms

A lot of algorithms have been proposed to infer PDFA from examples [4, 5, 7–9]. Most of them follow the same scheme based on state merging and summarized in Algorithm 1. Given a set of positive examples S_+, the algorithm first builds the probabilistic prefix tree acceptor (PPTA). The PPTA is an automaton accepting all the examples of S_+ (see left part of Figure 2 for an example, λ corresponding to the empty string). It is constructed such that the states corresponding to common prefixes are merged and such that each state and each transition is associated with the number of times it is used while parsing the learning set. This number is then used to define the function p. If $C(q)$ is the number of times a state q is used while parsing S_+, and $C(q, a)$ is the number of times the transition (q, a) is used while parsing S_+, then $p(q, a) = \frac{C(q,a)}{C(q)}$. Similarly, if $C_f(q)$ is the number of times q is used as final state in S_+ for each state q, we have $F(q) = \frac{C_f(q)}{C(q)}$.

The second step of the algorithm consists in running through the $PPTA$ (function *choose_states*(A)), and testing whether the considered states are statistically *compatible* (function *compatible*(q_i, q_j, α)). Several consecutive merging

Data: S_+ training examples (strings)
Result: A a PDFA
begin
  $A \leftarrow$ build_PPTA(S_+);
  while $(q_i, q_j) \leftarrow choose\_states(A)$ **do**
    if $compatible(q_i, q_j, \alpha)$ **then** $merge(A, q_i, q_j)$;
  end
  return A;
end

Algorithm 1. Generic algorithm for inferring PDFA

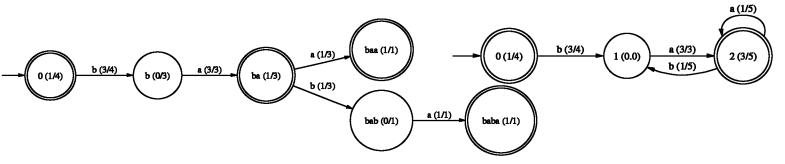

Fig. 2. PPTA of $S_+ = \{ba, baa, baba, \lambda\}$ on the left. On the right the PDFA obtained after two state mergings

operations are done in order to keep the automaton structurally deterministic. The algorithm stops when no more merging is possible. For example, the right part of Figure 2 represents the merging of the states labeled b and bab, and the merging of the three states labeled ba, baa, $baba$ from the PPTA on the left part of the figure.

2.3 Compatibility in the Algorithm ALERGIA

In ALERGIA [5], the compatibility of two states depends on: (i) the compatibility of their outgoing probabilities on the same letter, (ii) the compatibility of their probabilities to be final and (iii) the recursive compatibility of their successors.

Definition 2 *Two states q_1, q_2 are compatible iff: (i)* $\forall a \in \Sigma \left| \frac{C(q1,a)}{C(q1)} - \frac{C(q2,a)}{C(q2)} \right|$ *is not significantly higher than* 0*, (ii)* $\left| \frac{C_f(q1)}{C(q1)} - \frac{C_f(q2)}{C(q2)} \right|$ *is not significantly higher than* 0*, (iii) the two previous conditions are recursively satisfied for all the states reachable from* $(q1, q2)$*.*

The notion of significance is statistically assessed in ALERGIA. It consists in comparing the deviation between two proportions: $\left|\frac{x1}{n1} - \frac{x2}{n2}\right|$, where $n1 = C(q_1)$, $n2 = C(q_2)$, $x1$ equals either $C(q_1, a)$ or $C_f(q_1)$ and $x2$ either $C(q_2, a)$ or $C_f(q_2)$ $(a \in \Sigma)$.

The test of compatibility is derived from the Hœffding bound [6]. This bound is used to define a probability on the estimation error of a Bernoulli variable p estimated by the quantity $\frac{x}{n}$, which is a frequency observed over n trials.

$$P\left(\left|p - \frac{x}{n}\right| < \left(\sqrt{\frac{1}{2}\ln\frac{2}{\alpha}}\right) * \frac{1}{\sqrt{n}}\right) > 1 - \alpha$$

Since ALERGIA takes into account two frequencies ($\frac{x1}{n1}$ and $\frac{x2}{n2}$), it must add two estimation errors that assesses, in a way, the worst possible case.

Definition 3 *Two proportions are compatible in* ALERGIA *iff:*

$$\left|\frac{x1}{n1} - \frac{x2}{n2}\right| < \sqrt{\frac{1}{2}\ln\frac{2}{\alpha}}\left(\frac{1}{\sqrt{n1}} + \frac{1}{\sqrt{n2}}\right) \quad (1)$$

Despite the fact that this upper-bound is statistically correct, we can note that by adding two estimation errors, the test tends to often accept a state merging. Consequently, the probability to wrongly accept a merging (risk of second type β) is under-estimated, that can have dramatic effects on the final automata, particularly in the presence of noise. Moreover, the asymptotic bound introduced in inequality (1) is only relevant for large samples. In order to overcome this drawback, Kermorvant and Dupont have proposed MALERGIA [7] for dealing with small datasets.

2.4 Compatibility in the Algorithm MALERGIA

In MALERGIA, each state of the automaton is associated with a multinomial distribution modeling the outgoing transition probabilities and the final probability. In other words, each state is associated with a multinomial random variable with parameters $\tau = \{\tau_1, \ldots, \tau_K\}$, each τ_i corresponding to the transition probability of the i^{th} letter of the alphabet including a special final state symbol. In the PPTA each state is seen as a realization of the multinomial random variable τ (see [7] for more details). Two states are merged if they are both a realization of the same multinomial random variable. A statistical test following asymptotically a Khi-square distribution is used. When the constraints of approximation are not verified (*i.e.* for very small datasets), a Fisher exact test is used. However, in MALERGIA, this test requires the estimation of the probability of all contingency tables of size $2 \times K$ of the same marginal counts, that results in a very high complexity of the algorithm.

3 A New Compatibility Test Based on Proportions

In this section, we propose a new statistical approach overcoming the drawbacks of ALERGIA and MALERGIA and particularly relevant in presence of noise.

3.1 Statistical Framework

We use here a test of proportion comparison. It aims at comparing the proportions $\frac{x1}{n1}$ and $\frac{x2}{n2}$ (the same as those used in ALERGIA), estimators of the probabilities $p1$ and $p2$, and testing the hypothesis: $H_0 : p1 = p2$ versus $H_a : p1 \neq p2$. We compute the statistic:

$$Z = \frac{\frac{x1}{n1} - \frac{x2}{n2} - (p1 - p2)}{\sqrt{\hat{p}\hat{q}\frac{(n1+n2)}{n1n2}}} \qquad \text{where } \hat{p} = 1 - \hat{q} = \frac{x1 + x2}{n1 + n2}$$

Z approximately follows the normal distribution when H_0 is true. We reject H_0 in favor of H_a whenever $|Z| > z_{\alpha/2}$ where $z_{\alpha/2}$ is the $(1 - \alpha/2)$-percentile of the normal distribution.

Then we consider that two proportions are not statistically different if:

$$\left|\frac{x1}{n1} - \frac{x2}{n2}\right| < z_{\frac{\alpha}{2}} * \sqrt{\hat{p}\hat{q}\frac{(n1+n2)}{n1 * n2}}$$

Note that the constraints of approximation are satisfied when $n1 + n2 > 20$ or when $n1 + n2 > 40$ when either $x1$ or $x2$ is smaller than 5. When these conditions are not satisfied, we use a Fisher exact test, without the high calculation constraints of MAlergia.

3.2 Theoretical Comparison

We have seen before that the risk β is under-estimated in Alergia. We prove now that our test results in a more restrictive merging rule.

Theorem 1 $\forall \alpha < 0.734, \forall 0 < \alpha' \leq 1$:

$$z_{\frac{\alpha}{2}}\sqrt{\hat{p}\hat{q}\frac{(n1+n2)}{n1 * n2}} < \sqrt{\frac{1}{2}\ln(\frac{2}{\alpha'})}\left(\frac{1}{\sqrt{n1}} + \frac{1}{\sqrt{n2}}\right)$$

Proof. First we denote: $A = z_{\frac{\alpha}{2}}\sqrt{\hat{p}\hat{q}\frac{(n1+n2)}{n1*n2}}$ and $B = \sqrt{\frac{1}{2}\ln(\frac{2}{\alpha'})}(\frac{1}{\sqrt{n1}} + \frac{1}{\sqrt{n2}})$.

Since $\hat{p} \leq 1$ and $\hat{q} \leq 1$ and so $\sqrt{\hat{p}\hat{q}} \leq 1$, then we can deduce that:

$$A < z_{\frac{\alpha}{2}}\sqrt{\frac{(n1+n2)}{n1 * n2}} = z_{\frac{\alpha}{2}}\sqrt{\frac{1}{n2} + \frac{1}{n1}}$$

Then if we choose $\alpha < 0.734$, then the $(1 - \frac{\alpha}{2})$-percentile of the standard normal distribution is lower than 0.34, thus: $z_{\frac{\alpha}{2}} < 0.34 < \frac{1}{2}\ln(2) < \sqrt{\frac{1}{2}\ln(2)}$.

Moreover, for all $0 < \alpha' \leq 1$, $\ln(2) < \ln(\frac{2}{\alpha'})$, then

$$A \leq \sqrt{\frac{1}{2}\ln(\frac{2}{\alpha'})}\sqrt{(\frac{1}{n1} + \frac{1}{n2})}$$

and then since $\frac{1}{n2} + \frac{1}{n1} < \frac{1}{n1} + \frac{1}{n2} + \frac{2}{\sqrt{n1}\sqrt{n2}}$, for all $n1 > 0$ and $n2 > 0$, then $\sqrt{\frac{1}{n2} + \frac{1}{n1}} < \frac{1}{\sqrt{n1}} + \frac{1}{\sqrt{n2}}$ and we conclude that:

$$A \leq \sqrt{\frac{1}{2}\ln(\frac{2}{\alpha'})}(\frac{1}{\sqrt{n1}} + \frac{1}{\sqrt{n2}})$$

□

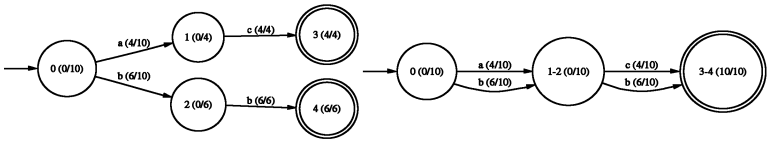

Fig. 3. Effect of a bad merging from a PPTA built with 4 strings ac and 6 strings bb

Assuming that we rarely build statistical tests with α higher than 0.5, the condition $\alpha < 0.734$ is not too constraining. The first direct consequence of this theorem is that our new merging rule is more restrictive, limiting the impact of potential noisy data. Secondly, such a rule tends to infer larger automata, both in the number of states and in the number of transitions. This situation can seem paradoxical. Actually, according to the theory of the learnable, and particularly in exact grammatical inference, too large automata tend to overfit the data resulting in a decrease of the generalization ability. This is true in exact grammatical inference, when we have both positive and negative examples, and when the goal consists in building a classifier which can predict, via a final state, the label of a new example. In this case, one must relax the merging constraint in the presence of noise, to allow a legitimate merging. Then we aim at inferring an automaton as small as possible to reduce the complexity of the model and then its VC-dimension. The problem seems to be different in probabilistic grammatical inference, where the inferred automaton is only able to provide a probability distribution. The error imputable to the automaton can come not only from an over-estimation but also from an under-estimation of the probability density. In this case, what is the consequence of a wrongly accepted merging due, for example, to the presence of noise? Figure 3 shows an explicit example. Before the merging, the probability of a string ac is $0.4 * 1 * 1 = 0.4$ and 0 for a string ab. Assume that a "bad" merging (of the states 1 and 2) is accepted, ac becomes under-estimated ($0.4 * 0.4 * 1 = 0.16$) and ab becomes more probable than ac ($0.4 * 0.6 * 1 = 0.24$). This example shows that, particularly in the presence of noise but also in noise-free situations, we must reduce the risk β, resulting in the rejection of some mergings, and then in the inference of larger automata.

Thus, we think that the use of a more specific and restrictive test is more relevant for dealing with noise. In MAlergia, Kermorvant and Dupont empirically note that their merging rule is also more restrictive. However, we think that it is not sufficient. Actually in the multinomial approach, the frequencies of a noisy transition can be absorbed by the global aspect of the test. In our proportion-based test, the merging rule is applied on each transition allowing us to better detect differences between the two tested states. Our proportion test also works with small samples and thus has not the problem of the asymptotic Hœffding bound. For very small samples, a Fisher test is used. While, in the multinomial approach, the number of contingency tables to consider increases exponentially with the size of the alphabet (K), in our framework, we only consider tables of a constant size 2×2. We reduce then the complexity of the test.

4 Evaluation in the Context of Noisy Data

We compare, in this section, automata inferred with the test based on the Hœffding bound, those obtained with the multinomial approach and those obtained with the proportion-based test, in two types of situations. The first one deals with cases where the target automaton is *a priori* known. In this case we can measure the distance between the inferred automata and the target automaton. However we do not know always this one. In this case, we evaluate the merging rules in another series of experiments using a perplexity measure. This criterion assesses the relevance of the model on a test sample. In order to show the effectiveness of our approach, experiments were done on two types of data, strings and trees.

4.1 Evaluation Criteria

Distance from the target automaton: [10] defined distances between two hidden Markov models introducing the co-emission probability, that is the probability that two independent models generate the same string. The co-emission probability of two stochastic automata $M1$ and $M2$, is denoted $A(M1, M2)$ and defined as follows: $A(M1, M2) = \sum_{s \in \Sigma^*} P_{M1}(s) * P_{M2}(s)$. Where $P_{Mi}(s)$ is the probability of s given the model Mi. The co-emission probability allows us to define a distance D_a between two automata $M1$ et $M2$:

$$D_a(M1, M2) = \arccos\left(\frac{A(M1,M2)}{\sqrt{A(M1,M1)*A(M2,M2)}}\right)$$

$D_a(M1, M2)$ can be interpreted as the measure of the angle between two vectors representing the automata $M1$, $M2$ in a space where the base is the set of strings of Σ^*.

Perplexity measure: When the target automaton is not known, the quality of an inferred model M can be evaluated by the average likelihood on a set of strings S relatively to the distribution defined by M:

$$LL = \frac{1}{\|S\|} \sum_{j=1}^{|S|} \log P_M(s_j)$$

where $P_M(s_j)$ defines the probability of the j^{th} string of S according to M. A perfect model can predict each element of the sample with a probability equal to one, and so $LL = 0$. In a general way we consider the perplexity of the test set which is defined by $PP = 2^{LL}$. A minimal perplexity ($PP = 1$) is reached when the model can predict each element of the test sample. Therefore we consider that a model is more predictive than another if its perplexity is lower.

4.2 Experimentations on Strings

Recall that our objective is to study the behavior of the three merging rules in the context of noisy data. To corrupt our training file, we replace a proportion γ (from 0.01 to 0.30) of letters of the training strings by a different letter randomly chosen in the alphabet. For each level of noise, we use several α parameters from

Base	Size	H	P	M	Sig
Reber D_a	3000	0.20 ± 0.153	0.16 ± 0.12	0.177± 0.13	yes
Reber P_e	3000	1.76± 0.14	1.74 ±0.13	1.75 ±0.13	yes
ATIS P_e	$\simeq$ 7000	92.4± 11.58	62.7±6.25	64.4±9.49	yes
Agaricus + P_e	4208	2.23 ± 0.80	1.86±0.37	1.92±0.48	yes
Agaricus − P_e	3918	2.64±1.21	2.06±0.52	2.13±0.60	yes
Badges + P_e	210	24.6 ±2.51	22.3±2.19	20.0±2.95	yes
Badges − P_e	120	27.3±2.6	24.3±2.31	20.5±3.11	yes
Promoters + P_e	56	3.80±0.07	3.93±0.05	3.91±0.16	no for **P** vs **M**
Promoters − P_e	56	2.61±0.79	2.79±0.62	2.47±0.96	yes

Fig. 4. Results on databases of strings. Yes in the column **Sig** means that all the deviations between **H** and **P**, **H** and **M** and **P** and **M** are significant, otherwise we indicate which deviation is not significant

0.0001 to 0.1. The results presented in this section correspond for each approach to the optimal α, that is the one which provides the smaller evaluation measure. Since we use different levels of noise, the results are presented for each dataset by the mean ± the standard deviation. We test the significance of our results using a Student paired t-test with a first oder risk of 5%. In the presentation of our results, those concerning the Hœffding test end with **H**, those for the proportion one with **P**, and those for the multinomial approach end with **M**. We indicate results obtained with D_a for the distance and P_e for the perplexity. The column **Sig** indicates the significance of the results.

We use a first database for which the target automaton is *a priori* known. This one represents the Reber grammar [11]. When the target is unknown, we suppose to have a training set and a test set. Only the first one contains noisy data. We evaluate the perplexity measure on the test set. We use here eight databases: a sample generated from the Reber grammar; the ATIS database [12]; and three databases of the UCI repository [13]: Agaricus, Badges and Promoters. For these three bases, we consider positive and negative examples as two different concepts to learn. We use a 5-folds cross validation procedure for all the databases, except for the ATIS one which already contains a training and a test set, and for which we use different sizes of the training set (from 1000 to 13044).

The results of the experiments are synthesized on Figure 4. Globally, and independently on the complexity costs, which are highly in favor of our test, the merging rules based on our proportion test and on the multinomial test provide better results than ALERGIA, except for Promoters. This result can be explained by the relatively small size of the sample. Globally, the multinomial test works better than our approach on small datasets (Badges, Promoters), this fact confirms the original motivation of MALERGIA. However when the size of the training set grows, the proportion based-test is better (Reber, ATIS, Agaricus). Considering the level of noise, we noted that the results are highly in favor of our approach, particularly when the noise is higher than 8%. This behavior on the database Agaricus is shown on Figure 5. Note that the difference between the two approaches increases with the level of noise.

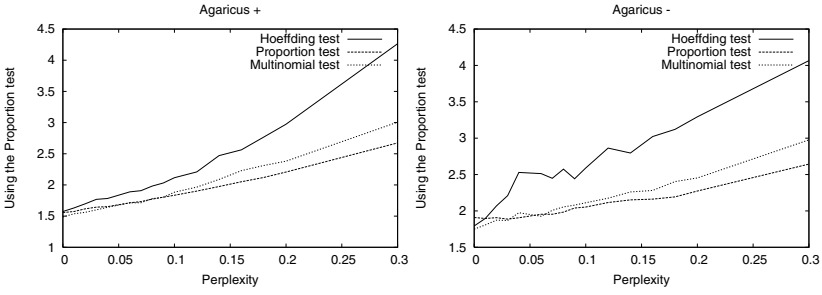

Fig. 5. Behavior of the merging rules on Agaricus *w.r.t.* different levels of noise

4.3 Experimentations on Trees

Since the interest about tree-structured data is increasing, notably because of their huge availability on the web, we also propose to evaluate our extension of ALERGIA to stochastic tree automata [14] (note that we consider bottom-up tree automata). The multinomial approach is not compared here because its adaption to bottom-up tree automata is not trivial.

Stochastic Tree Automata (STA): Tree automata [15] define a regular language on trees as a PDFA defines a regular language on strings. Stochastic tree automata are an extension of tree automata, defining a probability distribution on the tree language defined by the automaton. We use an extension of these automata taking into account the notion of type: stochastic many-sorted tree automata defined on a signature. We do not detail here these automata and their learning method. The interested reader can refer to [14, 16]. We only precise that a learned stochastic tree automaton allows to define a probability distribution on trees recognized by the automaton. In the context of trees, we change a proportion γ of leaves in order to corrupt the learning set.

Experiments: We use three target grammars, one concerning stacks of objects, one on boolean expressions and another artificial dataset Art2. From each grammar we generate a sample of trees. We keep the same protocol as presented for experiments on strings. For cases where the target automaton is unknown, we use five datasets. We take a sample from each of the three previous tree grammars. Then we also use the database exploited for the PKDD'02 discovery challenge[1] (converted in trees as described in [17]). Finally, we treat the database Student Loan of the UCI repository, converting prolog facts in trees as describes in [18]. The results of the two series of experiments are presented on Figure 6. Experimentations on trees confirm the results observed on strings. Automata obtained using the proportion test are better with a lower standard deviation than those inferred with the test based on the Hœffding bound.

[1] http://lisp.vse.cz/challenge/ecmlpkdd2002/

Base	Size	D_a **H**	D_a **P**	Sig
Stacks D_a	3000	0.241 ± 0.164	0.225±0.17	yes
Art2 D_a	3000	0.555 ± 0.138	0.190±0.1	yes
Bool. D_a	5000	0.1 ± 0.049	0.096±0.046	yes
Stacks P_e	3000	1.85 ± 0.056	1.78±0.063	yes
Art2 P_e	3000	3.68 ± 0.45	3.21±0.21	yes
Bool. P_e	4000	2.60 ± 0.026	2.45±0.01	yes
PKDD'02 P_e	4178	6.90 ± 1.99	1.94 ±0.14	yes
Student Loan P_e	800	5.09± 1.48	2.88 ±0.26	yes

Fig. 6. Results for trees on the 8 databases

5 Conclusion

In this paper, we addressed the problem of dealing with noise in probabilistic grammatical inference. As far as we know, this problem has never been studied but seems very important because of the wide range of applications it is related to. Since the main objective in the probabilistic grammatical inference framework is to correctly estimate the probability distribution of the examples, we brought out the paradoxical fact that larger automata deal better with noise than more general (smaller) ones. We studied this behavior in the context of state merging algorithms and gave the intuitive idea that a bad merging, due to the presence of noise, could lead to a very bad estimation of the target distribution. Consequently we propose to use a restrictive statistical test during the inference process. Practically, we have proposed to replace the initial statistical test of the ALERGIA algorithm by a more restrictive one based on proportion comparison. We have proved its restrictiveness and shown its interest, in the context of noisy data both on artificial and real datasets.

While our approach deals better with noise, we have empirically noticed, in noise-free situations, that the results are quite similar with those of ALERGIA. We have also compared our test with the multinomial approach used in MALERGIA. Our proportion-based test is not only relevant, in terms of complexity and perplexity, on small and large datasets, but also provide better results for high level of noise.

We are currently working on theoretical aspects of our work. We aim at proving that the acceptance of a bad merging, especially in the context of noisy data, implies a larger deviation from the target distribution than its rejection.

Acknowledgements

The authors wish to thank Christopher Kermorvant for his help and for having allowed us to easily compare our work with MALERGIA. We also want to thank Thierry Murgue for his help and for his experience in the evaluation of PFSA.

References

1. Brodley, C., Friedl, M.: Identifying and eliminating mislabeled training instances. In: Thirteeth National Conference on Artificial Intelligence AAAI/IAAI, Vol. 1. (1996) 799–805
2. John, G., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: 11th International Conference on Machine Learning. (1994) 121–129
3. Honavar, V., de la Higuera, C.: Introduction. Machine Learning Journal **44** (2001) 5–7
4. Carrasco, R., Oncina, J.: Learning stochastic regular grammars by means of a state merging method. In: Grammatical Inference and Applications, ICGI'94. Number 862 in LNAI, Springer Verlag (1994) 139–150
5. Carrasco, R., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. RAIRO (Theoretical Informatics and Applications) **33** (1999) 1–20
6. Hoeffding, W.: Probabilities inequalities for sums or bounded random variables. Journal of the American Association **58** (1963) 13–30
7. Kermorvant, C., Dupont, P.: Stochastic grammatical inference with multinomial tests. In: Proceedings of the Sixth International Colloquium on Grammatical Inference (ICGI). Volume 2484 of LNAI., Amsterdam, Springer (2002) 149–160
8. Ron, D., Singer, Y., Tishby, N.: On the learnability and usage of acyclic probabilistic automata. In: Computational Learning Theory, COLT'95. (1995) 31–40
9. Thollard, F., Dupont, P., de la Higuera, C.: Probabilistic dfa inference using kullback–leibler divergence and minimality. In Kauffman, M., ed.: Proceedings of the Seventeenth International Conference on Machine Learning. (2000) 975–982
10. Lyngsø, R., Pedersen, C., Nielsen, H.: Metrics and similarity measures for hidden Markov models. In: 7th International Conference on Intelligent Systems for Molecular Biology, ISMB '99 Proceedings, Heidelberg, Germany, AAAI Press, Menlo Park, CA94025, USA (1999) 178–186
11. Reber, A.: Implicit learning of artificial grammars. Journal of verbal learning and verbal behaviour **6** (1967) 855–863
12. Hirschman, L., Bates, M., Dahl, D., Fisher, W., Garofolo, J., Hunicke-Smith, K., Pallett, D., Pao, C., Price, P., Rudnicky, A.: Multi-site data collection for a spoken language corpus. In: Proc. DARPA Speech and Natural Language Workshop '92, Harriman, New York (1992) 7–14
13. Blake, C., Merz, C.: University of California Irvine repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/` (1998)
14. Habrard, A., Bernard, M., Jacquenet, F.: Generalized stochastic tree automata for multi-relational data mining. In: Proceedings of the Sixth International Colloquium on Grammatical Inference (ICGI). Volume 2484 of LNAI., Amsterdam, Springer (2002) 120–133
15. Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree Automata Techniques and Applications . Available on: `http://www.grappa.univ-lille3.fr/tata` (1997)
16. Carrasco, R., Oncina, J., Calera-Rubio, J.: Stochastic Inference of Regular Tree Languages. Machine Learning **44** (2001) 185–197
17. Habrard, A., Bernard, M., Jaquenet, F.: Mining probabilistic tree patterns in a medical database. Discovery Challenge of the 6th Conference PKDD'02 (2002)
18. Bernard, M., Habrard, A.: Learning stochastic logic programs. In Rouveirol, C., Sebag, M., eds.: Work-in-Progress Track at the 11th International Conference on Inductive Logic Programming. (2001) 19–26