Correction of Uniformly Noisy Distributions to Improve Probabilistic Grammatical Inference Algorithms
Amaury Habrard, Marc Bernard, Marc Sebban

To cite this version:

HAL Id: ujm-00378062
https://hal-ujm.archives-ouvertes.fr/ujm-00378062
Submitted on 23 Apr 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Correction of Uniformly Noisy Distributions to Improve Probabilistic Grammatical Inference Algorithms

Amaury Habrard and 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 aim at correcting distributions of noisy samples in order to improve the inference of probabilistic automata. Rather than definitively removing corrupted examples before the learning process, we propose a technique, based on statistical estimates and linear regression, for correcting the probabilistic prefix tree automaton (PPTA). It requires a human expertise to correct only a small sample of data, selected in order to estimate the noise level. This statistical information permits us to automatically correct the whole PPTA and then to infer better models from a generalization point of view. After a theoretical analysis of the noise impact, we present a large experimental study on several datasets.

Keywords. probabilistic grammatical inference, noisy data, distribution correcting

Introduction

Nowadays, machine learning algorithms have to process huge amount of data. Beyond this algorithmic constraint, they also have to deal with an important presence of noise, due to different reasons such as typing, transfer or experimental errors. To tackle these problems, a lot of data reduction techniques have been proposed to remove, either noisy examples (usually called prototype selection (Brodley & Friedl 1996)), or noisy features (called feature selection (John, Kohavi, & Pfleger 1994)). These approaches require generally not only positive but also negative instances of the concept to learn and use a discrimination criterion (entropy measures (Sebban & Nock 2000), classification success rate (Brodley & Friedl 1996), etc.) to detect the presence of noise. In this context, a noisy example is considered as a positive (resp. negative) instance which should be negatively (resp. positively) labeled in the absence of noise. Unfortunately, we often do not have access to negative examples. In natural language processing, for example, it is hard to enumerate a lot of counter-examples of the language. Thus, in this context, standard data reduction techniques are totally unsuitable when only positive examples are available.

Algorithms that learn only from positive data usually use statistical information in the learning sample to generate a model defining a probability distribution on positive data. Such approaches are usually called stochastic methods. In this context, a current trend is to consider that a noisy instance is a weakly probable example, not representative of the underlying distribution. Indeed, if such data (usually called outliers) would cover a significant part of the probability density function, they would become a relevant part of the concept to learn. We can then think that a weakly probable example is due to the presence of noise, or at least to a weak relevance for characterizing the target concept. Then, it would be reasonable to remove it, as it has already been done efficiently for tree-structured data in (Habrard, Bernard, & Sebban 2003b). However, as we mentioned before, stochastic models require the estimation of probabilities and then large learning sets. In such a context, we think that a distribution correction procedure is more suited than a data reduction technique aiming at definitively deleting noisy instances. Moreover, noise can not only generate outliers, but also modify the probability of some examples that characterize the concept to learn. These consequences can be dramatic for stochastic learning algorithms, such as probabilistic grammatical inference algorithms.

Grammatical inference (de la Higuera 2005) is a domain of machine learning whose goal is to learn language models from a set of sequences. Probabilistic grammatical inference aims at learning a probabilistic automaton which defines a distribution on the language recognized by this automaton. Learning data (only positive) are supposed to have been generated following a theoretical probability distribution. The goal is then to learn the target automaton which has generated the data. We focus here, more specifically, on algorithms based on state merging techniques (Carrasco & Oncina 1994; Thollard, Dupont, & de la Higuera 2000). They generally use the same base principle consisting in building, during a first step, a specific probabilistic automaton (called the PPTA for Probabilistic Prefix Tree Acceptor) recognizing only the learning examples. Then, for allowing generalization performances, the inference process is achieved by iteratively merging states considered as equivalent from a statistical point of view.

In this article, we focus on learning probabilistic automata from uniformly noisy data. Our objective is to correct this
Probabilistic Finite State Automata (PFSA) are a probabilistic extension of Finite State Automata and define a probability distribution on strings.

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

We only consider here deterministic PFSA (called PDFA). That 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 \(\Sigma\) (the set of all strings built on \(\Sigma\)), each state of \(Q\) has to be able to generate a string with a strictly positive probability and, \(p\) and \(F\) must satisfy the following consistency constraint:

\[
\forall q \in Q, F(q) + \sum_{a \in \Sigma} p(q, a) = 1.
\]

A string \(s_0 \ldots s_{n-1}\) is accepted by an automaton \(A\) iff there exists a sequence of states \(e_0 \ldots e_n\) 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: 

\[
P_A(s_0 \ldots s_{n-1}) = \left(\prod_{i=0}^{n-1} p(e_i, s_i)\right) F(e_n).
\]

The second step of the algorithm consists in running through the PPTA (function \(\text{choose\_states}(A)\)), and testing whether the considered states are statistically compatible (function \(\text{compatible}(q_i, q_j)\)). Several consecutive merging operations are done to keep the automaton structurally deterministic. The algorithm stops when no more merging is possible. For example, the automaton at the bottom part of Figure 2 represents the merging of the states labeled \(b\) and \(bab\), and the ones labeled \(ba\), \(baa\), \(bba\).

Theoretical Analysis in the Presence of Noise

Let us recall that our goal is to correct the PPTA constructed from noisy data with the help of a manually corrected sample. In this section, we study the impact of noise on the PPTA transition probabilities. We also show that the alphabet size has a direct influence on these probabilities. We assume that the noise is uniformly distributed on the letters of the learning sample, which corresponds to a reasonable hypothesis in front of the different possible types of errors.
Probabilistic automata learning is based on the frequencies of transitions and states. The presence of noise can highly disturb the estimation of some transitions and have a negative impact on the quality of the inferred automata. To tackle this drawback, we propose to correct the transition probabilities of the PPTA with the help of a small manually corrected sample. We already argued about the interest of such an approach. An important remark can be made here. While the corrected sample will help us to estimate the noise level, this sample can not, in any case, be used as a learning sample to infer a PPTA. Indeed, statistically speaking, this sample would be too small to estimate accurately the whole set of transitions, which would represent a too high number of parameters. The estimation of the single parameter \( \gamma \) is, on the other hand, more reasonable. In this context, we begin to show how we exploit the corrected sample to estimate \( \gamma \), before presenting how we use it to correct the PPTA.

### Noise Estimation by Linear Regression

Let us consider two PPTA: \( A_b = (Q_b, \Sigma, \delta_b, q^0_b, F_b) \) built from the noisy sample, and \( A_w = (Q_w, \Sigma, \delta_w, q^0_w, F_w) \) from the manually corrected one. Let \( q_b \in Q_b \) a state of \( A_b \), \( w \in \Sigma^* \) such that \( \delta_b(q^0_b, w) = q_b \) and \( q_w \in Q_w \) such that \( \delta_w(q^0_w, w) = q_w \). \( q_w \) is the state recognizing the same prefix as \( q_b \). We assume that \( C(q_w) = n_w \) and \( C(q_b) = n_b \). We consider the following set of pairs of transition probabilities: \( E_{\text{init}} = \{(p_{\Sigma}(q_w, a), p_{\Sigma}(q_b, a))|a \in \Sigma\} \).

Each pair of reals defines a point in \([0; 1]^2\). According to Equation 3, there is a linear dependence between the values of the probabilities of noisy transitions and noise-free ones, defined by an equation of the form \( y = C + x + d \). 

**Cox (1987):** 
\[
C = \frac{\sum_{a \in \Sigma}(p_{\Sigma}(q_w, a) - \frac{1}{|\Sigma|}) + (p_{\Sigma}(q_b, a) - \frac{1}{|\Sigma|})}{\sum_{a \in \Sigma}(p_{\Sigma}(q_w, a) - \frac{1}{|\Sigma|})^2}, \quad d = \frac{1}{|\Sigma|} - C + \frac{1}{|\Sigma|}.
\]
Using $C$ and Equation 3, we estimate $\gamma$ such that:

$$C = 1 - \frac{[\Sigma](\gamma)}{[\Sigma] - 1} \Rightarrow \gamma = \frac{([\Sigma] - 1) * (1 - C)}{[\Sigma]}$$

(4)

However, probability estimations of $A_\pi$ are accurate relatively to the size of the corrected sample. Rather than using these estimations, we propose to compute the bounds of a confidence interval which takes into account the size of $A_\pi$. Given an estimation $\hat{p}_\pi(q_\pi, a)$ of a probability $p_\pi(q_\pi, a)$ estimated from $n_\pi$ instances, a confidence interval, with a risk $\alpha$, around $p_\pi(q_\pi, a)$ is defined by:

$$\hat{p}_\pi(q_\pi, a) - u_{\alpha/2} \sqrt{\frac{\hat{p}_\pi(q_\pi, a) P_{\pi}(q_\pi, a)}{n_\pi}} < p_\pi(q_\pi, a) < \hat{p}_\pi(q_\pi, a) + u_{\alpha/2} \sqrt{\frac{\hat{p}_\pi(q_\pi, a) P_{\pi}(q_\pi, a)}{n_\pi}}$$

where $p_\pi(q_\pi, a) = 1 - \hat{p}_\pi(q_\pi, a)$, and $u_{\alpha/2}$ corresponds to the $(1 - \alpha/2)$ percentile of the normal distribution.

We compute this interval for each couple of transition probabilities. We obtain two sets, $E_{inf}$ corresponding to the pairs of transitions of the lower bands of each interval, and $E_{sup}$ to the upper ones:

$$E_{inf} = \{(\hat{p}_\pi(q_\pi, a) - \hat{p}_\pi(q_\pi, a) | a) \in \Sigma\},$$
$$E_{sup} = \{(\hat{p}_\pi(q_\pi, a), \hat{p}_\pi(q_\pi, a) + \hat{p}_\pi(q_\pi, a) | a) \in \Sigma\}.$$  

Where $\hat{p}_\pi(q_\pi, a) - \hat{p}_\pi(q_\pi, a) = p(x, a) - u_{\alpha/2} \sqrt{\frac{p(x, a)(1 - p(x, a))}{n_x} + \frac{\hat{p}_\pi(q_\pi, a)(1 - \hat{p}_\pi(q_\pi, a))}{n_x}}$  

and $\hat{p}_\pi(q_\pi, a) + \hat{p}_\pi(q_\pi, a) = p(x, a) + u_{\alpha/2} \sqrt{\frac{p(x, a)(1 - p(x, a))}{n_x} + \frac{\hat{p}_\pi(q_\pi, a)(1 - \hat{p}_\pi(q_\pi, a))}{n_x}}$.

We use each of the sets of points in order to obtain two lines of regression $y = C_{sup} x + d_{sup}$ and $y = C_{inf} x + d_{inf}$ defining the minimal and maximal values for $\gamma$ that can be obtained, taking into account the sizes of both of the noisy and corrected samples. Finally, we estimate $\gamma$, according to Equation 4, with the slope of the bisecting of these two lines, defined by the equation:

$$\frac{C_{inf} x + d_{inf} - C_{sup} x - d_{sup}}{\sqrt{C_{inf} x + d_{inf} + \frac{1}{C_{inf} x + d_{inf}}}} = \frac{y - C_{inf} x - d_{inf}}{\sqrt{C_{inf} x + d_{inf} + \frac{1}{C_{inf} x + d_{inf}}}}$$

Let us consider an example to illustrate our approach. Let $\Sigma = \{a, b, c\}$ and $\gamma = 0.25$. Figure 3(a) indicates the transition probabilities for the two states $q_\pi$ and $q_\pi$ estimated from 100 and 1000 examples. Then, we compute, using a risk $\alpha = 0.10$, the sets $E_{sup} = \{(0.1494, 0.2019), (0.48, 0.393), (0.583, 0.456)\}$ and $E_{inf} = \{(0.0505, 0.173), (0.319, 0.357), (0.418, 0.149)\}$. Using these two samples, we compute the two regression lines, respectively $D_{sup}$ and $D_{inf}$, and their bisecting line $D_{bissec}$. The result is presented on Figure 3(b). The slope of the bisecting line is $0.652376$ and provides an estimation $\hat{\gamma} = 0.25$. This estimation is relatively closed to the theoretical value $\gamma$, despite the fact that less than 10% of the examples have been corrected.

**Correcting the PPTA**

According to Equation 2, a noisy transition probability evolves linearly with $\gamma$. After estimating this level of noise, we can correct the probability $p_\pi(q_\pi, a)$ by:

$$p_\pi(q_\pi, a) = \frac{([\Sigma] - 1)p_\pi(q_\pi, a) - (1 - F(q_\pi))\gamma}{([\Sigma] - 1 - \gamma)\Sigma}$$

More precisely, rather than correcting probabilities, we correct the frequencies:

$$C(q_\pi, a) = \frac{([\Sigma] - 1)c(q_\pi, a) - (1 - [\Sigma] - C(q_\pi, a))\gamma}{([\Sigma] - 1 - \gamma)\Sigma}$$

Note that a given correction has an effect on the states accessible from the corrected transitions. We need then to take into account these changes in the PPTA. We use a breadth first search of the PPTA to guarantee an efficient correction. Formally, for each state $q$ that recognizes a string coming from states directly accessible from $q_\pi$, the outgoing transitions of $q$ are modified as follows, $\forall w \in \Sigma$, $\forall a \in \Sigma$ such that $\exists q \in Q$, $\delta'(q, b, a) = q$, $\forall w \in \Sigma \cup \{\#\}$:

$$C'(\delta'(q, b, a), w, e) = \frac{([\Sigma] - 1)c'(\delta'(q, b, a), w, e) - \gamma N_{w,e}}{([\Sigma] - 1 - \gamma)\Sigma}$$

where $N_{w,e} = \sum_{a \in \Sigma} C(\delta(q, b, a), w, e)$ represents the amount of letters $e$, accessible from the string $w$, that can be affected by the correction of $q_\pi$, and $C(q, \#) = C'(q, \#)$. We summarize our approach in Algorithm 2.

**Algorithm 2:** Algorithm for correcting probabilities.
Experimental Evaluation

In this section, we present a set of experiments allowing us to evaluate the relevance of our approach in the presence of noisy data. For this purpose, we use the algorithm Alergia (Carrasco & Oncina 1994), which is one of the best known state merging algorithms. We compare its performance with and without a PPTA correction, according to two criteria. The first one corresponds to the case where the target automaton is a priori known. In this case, we measure a distance between the inferred model and the target to evaluate the learned model quality. The second one corresponds to the case where the target automaton is unknown. In this context, we evaluate the inferred models with a perplexity measure on a test set.

Evaluation Criteria

Probabilistic Distance  Lyngsø et al. defined distances between two Hidden Markov Models introducing the co-emission probability, as the probability that two independent models generate the same string (Lyngsø, Pedersen, & Nielsen 1999). (Carrasco & Rico-Juan 2002) presents an adaptation of the co-emission to stochastic tree automata. The probability that two probabilistic models A and A’ generate the same strings is defined as: \( C(A, A’) = \sum_{S \subseteq \Sigma} P_A(s)P_A’(s) \), where \( P_A(s) \) is the probability of s given the model A. This co-emission probability allows us to define a probabilistic distance \( D_o \), which can be interpreted as the measure of the angle between two vectors of probabilities: probabilistic automata in a space where base is the set \( \Sigma^* \):

\[
D_o(A, A’) = \arccos\left( \frac{C(A, A’)}{\sqrt{C(A, A)C(A’, A’)} } \right)
\]

Perplexity  The quality of a probabilistic automaton A on a set of sequences S, can be evaluated by computing the mean of the log-likelihood of the elements of S relatively to the distribution defined by A: \( LL = \frac{1}{|S|} \sum_{s \in S} \log p(s | A) \), where \( p(s_j | A) \) is the probability of the \( j^{th} \) string of S. A perfect model can predict each element of the test set 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.

Experimentations

We carried out two series of experiments: one when the target is unknown and one when it is a priori known. For the first series, we used 3 datasets of the UCI Irvine (Blake & Merz 1998): AGARICUS, TICTACTOE and BADGES containing respectively 8126, 769 and 530 examples. Each base is composed of a positive and a negative set, used for classification tasks. We decided to use the negative sample as another concept to learn, which allows us to have 3 additional datasets. We also used a database of world first names beginning with the letter A (FIRSTNAME). This base has 1045 male first names (M) and 842 female ones (F). For the series with a known target model, we used three sources.

The Reber grammar (Reber 1967), from which we generated three samples: REBER1 with 500 examples, REBER2 with 5000 examples and REBER3 composed of 10000 examples. We also used two target automata recognizing the languages \( ac^a + ba^b \) (ART1) and \( ab^a + bb^b \) (ART2). We generated a sample of 10000 instances for ART1 and 2000 for ART2.

Our experimental set up consisted in adding noise to each learning sample, with different levels, from 1% to 50%. We then randomly chose a given proportion of examples to construct the sample manually corrected by the expert. This sample represents 25% of the original one for datasets having less than 400 examples, 10% for those having less than 2000 examples and 5% for the others, except for those with 10000 examples where it represents 2.5% of the original size. We applied our correction method, for each corrupted file, using a risk of 5% for the computation of confidence intervals. Finally, for each obtained file (noise-free, noisy and noisy, and corrected), we learned an automaton using the algorithm Alergia. The quality criterion is \( D_o \) for samples where the target automaton is known and the perplexity for the others. For this last measure, we used a 5 fold cross-validation. Moreover, in order to compare our ability to deal with noisy data, we compared our approach with the results obtained by the adaptation of Alergia proposed in (Habrard, Bernard, & Sebban 2003a) (denoted by \( PA \)) on noisy samples. This adaptation allows one to reduce the impact of noise using a more restrictive merging rule.

The results are presented in Table 1. The name of the samples is indicated in column Base and the quality measure \( (D_o \) for the distance or \( P_e \) for the perplexity) in column M. The optimal result, the one obtained with a whole noise-free sample, is indicated in column WN. Column N describes results obtained from the uncorrected files, column C those after our automatic correcting method, and column PA, those obtained with the adaptation PA. For these columns, we indicate the average of the results on all the levels of noise ± the standard deviation. In order to verify that inferring the automaton only from the small file corrected by the expert is not sufficient, we present in column U the results obtained in this context.

For all the databases, automata inferred from samples corrected with our method (C) are significantly better than

<table>
<thead>
<tr>
<th>Base</th>
<th>WN</th>
<th>M</th>
<th>N</th>
<th>C</th>
<th>U</th>
<th>PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGARICUS+</td>
<td>P1</td>
<td>1.6</td>
<td>4.1</td>
<td>2.9</td>
<td>3.3</td>
<td>0.4</td>
</tr>
<tr>
<td>AGARICUS+</td>
<td>P2</td>
<td>1.7</td>
<td>4.1</td>
<td>2.9</td>
<td>3.3</td>
<td>0.4</td>
</tr>
<tr>
<td>BADGES+</td>
<td>P1</td>
<td>25</td>
<td>29.4</td>
<td>28.6</td>
<td>0.1</td>
<td>32.1</td>
</tr>
<tr>
<td>BADGES+</td>
<td>P2</td>
<td>32.4</td>
<td>32.4</td>
<td>32.4</td>
<td>0.1</td>
<td>32.4</td>
</tr>
<tr>
<td>TICTACTOE+</td>
<td>P1</td>
<td>3.6</td>
<td>3.6</td>
<td>3.6</td>
<td>0.1</td>
<td>3.6</td>
</tr>
<tr>
<td>TICTACTOE-</td>
<td>P2</td>
<td>3.2</td>
<td>3.6</td>
<td>3.6</td>
<td>0.1</td>
<td>3.4</td>
</tr>
<tr>
<td>FIRSTNAME M</td>
<td>P1</td>
<td>15.2</td>
<td>20.9</td>
<td>23.1</td>
<td>0.1</td>
<td>16.2</td>
</tr>
<tr>
<td>FIRSTNAME M</td>
<td>P2</td>
<td>17.5</td>
<td>23.2</td>
<td>23.2</td>
<td>0.1</td>
<td>14.2</td>
</tr>
<tr>
<td>ART1</td>
<td>P1</td>
<td>1.7</td>
<td>2.0</td>
<td>0.3</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>ART2</td>
<td>P1</td>
<td>1.3</td>
<td>1.7</td>
<td>0.3</td>
<td>1.4</td>
<td>0.1</td>
</tr>
<tr>
<td>REBER1</td>
<td>P1</td>
<td>1.6</td>
<td>2.6</td>
<td>0.4</td>
<td>1.2</td>
<td>0.2</td>
</tr>
<tr>
<td>REBER1</td>
<td>P2</td>
<td>1.6</td>
<td>2.1</td>
<td>0.6</td>
<td>1.6</td>
<td>0.1</td>
</tr>
<tr>
<td>REBER2</td>
<td>P1</td>
<td>1.6</td>
<td>2.1</td>
<td>0.5</td>
<td>1.9</td>
<td>0.1</td>
</tr>
<tr>
<td>REBER2</td>
<td>P2</td>
<td>1.6</td>
<td>2.1</td>
<td>0.5</td>
<td>1.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Average</td>
<td>D_n</td>
<td>0.11</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
<td>0.1</td>
</tr>
</tbody>
</table>
those learned directly from the noisy samples (N). Moreover, our technique is better, on average, than the approach P.A which confirms its interest and relevance in the context of dealing with noise in probabilistic grammatical inference. We can also note that the standard deviation is significantly smaller, that denotes a good robustness of the approach. Moreover, our results confirm that the use of the corrected sample size, for C and U and a level of noise of 25%. The automaton, obtained after correction phase, is on average, better than those learned only from the manually corrected sample. An important point concerns the size of the manually corrected sample. On Figure 4, we represent the perplexity evolution on the base Reber1, relatively to the corrected sample size, for C and U and a level of noise of 25%. The automaton, obtained after correction, is better until a size representing less than 20% of the original sample, that corresponds to less than 100 examples. Beyond this point, the manually corrected sample is logically self-sufficient.

Figure 4: Impact of the corrected sample size.

Conclusion

In this paper, we proposed a technique allowing us to automatically correct a learning sample containing a uniformly distributed noise. Our approach is doubly interesting: first, we are able to correct the learning sample given the level of noise. Secondly, we provide a solution for estimating this level with the help of a small sample corrected by an expert. We experimentally showed the interest of our method in the context of learning probabilistic automaton from noisy data.

We aim at extending our approach to process other kinds of noise (such as Gaussian noise for example). While the estimation strategy of γ would not be changed, we have actually to define another functional relationship between values of the transition probabilities of the corrupted PPTA and those of the manually corrected one.

Another remark concerns the size of the corrected sample. When it is high, the correction is obviously efficient, from a statistical point of view. However, the correction of too large a sample, by a human expert, is not reasonable. Then, we would like to theoretically study the minimal size for an efficient correction.

References