Stochastic grammatical inference with multinomial tests

Christopher Kermorvant[1] and Pierre Dupont[2]

[1] EURISE, Université Jean Monnet, Saint-Etienne, France
kermorva@univ-st-etienne.fr
[2] INGI, University of Louvain, Louvain-la-Neuve, Belgium
pdupont@ingi.ucl.ac.be

Abstract.[3] We present a new statistical framework for stochastic grammatical inference algorithms based on a state merging strategy. We propose to use multinomial statistical tests to decide which states should be merged. This approach has three main advantages. First, since it is not based on asymptotic results, small sample case can be specifically dealt with. Second, all the probabilities associated to a state are included in a single test so that statistical evidence is cumulated. Third, a statistical score is associated to each possible merging operation and can be used for best-first strategy. Improvement over classical stochastic grammatical inference algorithm is shown on artificial data.

1 Introduction

The aim of stochastic regular grammatical inference is to learn a stochastic regular language from examples, mainly through learning the structure of a stochastic finite state automaton and estimating its transition probabilities. Several learning algorithms have been proposed to infer the structure of a stochastic automaton from a sample of words belonging to the target language [2, 13, 3, 14]. These algorithms are based on the same scheme: they start by building a tree which stores exactly the sample set and test possible state merging according to a fixed order. The two key points of these algorithms are the order in which the possible merging operations are evaluated and the compatibility function which evaluates whether or not two states can be merged.

The order chosen to evaluate candidate merging operations is usually hierarchical: states are ordered according to their depth in the tree and for a given depth according to the symbol labeling their incoming edge, the symbols of the alphabet being ordered according to an arbitrary order. Alternative strategies, named data-driven [5] or evidence-driven [7] have been proposed to explore the tree. However these strategies have been applied in a non probabilistic framework and are not directly dependent on the statistical significance of a merging operation.

[3] Published in *Proceedings of 6th International Colloquium on Grammatical Inference: Algorithms and Applications*, LNAI, No. 2484, Springer Verlag, pp. 149–160, Amsterdam, 2002.

Merging compatibility function which have been proposed are based either on local statistical tests [2, 13, 3] or on a global test [14]. This global test is derived from the Kullback-Leibler divergence and the merged states are chosen in order to balance the divergence from the sample distribution and the size of the inferred automaton. The global test has been shown to yield better results on a language modeling task, but its complexity is significantly higher than local tests. In the present work, we will restrict our attention to local tests.

Local compatibility tests evaluate a candidate pair of states to be merged according to two criteria:

- the compatibility of probabilities associated with outgoing transitions labeled by the same symbol
- the compatibility of states final probabilities

For a given transition, Carrasco and Oncina [2] proposed to use Hoeffding bounds to evaluate an upper bound of the difference between the transition probability estimated on the sample set and the theoretical probability. From this bound, they derive a compatibility criterion for two estimated transition probabilities. However, the proposed tests suffer from the following limitations:

- the transition compatibility test is derived from an asymptotic bound and is therefore designed for large sample sets. The case of small sample sets is not addressed whereas it is particularly important for real life applications.
- the state compatibility test is based on several independent transition compatibility tests, and no cumulated evidence is used.
- the compatibility test does not interact with the evaluation order, even if data or evidence based order is used.

Solutions were proposed only for the first limitation. In [13], states with too low frequencies are not considered during the inference and afterward are merged into special *low frequency* states. In [16], a statistical test is used to separate low frequency states, which are merged at the end of the inference according to heuristics.

We propose a new compatibility test based on a classical multinomial goodness of fit test. For each state, the set of outgoing transitions probabilities may be modeled by a multinomial probability distribution on the alphabet. In this framework, for each state, the set of observed frequencies for these transitions is the realization of a multinomial random variable. The compatibility test between two states is then a classical statistical test : given the observed transition frequencies, test the hypothesis that the two states share the same underlying multinomial probability distribution. The advantages of this approach are:

- small value frequencies can be specifically dealt with, using exact tests.
- all outgoing transitions probabilities are considered in a single test, such that evidence is cumulated.
- by normalizing the test score, several merging operations can be compared and the evaluation order can be modified accordingly.

The structure of this paper is as follows : first we recall the definition of probabilistic finite state automata and present the Alergia inference algorithm. Then we present a new multinomial framework for the inference algorithm and propose solutions for small sample case and evidence driven search strategy. Finally, the proposed tests and strategies are compared on artificial data.

2 Probabilistic Finite State Automaton

We consider probabilistic finite state automata (PFSA), which are a probabilistic extension of finite state automata. A PFSA $\mathcal{A}$ is defined by $< Q, \Sigma, \delta, \tau, q_0, F >$ where

- Q is a finite set of states
- Σ is the alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ is a transition function
- $\tau : Q \times \Sigma \rightarrow]0, 1]$ is a function which returns the probability associated with a transition
- q_0 is the initial state,
- $F : Q \rightarrow [0..1]$ is a function which returns the probability for a state to be final

Furthermore, we only consider PFSA which are structurally deterministic. This constraint comes from the learning algorithm. 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 exits.

In order to define a probability distribution on Σ^* (the set of all words built on Σ), τ and F must satisfy the following consistency constraint :

$$\forall q \in Q \ , \left[\sum_{a \in \Sigma} \tau(q, a)\right] + F(q) = 1$$

A string $a_0 \cdots a_{l-1}$ is generated by an automaton $\mathcal{A}$ iff there exists a sequence of states $e_0 \cdots e_l$ such that

- $e_0 = q_0$
- $\forall i \in [0, l-1], \ \delta(e_i, a_i) = e_{i+1}$
- $F(e_l) \neq 0$.

The automaton assigns to the string the probability

$$P_{PFSA}(a_0 \cdots a_{l-1}) = \left[\prod_{i=0}^{l-1} \tau(e_i, a_i)\right] * F(e_l)$$

Note that PFSA are a particular case of Markov models with discrete emission probabilities on transitions and with final probabilities.

Algorithm 1 Generic PFSA induction algorithm

Input:
I_+, training set (sequences)
α, a precision parameter
Output: a Probabilistic Finite State Automata
A $\leftarrow$ *build_PPTA*(I_+)
while $(q_i, q_j) \leftarrow$ *choose_states*(A) **do**
  if *is_compatible*(q_i, q_j, α) **then**
    merge(A,q_i, q_j)
  end if
end while
return A

2.1 Learning algorithm

Several algorithms have been proposed to infer PFSA from examples [2, 13, 14]. All these algorithms are based on the same scheme, which is presented as algorithm 1.

Given a set of positive examples I_+, the algorithm first builds the probabilistic prefix tree acceptor (PPTA). The PPTA is an automaton accepting all examples of I_+, in which 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 sample set. This count is then used to define the function τ: if $C(q)$ is the number of times the state q is used while parsing I_+, and $C(q, a)$ is the number of times the transition (q, a) is used while parsing I_+, then $\tau(q, a) = \frac{C(q,a)}{C(q)}$. Similarly, if $C_f(q)$ is the number of times q is used as final state in I_+ 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 states are compatible as a function of the precision parameter α (*is_compatible*(q_i, q_j, α)). If the states are compatible, they are merged (function *merge(A,*q_i, q_j*)*). Usually, several consecutive merging operations are done in order to keep the automaton structurally deterministic. The algorithm stops when no more merging is possible. In the case of the Alergia algorithm [2], the compatibility of two states is based on three different tests : the compatibility of their outgoing probabilities on the same letter, the compatibility of their probability to be final and the recursive compatibility of their successors.

More formally, the compatible test is derived from Hoeffding bounds [6]. Two states q_1 and q_2 are compatible iff:

$$\forall a \in \Sigma \qquad \left|\frac{C(q_1, a)}{C(q_1)} - \frac{C(q_2, a)}{C(q_2)}\right| < \sqrt{\frac{1}{2}\ln\frac{2}{\alpha}}\left(\frac{1}{\sqrt{C(q_1)}} + \frac{1}{\sqrt{C(q_2)}}\right) \quad (1)$$

$$\left|\frac{C_f(q_1)}{C(q_1)} - \frac{C_f(q_2)}{C(q_2)}\right| < \sqrt{\frac{1}{2}\ln\frac{2}{\alpha}}\left(\frac{1}{\sqrt{C(q_1)}} + \frac{1}{\sqrt{C(q_2)}}\right) \quad (2)$$

$$\forall a \in \Sigma, \delta_Q(q_1, a) \text{ and } \delta_Q(q_2, a) \text{ are compatible} \quad (3)$$

However, these compatibility tests suffer from several limitations. First the tests 1 and 2 are done independently and no cumulated evidence is used. Second, theses tests derived from Hoeffding bounds which are asymptotic results. Their behavior on finite and limited data is not considered. They are used to prove the identification in the limit of the structure of the target automaton by the algorithm , but might not be adapted for small sample cases.

The search order followed by the Alergia algorithm can also be improved. This algorithm searches for possible merging between states of the PPTA using a hierarchical order. States are ordered according to their depth in the tree and for a given depth according to the symbol labeling their incoming edge, the symbols of the alphabet being ordered according to an arbitrary order. When sufficient data is available, the order in which merging operations are done at a given depth is not critical, since only relevant state pairs are compatible. However, in the case of limited data, it is important to perform first the merging operations supported by the most evidence.

In section 3, we propose a framework for compatibility tests dealing with these limitations.

3 Multinomial state model

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 parameter $\tau = \{\tau_1, \tau_2, \cdots, \tau_K\}$, each τ_i corresponding to the transition probability on the ith letter of the alphabet including a special final state symbol. If a transition on a given letter does not exist from a given state, its probability is set to zero. In the PPTA, each state q is seen as a realization of the multinomial random variable $\tau^q = \{\tau_1^q, \tau_2^q, \cdots, \tau_K^q\}$ of the state in the target automaton it corresponds to. The problem of identifying the target automaton is the same as finding the states in the PPTA which correspond to the same states in the target automaton and thus must be merged. In our framework, states of the PPTA which are assumed to be the realization of the same random variable can be checked for compatibility according to a statistical test.

3.1 Multinomial compatibility test

We consider the H_0 hypothesis that two states q_1 and q_2 of the PPTA must be merged. In this case, they are both a realization of the same multinomial random variable associated with the state of the target automaton they correspond to, $\tau^q = \{\tau_1^q, \tau_2^q, \cdots, \tau_K^q\}$. Using notations of section 2.1, for each state the expected frequency for each transition i is respectively $C(q_1)\tau_i^q$ and $C(q_2)\tau_i^q$. The unknown parameters of the random variables τ^q can be estimated by maximum likelihood:

$$\hat{\tau}_i^q = \frac{C(q_1, i) + C(q_2, i)}{C(q_1) + C(q_2)}$$

	letter a	letter b	total
state q_1	$C(q_1, a)$ $H_{1a} = C(q_1)\hat{\tau}_a^q$	$C(q_1, b)$ $H_{1b} = C(q_1)\hat{\tau}_b^q$	$C(q_1)$
state q_2	$C(q_2, a)$ $H_{2a} = C(q_2)\hat{\tau}_a^q$	$C(q_2, b)$ $H_{2b} = C(q_2)\hat{\tau}_b^q$	$C(q_2)$
total	$C(a) = C(q_1, a) + C(q_2, a)$	$C(b) = C(q_1, b) + C(q_2, b)$	$N = C(q_1) + C(q_2)$

Fig. 1. Observed and expected transition frequencies on a two letters alphabet for two states under H_0 hypothesis

The expected frequencies H_{qi} are then

$$C(q_1)\hat{\tau}_i^q = C(q_1)\frac{C(q_1, i) + C(q_2, i)}{C(q_1) + C(q_2)} \quad \text{and} \quad C(q_2)\hat{\tau}_i^q = C(q_1)\frac{C(q_1, i) + C(q_2, i)}{C(q_1) + C(q_2)}$$

Figure 1 summarizes these results for a two letters alphabet in a contingency table.

The Pearson statistic [11] is one of the most classical statistics to test the H_0 hypothesis:

$$X^2 = \sum_k \left(\frac{C(q_1, k) - H_{1k}}{H_{1k}} + \frac{C(q_2, k) - H_{2k}}{H_{2k}}\right)$$

Several other statistics have been proposed like the log-likelihood ratio statistics [15] or the power-divergence statistics family [4]. All these statistics follow asymptotically a χ^2 distribution with $K - 1$ degrees of freedom. The H_0 hypothesis will be rejected with confidence α if the test statistic X^2 is larger than $\chi^2(K - 1, \alpha)$. This statistic can be used in the following conditions:

- the sample set must be large enough to allow a multinormal approximation of the multinomial distribution. Typically, $C(q)$ must be larger than 20 and $C(q, i)$ must be larger than 5 for all letters.
- the dimension of the multinomial random variables must be constant with respect to the sample size.

In particular, the first condition implies that this statistic can not be used to compare two states as soon as a transition is observed in one of them and not observed in the other. Section 3.2 proposes a solution to this problem, common in real data.

3.2 Small sample case

In the case where χ^2 statistics can not be applied due to too small observed frequencies, the Fisher exact test can be used. Given two states and their transition frequencies summarized in a contingency table, as show on Figure 1, this test consists in computing the probability of all the contingency tables with the same marginal counts as the tested table (same values for $C(q_i), C(q_j), C(a), C(b)$) and at least as unfavorable to H_0. For fixed marginal counts, the probability

Algorithm 2 Recursive contingency table enumeration algorithm

Input:
a $2 \times c$ contingency table: x[2][c]
current position in table : j
line and column table sums : $n_0, n_1, C_0, \cdots, C_r$
partial column sum for current position sc
partial cell sum for cells already set sx
total table sum N
Output: enumerate all tables with same marginal counts
$lj \leftarrow max(sc + n_0 + C_j - sx - N, 0)$ {set minimal value}
$uj \leftarrow min(n_0 - sx, c_j)$ {set maximal value}
for $x(1,j) = lj$ to uj **do**
  $x(2,j) \leftarrow C_j - x(1,j)$ {set line 2 value}
  $sc \leftarrow sc + C_j$ {update sc}
  $sx \leftarrow sx + x(1,j)$ {update sx}
  if $j \neq c - 1$ **then**
    Enumerate_table(x[2][c], $j + 1, n_0, n_1, C_0, \cdots, C_r, sc, sx, N$) {recursive call}
  else
    $x(1,j) \leftarrow n_0 - sx$ {set line 1 value}
    $x(2,j) \leftarrow C_j - x(1,j)$ {set line 2 value}
    output table
  end if
  $sc \leftarrow sc - C_j$ {update sc}
  $sx \leftarrow sx - x(1,j)$ {update sx}
end for

of a contingency table is given by an hypergeometric distribution. To compute the Fisher exact test, we enumerate all the contingency tables with the same marginal counts as the tested table and at least as unfavorable to H_0, we add their probability computed by the hypergeometric distribution and directly compare this sum to the confidence threshold α to accept or reject H_0.

March [8] proposed an iterative algorithm to enumerate all the contingency tables at least as unfavorable to H_0. We propose a recursive version of this algorithm in case of $2 \times k$ tables (see algorithm 2). This algorithm only enumerates the tables with correct marginal counts. It consists in a loop on all cells of the table, in which all possible values are enumerated. Given a cell and a possible value for this cell, the possible values for all other cells are computed with a recursive call to the enumeration function.

3.3 Algorithmic complexity of the test

Using the multinomial compatibility test does not increase the initial complexity of the inference algorithm since χ^2 values can be tabulated. When using the Fisher exact test, the number of contingency tables we need to evaluate in order to compute is exponential in the number of degree of freedom in the table, which is the size of the alphabet. However, several solutions have been proposed

Algorithm 3 Evidence driven state merging

Input: a Probabilistic Prefix Tree Acceptor
Output: a Probabilistic Finite State Automata
  set the initial state to red
  set the direct successors of the initial state to blue
  while there is a blue node **do**
    evaluate all red/blue merging
    if there exists a blue node incompatible with all red node **then**
      promote the shallowest such blue node to red
    else
      perform the highest score red/blue merging
    end if
  end while

to evaluate the Fisher exact test without a complete enumeration of the possible tables using properties of the multinomial distributions [10] or with dynamic programming methods [9]. An hybrid algorithm, using both exact tests and normal approximations has also been proposed [1].

3.4 Evidence driven search strategies

Alternative search strategies have been proposed in the framework of deterministic finite automaton induction from positive and negative examples [5, 7]. Data-dependent strategy [5] is based on the idea that merging operations that are supported by the most evidence must be done first. A variant of this strategy, successfully implemented with an additional merging order constraint by Lang [7], is known as the Blue-Fringe algorithm (see Algorithm 3). This algorithm consists in maintaining a set of states already checked (red states) and a set of states candidate for a possible merging operation (blue states). Blue states are states directly accessible from a red state. All red/blue states pairs are considered. If at least one blue state is incompatible with all the red states, the such blue node with lowest depth is promoted to red. Otherwise, the red/blue merging operation with the highest score is done. Red and blue sets are updated and all red/blue states pairs are considered again. The algorithm stops when there is no more blue state.

We propose here an extension of the Blue-Fringe algorithm to infer probabilistic automata. In the case of a multinomial compatibility test, we need to compare the χ^2 values of all possible merging operations between a red and a blue state. It is not possible to directly compare χ^2 values since they depend both on the number of observations and the size of the contingency table.

To compare the possible merging operations, we propose to use the p-value, the significance level of the test. The p-value is the smallest value of α for which H_0 is rejected. A possible merging operation with a high p-value denotes a strong association between the two states whereas a small p-value denotes a weak association. Other χ^2 results comparison coefficients, like Cramer's V coefficient,

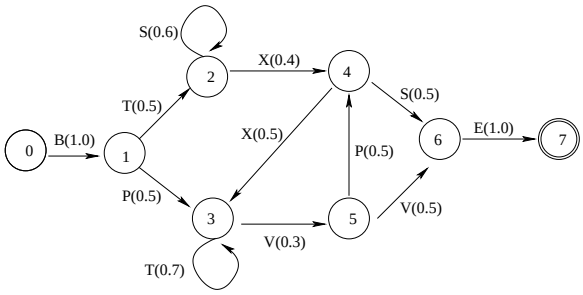

Fig. 2. The SDFA corresponding to the Reber grammar

could be used, but they should be adapted to take into account the case when the Fisher test is used. With a multinomial compatibility test in algorithm 3, the merging operation with the highest score is the one with the highest p-value.

4 Experiments

In order to compare our approach with previously published results [2, 3], we have tested the multinomial based inference on artificial data. We used the Reber grammar [12], presented on Figure 2, as the target automaton. We have inferred automata on randomly generated learning sample sets of size varying from 5 to 100 strings (about 100 to 2000 symbols). All results presented in this section are averaged over 50 different runs of the experiments. The inference algorithms tested were Alergia [2], the proposed inference algorithm based on the multinomial compatibility function, denoted as Malergia, and the evidence-driven variation, denoted as Blue-Malergia.

Since our goal is to improve inference when available data is limited, we do not present inference results in the limit (when the learning set size grows to infinity) but rather on small size learning sets.

All presented inference algorithms depend on a learning parameter (α in algorithm 1). For each algorithm, this parameter has been tuned to the value leading to the fastest convergence to an automaton with the same number of states as the target automaton ($\alpha = 0.1$ for Alergia and $\alpha = 0.005$ for Malergia and $\alpha = 0.001$ for Blue-Malergia). Note that , if we use standard thresholds on frequencies to decide when to apply exact tests, the number of parameters needed when using multinomial tests is the same as Alergia.

Figure 3 shows the number of states of the inferred automaton as a function of the size of the training set for the three algorithms. The two versions of Malergia show a faster convergence to the correct number of states.

We have evaluated the statistical distance between the target language and the inferred language. On a large sample S of the target language (10 000 words, 96232 symbols), we have computed the average difference between the probability P_t assigned by the target automaton and the probability P_i assigned by the

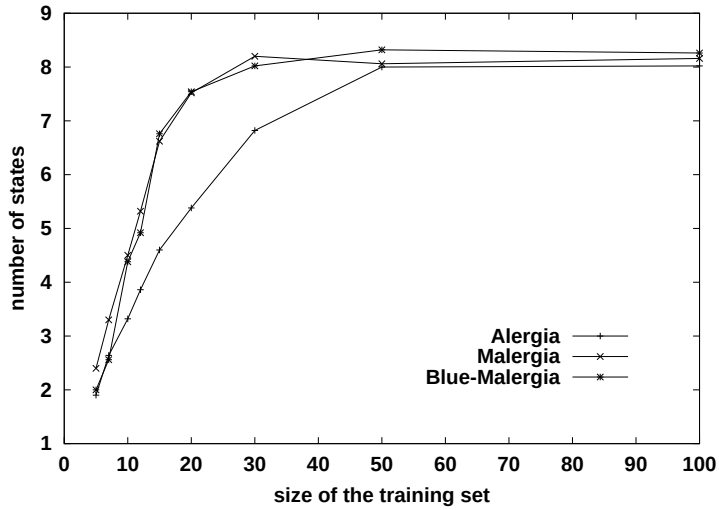

Fig. 3. Number of states of the inferred automaton for increasing learning set size for Alergia, Malergia and Blue-Malergia.

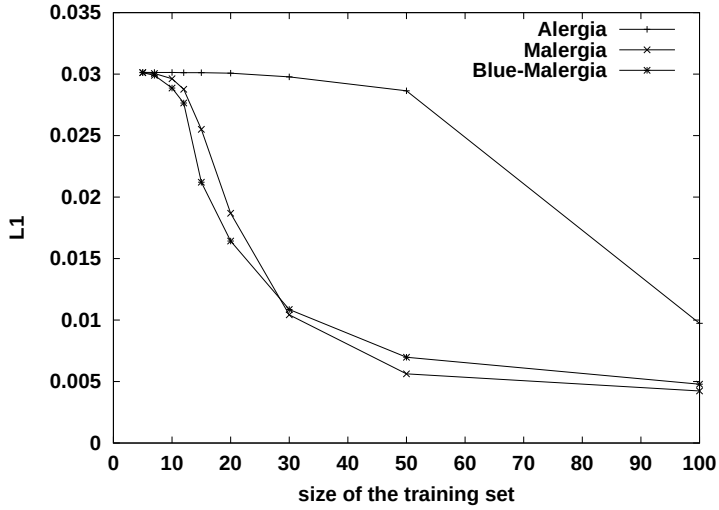

Fig. 4. Value of $\bar{L}1$ for increasing learning set size for automata inferred with Alergia, Malergia and Blue-Malergia

inferred automaton:

$$\bar{L1} = \frac{1}{|S|} \sum_{w \in S} |P_t(w) - P_i(w)|$$

Figure 4 shows the value of $\bar{L1}$ when the size of the learning sample is increasing. The use of multinomial tests significantly reduce the average error made when assigning a probability to a word in the target language with the inferred automaton. However , on this task, the advantage of using evidence driven strategy is not shown. This point should be further explored.

5 Conclusion

We have proposed a new statistical framework for grammatical inference algorithms based on a state merging strategy. Each state is considered as a realization of a multinomial distribution and a merging operation of two states is evaluated with a χ^2 test. In this framework, small sample case can be particularly dealt with, all frequencies concerning two states to be merged are used in a single test so that statistical evidence is cumulated and possible merging operations can be compared. Further evaluations of this framework should be conducted, in particular on real data.

References

1. J. Baglivo, D. Olivier, and M. Pagano. Methods for analysis of contingency tables with large an small cell counts. *Journal of the American Statistical Association*, 83(404):1006–1013, 1988.
2. R. C. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *Proc. Int. Coll. on Grammatical Inference*, volume 862 of *Lecture Notes in Artificial Intelligence*, pages 139–152. Springer Verlag, 1994.
3. R. C. Carrasco and J. Oncina. Learning deterministic regular grammars from stochastic samples in polynomial time. *RAIRO (Theoretical Informatics and Applications)*, 33(1):1–20, 1999.
4. N. Cressie and T.R.C. Read. Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society Series B*, 46:440–464, 1984.
5. C. de la Higuera, J. Oncina, and E. Vidal. Identification of DFA: data-dependent vs data-independent algorithms. In Laurent Miclet and Colin de la Higuera, editors, *Proceedings of the Third International Colloquium on Grammatical Inference (ICGI-96): Learning Syntax from Sentences*, volume 1147 of *LNAI*, pages 313–325, Berlin, September 25–27 1996. Springer.
6. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
7. Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the Abbadingo One DFA learning competition and a new evidence-driven state merging algorithm. In Springer-Verlag, editor, *Proc. Int. Coll. on Grammatical Inference*, volume 1433 of *LNAI*, pages 1–12, 1998.
8. D.L. March. Exact probability for r × c contingency tables. *Communications of the ACM*, 15(11):991–992, November 1972.

9. C. Mehta and N.R. Patel. A network algorithm for performing fisher's exact test in r×c contingency tables. *Journal of the American Statistical Association*, 78(382):427–434, 1983.
10. M. Pagano and K. Taylor Halvorsen. An algorithm for finding the exact significance levels of r ×c contingency tables. *Journal of the American Statistical Association*, 76(376):931–934, 1981.
11. K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophy Magazine*, 50:157–172, 1900.
12. A.S. Reber. Implicit learning of artificial grammars. *Journal of verbal learning and verbal behaviour*, 6:855–863, 1967.
13. D. Ron, Y. Singer, and N. Tishby. On the learnability and usage of acyclic probabilistic automata. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 31–40, Santa Cruz, CA, 1995. ACM Press.
14. F. Thollard and P. Dupont. Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In *Proc. Int. Conf. on Machine Learning*, pages 975–982. Morgan Kaufmann, San Francisco, CA, 2000.
15. S.S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics*, 9:60–62, 1938.
16. M. Young-Lai and F. WM. Tompa. Stochastic grammatical inference of text database structure. *Machine Learning*, 40:111–137, 2000.