BLUE*: a Blue-Fringe Procedure for Learning DFA with Noisy Data

Marc Sebban MARC.SEBBAN@UNIV-ST-ETIENNE.FR
Jean-Christophe Janodet JANODET@UNIV-ST-ETIENNE.FR
Frédéric Tantini FREDERIC.TANTINI@C2M.UNIV-ST-ETIENNE.FR
EURISE, Faculty of Sciences, 23 rue Paul Michelon, University of Jean Monnet, 42023 Saint-Etienne, FRANCE

Grammatical Inference is a subtopic of machine learning whose aim consists in learning models of languages such as grammars, deterministic finite automata (DFA) or stochastic automata. Among the algorithms aiming at learning DFA, those based on state merging are widely studied, and particularly RPNI (Oncina & García, 1992) and EDSM (Lang et al., 1998). Both of them learn from a sample $E = E_+ \cup E_-$, and try to infer, by state merging, a small DFA that accepts all the strings of E_+ (called the positive examples), and rejects all those of E_- (called the negative ones). RPNI and EDSM are *exact* learning algorithms because they fit the data: it is proven that if E contains some special (*characteristic*) strings, then these algorithms are able to infer, in polynomial time, the DFA that produced the data (Oncina & García, 1992), called the *target* DFA. However, the presence of noisy data challenges these theoretical properties. Since they are not immune to overfitting, noisy data penalize the DFA they produce in terms of number of states and error rate.

In (Sebban & Janodet, 2003), we described a first approach that aimed at limiting the risk of overfitting. We relaxed the merging rule of RPNI and introduced a new algorithm, called RPNI*. The first task RPNI* achieves is the construction of the PTA (*prefix tree acceptor*) of the strings of E_+, that is the greatest trimmed DFA accepting only the strings of E_+. Its states are numbered following the hierarchical order over the prefixes of E_+ (Oncina & García, 1992) (see the upper DFA in Fig.1). A state is *positive* (or final) if it contains strictly more positive strings (of E_+) than negative ones (of E_-), and *negative* otherwise. Then RPNI* runs along these states following the ordering. When state i is considered, RPNI* tries to merge it with states $0, \dots, i-1$, in order. Merging two states means to collapse them into one new state, whose number is the smallest of the two merged ones. As for the outgoing transitions, they are themselves merged together if they are labeled with the same letter, and in such a case, the two pointed states are recursively merged.

```
Data   : a sample E+, E-
Result : a DFA A
A ⟵ PTA(E+)
for i = 1 to N − 1 do
    j ⟵ 0; continue ⟵ true
    while (j < i) and continue do
        B ⟵ compute_merging(A, i, j)
        C ⟵ compute_final_states(B, E+, E-)
        if statistically_acceptable(C, E+, E-) then
            A ⟵ C; continue ⟵ false
        j ⟵ j + 1
return(A)
```

Algorithm 1: Pseudo-code of RPNI*.

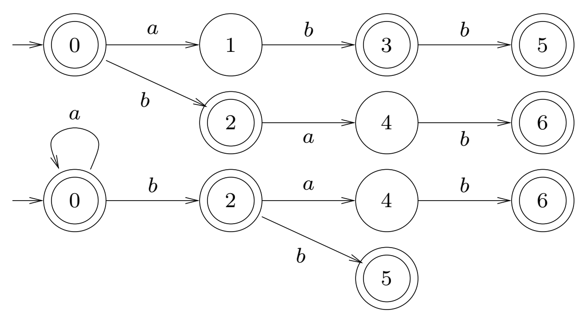

Figure 1. We assume here that $E_+ = \{\lambda, b, ab, abb, bab\}$ and $E_- = \{aa, ba\}$ (λ denotes the string with no letter). The upper DFA is the PTA of E_+. The lower DFA results of the merging of states 1 and 0.

A merging can be acceptable or not. More precisely, we say that a negative (resp. positive) string is *misclassified* if it is contained by a positive (resp. negative) state. A merging is *statistically acceptable* if the proportion p_2 of misclassified strings in the whole DFA after the merging is not significantly higher than the proportion p_1 of misclassified strings computed before the merging. Statistically speaking, we test the null hypothesis $\mathcal{H}_0 : p_1 = p_2$, *vs* the alternative one $\mathcal{H}_a : p_2 > p_1$. This test is one-tailed since only a sufficiently large value of the statistic $p_2 - p_1$ must lead to rejection of the tested hypothesis. Actually, a small value of the statistic (and of course a negative one) does not challenge the quality of the merging. The quantities p_1 and p_2 are unknown, because they correspond to the theoretical errors of the current DFA respectively before and after the merging. They can only be assessed by the empirical errors $\hat{p}_1 = N_1/N$ and $\hat{p}_2 = N_2/N$ computed from the learning set, where N_1 (resp. N_2) is the total number of misclassified learning examples before (resp. after) the merging, and N is the learning set size (*i.e.* the number of words in $E_+ \cup E_-$). $\hat{p}_1$ and $\hat{p}_2$ are independent random variables and are unbiased estimators of p_1 and p_2.

In our test, we are interested in the difference $\hat{p}_2 - \hat{p}_1$ whose mean and variance are $E(\hat{p}_2 - \hat{p}_1) = p_2 - p_1 = 0$ and $Var(\hat{p}_2 - \hat{p}_1) = (p_2(1-p_2) + p_1(1-p_1))/N = 2pq/N$, with $p = p_1 = p_2$ and $q = 1-p$ under the null hypothesis $\mathcal{H}_0$. p is estimated by the mean of the two proportions of misclassified examples before and after the merging: $\hat{p} = (\hat{p}_1 + \hat{p}_2)/2$. If $Np > 5$ and $Nq > 5$, the approximation conditions to the normal distribution are satisfied, so the variable $T = \hat{p}_2 - \hat{p}_1$ follows the normal law $\mathcal{N}(p_2 - p_1, \sqrt{2\hat{p}\hat{q}/N})$. We need to determine the threshold Z_α, called *critical value at the risk* α, which defines the bound of the rejection of $\mathcal{H}_0$ and corresponds to the $(1-\alpha)$-percentile of the distribution $\mathcal{N}(p_2 - p_1, \sqrt{2\hat{p}\hat{q}/N})$:

$$P(T > Z_\alpha) = P(T^{cr} > Z_\alpha / \sqrt{2\hat{p}\hat{q}/N}),$$

where T^{cr} is the centered and reduced variable. So

$$P(T > Z_\alpha) = \alpha \text{ iff } Z_\alpha = U_\alpha \sqrt{2\hat{p}\hat{q}/N},$$

where U_α is the $(1-\alpha)$-percentile of the normal law $\mathcal{N}(0,1)$. If $T > Z_\alpha$, we reject the hypothesis $\mathcal{H}_0$, thus the merging, with a risk of $\alpha\%$. On the contrary, if $T \leq Z_\alpha$, then the merging is statistically validated, thus accepted.

From an experimental standpoint, we showed in (Sebban & Janodet, 2003) that RPNI* could yield a significant improvement over RPNI's performances. However, these performances are challenged by the competition "Learning DFA with Noisy Data". Indeed, RPNI* is able to learn small DFA from dense sample but does not behave so well on large DFA that must be learnt from sparse samples. It was known since the competition "Abbadingo One" that RPNI had the same problem. The winner of this last competition had the idea of improving RPNI by 1) delaying a merging as much as possible in order to have several choices of possible mergings and 2) performing the "best" merging between them (see (Lang et al., 1998) for details). We have decided to follow the same line here, by adapting a *blue-fringe*-like procedure to the presence of noisy data. This approach is enforced by the fact that we use statistical tests to accept or reject a merging: Such tests provide rigourous indicators of the quality of an acceptance or a rejection of a merging throughout the risks of first and second order. We take advantage of these risks to select the best mergings.

```
Data   : a sample E+, E−
Result : a red DFA A
A ⟵ colored_PTA(E+)
while there exists a blue state do
    P ⟵ ∅; M ⟵ ∅
    foreach blue state b do
        no_merge_found ⟵ true
        foreach red state r do
            if are_mergeable(A,b,r) then
                no_merge_found ⟵ false
                M ⟵ M ∪ {(b,r)}
        if no_merge_found then  P ⟵ P ∪ {b}
    if P ≠ ∅ then
        A ⟵ do_best_promotion(A,P)
    else A ⟵ do_best_merging(A,M)
return(A)
```

Algorithm 2: Pseudo-code of BLUE*

BLUE* works with red-blue-white DFA. As RPNI*, the first task BLUE* achieves is the construction of the PTA of E_+. However, every state of this PTA will have a "colored" life: The initial state of the PTA is red, its immediate successors by transitions are blue and all the other states are white. During the execution of the algorithm, the red states form the stable part of the current DFA *w.r.t.* the mergings: if state i is red, then its mergings with states $0, 1, \ldots, i-1$ were tested and rejected, so state i will necessarily be a state of the final DFA. As for the non-red states, they may be either blue or white.

More precisely, a non-red state will become blue iff it is the successor of a red state by a transition, and white otherwise. At each round of the main loop, BLUE* focuses on these blue states and tries to merge them with all the red states. As we wrote it before, the general strategy of BLUE* is to delay the acceptable mergings as much as possible in order to maximize their number and to perform the best one among them. Therefore, at the end of the second loop of BLUE*, a blue state b has two possible status: either b is *mergeable* with at least one red state, and then this merging may be chosen to perform the best merging; this is the reason why BLUE* maintains a list M of all possible mergings. Or b is not *mergeable* with any red state, and then we say that it is *promotable* and keep it in list P.

Promoting a blue state means to recolor it in red and to recolor its successors by a transition in blue. If there exist promotable blue states at the end of the second loop (in list P), then BLUE* chooses one of them and promotes it. As several blue states may be promotable, it seams reasonnable to promote the *best* one, that is to say, the one which has the greatest chance to be really red in the target DFA[1]. As we mentioned it above, a blue state is promotable into a red state iff all its mergings with the red states failed. However, this rejection may be due to the presence of noisy data, *i.e.*, we would have accepted this merging in the absence of noise. We can easily measure the risk $\alpha_{b,r}$ of having wrongly rejected the merging of blue state b and red state r, since it is the risk of first order of our proportion comparison test:

$$\alpha_{b,r} = P(\mathcal{H}_0 \text{ rejected} \mid p_2 - p_1 = 0)$$

$$\Leftrightarrow Z_{\alpha_{b,r}} = (\hat{p_2} - \hat{p_1})/\sqrt{2\hat{p}\hat{q}/N}$$

Let us define $\alpha_b = \max_r \alpha_{b,r}$. α_b is the risk of having rejected the merging of $b \in P$ with *every* red state r. So if we choose to promote the blue state b which *minimizes* α_b, we minimize the risk of promoting a blue state that should be merged in the target DFA. So the promoted state must be: $b^* = \text{argmin}_{(b \in P)} \alpha_b$.

When all blue states are mergeable, our aim is to realize the best merging. A first idea is to choose a pair $(b, r) \in M$ that minimizes the risk of having wrongly accepted their merging, which corresponds to the risk of second order of our merging test. More precisely, given a pair $(b, r) \in M$, let $\beta_{b,r} = P(\mathcal{H}_0 \text{ accepted} \mid \mathcal{H}_a \text{ true})$. This definition must be improved, since we do not know the statistical law followed by hypothesis $\mathcal{H}_a : p_2 - p_1 > 0$. So we fix a parameter $\delta > 0$ and we overcome the difficulty by evaluating:

$$\begin{aligned}
\beta_{b,r}(\delta) &= P(\mathcal{H}_0 \text{ accepted} \mid p_2 - p_1 = \delta) \\
&= P(Z \leq Z_\alpha \mid p_2 - p_1 = \delta) \\
&= P((\hat{p_2} - \hat{p_1})/\sqrt{2\hat{p}\hat{q}/N} \leq Z_\alpha \mid p_2 - p_1 = \delta) \\
&= P(\tfrac{(\hat{p_2} - \hat{p_1}) - \delta}{\sqrt{2\hat{p}\hat{q}/N}} \leq Z_\alpha - \tfrac{\delta}{\sqrt{2\hat{p}\hat{q}/N}}) \\
&= P(\mathcal{N}(0,1) \leq Z_\alpha - \delta/\sqrt{2\hat{p}\hat{q}/N})
\end{aligned}$$

As the minimum value of the set $\{\beta_{b,r}(\delta) : (b, r) \in M\}$ is independent of δ, we may choose the pair (b, r) that minimizes the above quantity: $(b^*, r^*) = \text{argmin}_{(b,r) \in M} \beta_{b,r}(\delta)$. However, this criterion favours uninteresting mergings, *i.e.*, mergings that do not allow to earn states. In other words, such mergings are safe but unuseful. So we decide to define $n_{b,r}$ as the number of the states that are earned after the merging of b and r, and we select the pair that minimizes $\beta_{b,r}(\delta)$ while maximizing $e_{b,r}$:

$$(b^*, r^*) = \text{argmin}_{(b,r) \in M} \frac{\beta_{b,r}(\delta)}{e_{b,r}}$$

[1]We could also to promote all of them, but the creation of a new red state may transform another promotable blue state into a mergeable one ...

References

Lang, K., Pearlmutter, B., & Price, R. (1998). Results of the abbadingo one DFA learning competition. *Fourth International Colloquium on Grammatical Inference* (pp. 1–12).

Oncina, J., & García, P. (1992). *Inferring regular languages in polynomial update time*, vol. 1 of *Machine Perception and Artificial Intelligence*, 49–61. World Scientific.

Sebban, M., & Janodet, J. (2003). On state merging in grammatical: a statistical approach for dealing with noisy data. *Twentieth International Conference on Machine Learning* (pp. 688–695).