Learning Shifted Automatons

Andrew Hogue[*]

November 4, 2003

We look at forming finite automata to match examples where repeating patterns are expected.

For example, take the set of observations $O = \{o_1, ..., o_m\} = \{$L,A,B,C,A,B,C,A,B,C,A,B,C,R$\}$. We notice that the observations contain a repeating pattern, $\{$A,B,C$\}$, of which we would like advantage. We define the notion of the *edit distance*, γ, between two observations to be[1]:

$$\gamma(x,y) = \begin{cases} 0 & \text{if } x = y; \\ 1 & \text{if } x = null \text{ or } y = null \\ 1 & \text{if } x \neq y. \end{cases}$$

By "shifting" the observations by some number of steps n, we can attempt to find a "period" for the repetitions in the observations:

n																	
0	L	A	B	C	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	R			
1		L	A	B	C	A	B	C	A	B	C	A	B	C	R		
2			L	A	B	C	A	B	C	A	B	C	A	B	C	R	
3				L	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**	A	B	C	R
...																	

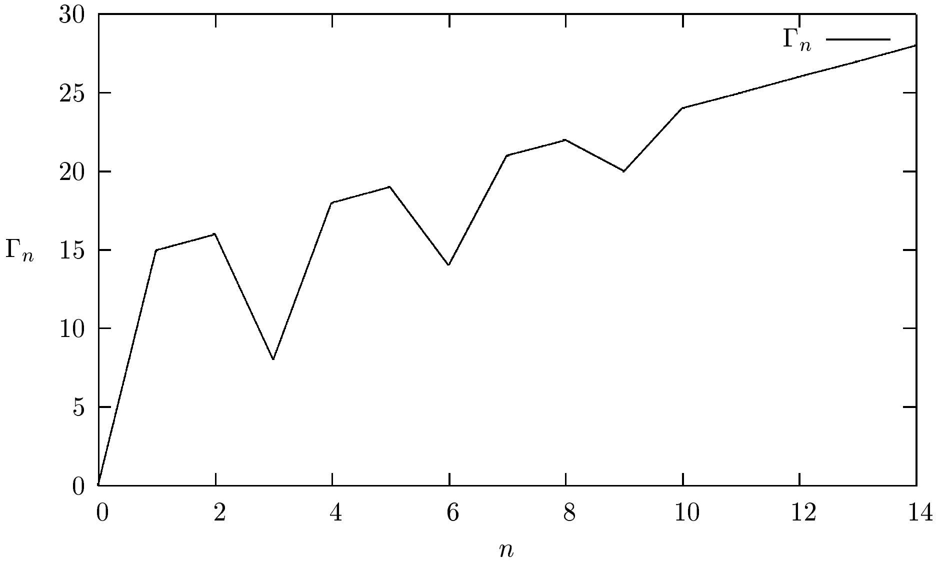
We note that for $n = 3$, several of the observations "line up". If we look at the sum of the edit distances

$$\Gamma_n = \sum_{i=1}^{m} \gamma(o_i, o_{i+n}),$$

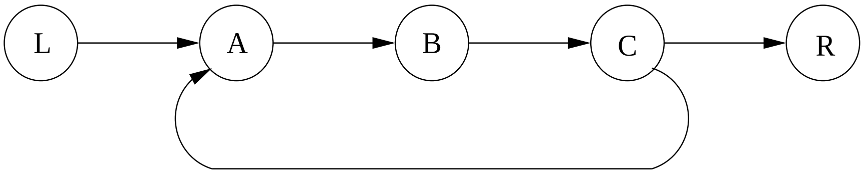
we get a graph like:

[*] ahogue@lcs.mit.edu

[1] In practice, edit distance is calculated by directly comparing the two nodes under consideration, as well as their children if they are tree structures.



This plot gives a clear indication of a repetative pattern in the observations. Using the minima at $n = \{3, 6, 9\}$, we can pull out the sequence $\{A, B, C\}$ and merge it into a loop in the automaton. This allows us to end up with the following automaton:



Another way to look at these repetitive patterns is by comparing each observation with every other observation. We get a matrix like:

	L	A	B	C	A	B	C	A	B	C	A	B	C	R
L	0	1	1	1	1	1	1	1	1	1	1	1	1	1
A		0	1	1	**0**	1	1	**0**	1	1	**0**	1	1	1
B			0	1	1	**0**	1	1	**0**	1	1	**0**	1	1
C				0	1	1	**0**	1	1	**0**	1	1	**0**	1
A					0	1	1	**0**	1	1	**0**	1	1	1
B						0	1	1	**0**	1	1	**0**	1	1
C							0	1	1	**0**	1	1	**0**	1
A								0	1	1	**0**	1	1	1
B									0	1	1	**0**	1	1
C										0	1	1	**0**	1
A											0	1	1	1
B												0	1	1
C													0	1
R														0

Once again, we notice a clear diagonal pattern. By shifting nodes by $n = 3$, we get a much higher rate of matching (that is, smaller edit distances along the diagonal).