Behavior Classification for Turing Machines

Nestor Diaz
Faculty of Electronics and Telecommunications Engineering
Universidad del Cauca, Popayán, Colombia
and
Faculty of Engineering
Systems Engineering and Computation School
Universidad del Valle, Cali, Colombia

A classification for Turing machines is described. Quantitative descriptors for Turing machine behavior are used for measuring repetitiveness, periodicity, complexity and entropy. These descriptors allowed identifying several kinds of behavior for Turing machines, using an approach based on machine learning. The classification was tested and generality was probed over different configurations of Turing machines.

1. Introduction

Some decades ago, Stephen Wolfram developed a classification for cellular automata where a particular cellular automaton is classified into one of four classes according to some characteristics of its evolution [1]. We achieved a classification for Turing machines (TMs) in several classes according to the complexity of their evolution.

Alan Turing proposed TMs in the first half of the twentieth century [2]. Turing machines are an essential element in the development of computer science because they are simpler theoretical computers. Turing machines consist of a set of states for a mobile head, a set of colors that can be written by the head, and a tape where the head can write a color from the set of colors. In this paper, we use Wolfram Mathematica's implementation of TMs. From a starting initial configuration, the head follows a set of rules that decides changes to be made on the tape. An evolution of a TM is performed for several steps, and on the record of this evolution, we can analyze patterns and complexity of behavior.

Some previous work has analyzed behavior, not specifically in TMs, but in cellular automata. Cellular automata share similarities with TMs, because they follow rules that evolve in time in a similar way. In [3], Zenil reviewed and proposed some mechanisms to compress the behavior of cellular automata and Turing machines, and discussed the inherent capability of measuring the complexity of their

https://doi.org/10.25088/ComplexSystems.26.3.283

behavior, also establishing a link to the notion of computational universality extrapolated from the analysis. In the same way, that approach was elaborated to a point where it was related to Wolfram's classification of cellular automata [4]. We can find a complementary approach in [5], which was focused on some concepts drawn from information theory and the specific application of Shannon's entropy and Kolmogorov's complexity to the analysis of behavior for cellular automata and their capabilities to characterize complexity of behavior. In the same way as in previous work, we were able to establish the relationship between their analysis and Wolfram's classification for cellular behavior.

The motivation behind this paper is to offer a way to analyze TM behavior through a machine learning approach.

2. Materials and Methods

In this section, we describe each step of our methodology and assess classification results.

2.1 Features Description

Our set of features was defined to measure variability and complexity in TM evolution.

> *First moment*. Quantification of state variability between each pair of consecutive states.
>
> *Second moment*. Quantification of variability for sequences of states.
>
> *Compression ratio*. Ratio for the difference of a compressed representation for a TM evolution and a compressed representation for nonzero sequences in TM evolution.
>
> *Entropy*. Measures entropy for blocks in the TM evolution.
>
> *Nonperiodicity*. Measures periodicity, low values for highly repetitive sequences.

The described set of features is adapted from [6], where an attempt was made to identify 4-state, 4-color TMs with interesting behavior.

2.2 Clustering Evaluation

To assess clustering quality, we used Silhouettes [7]. Silhouettes is a powerful tool that allowed us to envisage graphically and quantitatively how well classified the TMs are for a dataset. For each TM in a cluster, a value in the range $[-1, 1]$ is computed. A positive Silhouette value means that TMs in that cluster are consistently grouped with regard to the whole dataset. The greater the Silhouette, the higher the quality of the clustering.

2.3 Dataset Description

All data generated in this section was computed in Mathematica, taking advantage of its built-in support for TM simulation.

2.3.1 Data Building

We focused our data generation on TMs with three colors and three states, which gave us around two billion TMs. Within this space, we had the option to sample randomly several datasets, which enabled us to proceed with an iterative process toward a classification of TM behavior with high quality. Each dataset was built in the same conditions: features were computed over 100 evolution steps, starting at the same initial condition (i.e., head at state 1 on a tape filled with zeros).

2.3.2 Data Exploration

At first, we did not know anything about distribution or characteristics of the data. With the aim of exploring data distribution, we generated a random set of 4200 TMs and applied k-means clustering with four clusters. Cluster distribution was biased to a big cluster with 98% of the TMs in it and the rest in the other three clusters. A biased dataset makes it hard to train any machine learning algorithm. To overcome this issue, we used the initial clustering to classify a new dataset of 100 000 TMs and from it took all TMs in the small clusters (3240 TMs), and a subsample of 2760 TMs of the biggest cluster. This second dataset was less biased and allowed us to train classification algorithms with more confidence. The impact of imbalanced data in clustering is an issue that has been previously described in [8].

With dataset number two, we trained a new classifier (using Mathematica's `ClusterClassify`). Cluster distribution was improved, resulting in three big clusters with densities of 65%, 23% and 9%, and a small cluster with a density of 3%. This classification was worth analyzing.

In Figure 1, we can see clusters with high quality (Silhouette scores above 0.85). The Silhouettes plot in Figure 1 shows that most TMs in all clusters have a high degree of membership, especially for clusters 1 and 4. The weighted Silhouette for this clustering is 0.927.

With the steps as described, we learned that TM sampling has to be made in a stratified way to avoid bias for oversampling of the biggest class.

2.4 Classification Building

Knowing some characteristics of our data, we proceeded to build our classification for TMs. At this point, we knew that four classes is not the best partitioning. Therefore, we had to find an optimum for the number of clusters and associate to it the optimal classification. Our case is a typical situation for unsupervised learning, because we were trying to learn the classes that are implicitly represented by our features in the TM set.

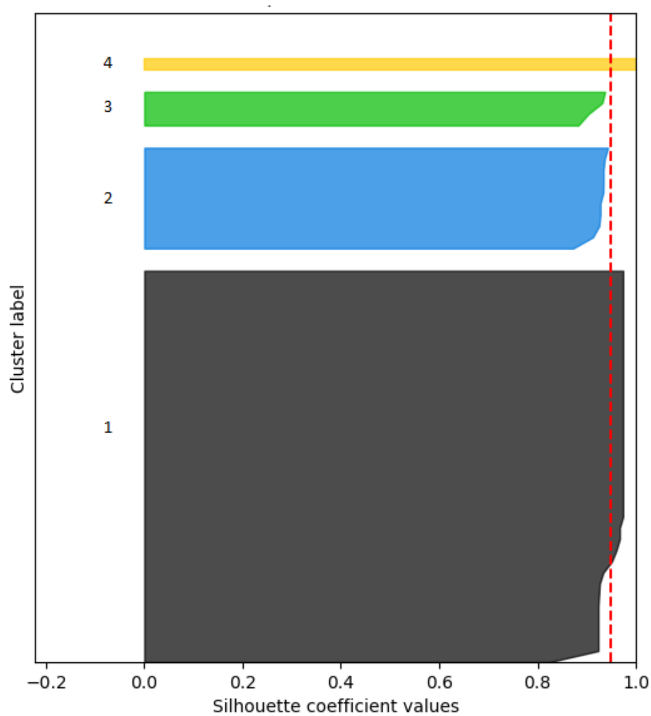

Figure 1. Clusters distribution and quality in data exploration.

Our third dataset was larger than the previous ones; we were sampling one million TMs. This dataset was used for training and validation. We used subsamples of 100 000 TMs for training and validation, with cluster sizes ranging from 4 to 10 clusters. For each clustering, we followed the same procedure: (1) sample 100 000 TMs for training; (2) sample 100 000 TMs for validation; (3) build classification with Mathematica's ClusterClassify; and (4) evaluate clustering quality by means of Silhouettes analysis.

2.4.1 Optimal Clustering

The best TM classification was achieved with eight, nine and 10 clusters. Figure 2 shows the quality for classification with the number of clusters in the range [3, 12]. Classification quality improves as the number of clusters increases, with a peak at nine clusters. In Figure 3, we show the Silhouette plot for an optimal clustering with nine clusters. For every cluster, TMs show higher membership with positive Silhouette coefficient values and just a few negative values.

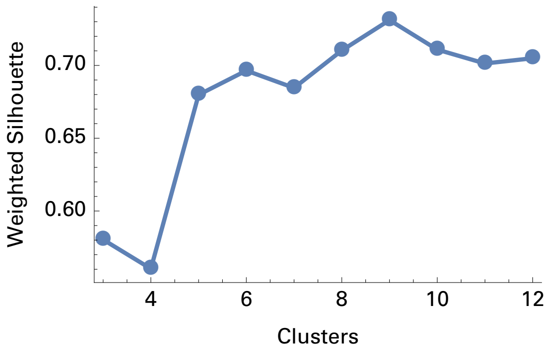

Figure 2. Number of clusters versus weighted Silhouette.

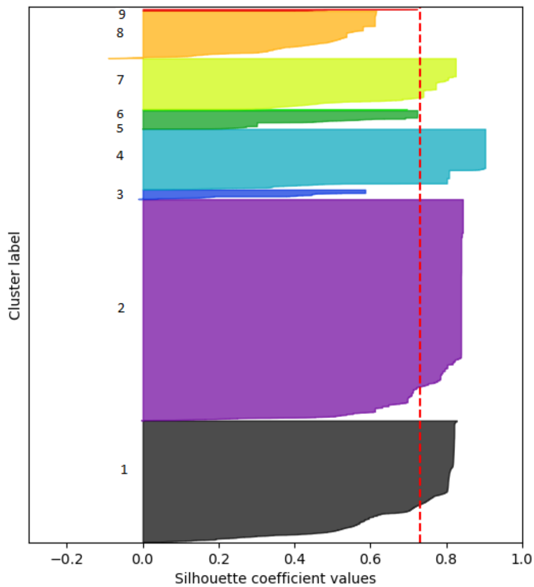

Figure 3. Silhouettes classification with nine clusters on an unbiased dataset.

At the conclusion of this analysis, we selected nine clusters as the optimal number of classes, because the Silhouettes analysis gives better performance, both numerically and graphically.

2.4.2 Improved Classification

We built a fourth dataset, aiming to improve the classification with nine clusters by using an unbiased dataset. This dataset has 40% of the TMs for cluster 1, 25% of the TMs for cluster 2 and 35% of the TMs for other clusters. We trained a new classifier whose quality analysis is depicted in Figure 3. This new classifier had better performance in each cluster; in clusters 1 and 2, TM Silhouettes are improved and the proportion of misclassified TMs is reduced. The weighted Silhouette for this classification is 0.7312; for the previous one it was 0.6314.

As a result of the methods described in this section, we achieved a classification of TMs based on behavior. Quality assessment in Figure 3 allows us to consider this classifier as one that is able to differentiate with confidence between classes of behavior in the evolution of TMs.

3. Results and Discussion

3.1 Analysis of Classes

In Section 2, we described the method that allowed us to obtain a classification for TM behavior. In this section, we analyze the nine classes and the behavior associated to them.

3.1.1 Cluster Medoids

Our set of cluster medoids is comprised of TMs that are representative for each group. We compared these TMs in a graphical way using a dendrogram. The dendrogram depicted in Figure 4 was built from plots for the nine medoids of the clustering described in Section 2.4.2. In Figure 4, we can distinguish three major classes: class I, simpler behavior, low entropy, clusters 1, 2, 4, 8, 7 and 6; class II, complex behavior, high entropy, clusters 7, 3 and 5; class III, repetitive but not simple behavior, medium entropy; class IV, complex behavior, medium to high entropy, medium periodicity. Class I contains 96% of 3-state, 3-color TMs. This classification is not an arbitrary one, but a result extracted from the dendrogram in Figure 4. We decided to cut the dendrogram at level three, because this way the very differentiated clusters 3, 6 and 9 form independent groups and the more related set of clusters is left in one big class. This classification has similarities with the known Wolfram's classification for cellular automata behavior described in [4].

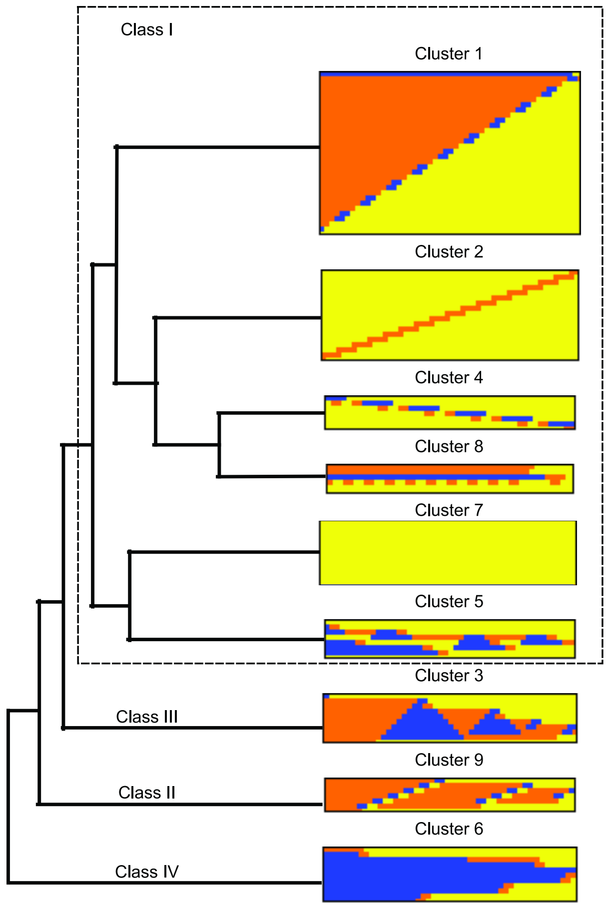

Figure 4. Dendrogram for medoids of the optimal clustering with nine clusters.

3.1.2 Class I

Turing machines in class I show a highly homogeneous behavior characterized by medoids with the lowest values for entropy, high compressibility, and low variability to the left or the right of the tape. However, every cluster in class I has some particularities. For example, clusters 1 and 8 have the biggest entropies, which indicates that there is some complexity in their behavior, even when in the end they become uniform. Turing machines in cluster 1 have the lowest compression ratio; even so, it is highly periodic. Turing machines in cluster 8 are less periodic, but the compression ratio is larger than that of the TMs in cluster 1. In Figure 5(a), we show a compressed version of a TM in class I. The compressed behavior keeps changes of state to the left or the right in the tape.

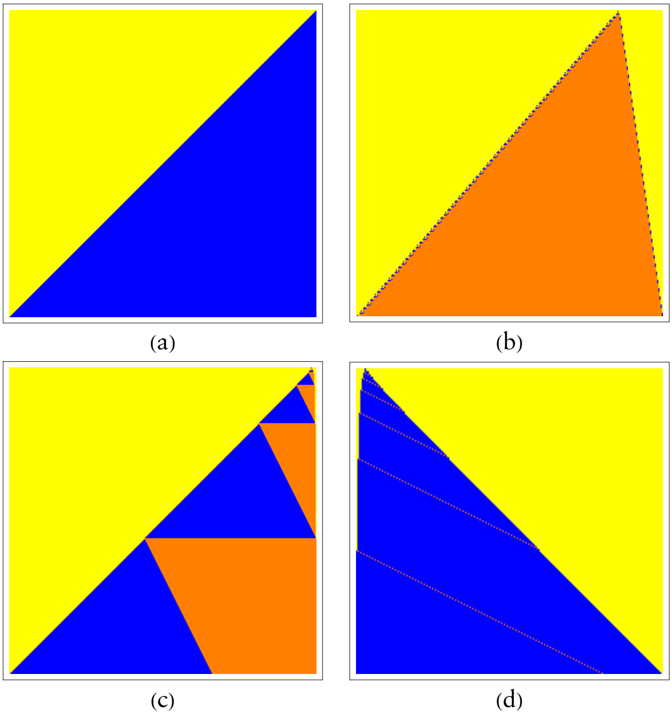

Figure 5. Examples of behavior in a compressed representation of the evolution for TMs in each of the proposed classes. (a) Class I: 3-state, 3-color TM number 177 514 146 690. (b) Class II: 3-state, 3-color TM number 160 668 677 918. (c) Class III: 3-state, 3-color TM number 51 748 890 591. (d) Class IV: 3-state, 3-color TM number 100 715 835 730.

3.1.3 Class II: Periodicity and Stability

In class II, we group TMs that show a behavior slightly more complex than in those in class I. The behavior of the compressed evolution shows some repetitive patterns that remain over the evolution of the TMs (Figure 5(b)). The TM evolution has some variability toward the right or left side in a less monotonous way than for TMs in class I.

3.1.4 Class III: Nonperiodic

Turing machines in class III show more variability in evolution, alternating patterns between states and growing in size as evolution progresses. Figure 5(c) shows a TM that displays this kind of behavior.

3.1.5 Class IV: Complex Patterns

This class represents TMs that have both complex and periodic behavior. Entropy in class IV is just above the entropy for TMs in class III, but the patterns exhibit interesting complexity. Compressed TM behavior presented in Figure 5(d) is very interesting because state 2 (orange color) appears in successions that resemble Mersenne numbers [9]. In Figure 5(d), with 50 000 simulation steps we obtain the sequence 3, 7, 15, 31, 63 and 127 (if you count orange states in each oblique line). This behavior resembles some kind of computation done by this TM; it will be interesting to focus on this TM in future work.

3.2 Classification Generality

We assessed generality for our proposed classification, using 12 datasets drawn from a set of TM configurations. We used configurations in the range [2, 6] for both states and colors. Figure 6 shows the weighted Silhouette for each configuration. The lower Silhouette in Figure 6 is for 6-state, 6-color TMs, though a Silhouette greater than 0.55 is good. For each dataset in this test, we obtained a high Silhouette value, which shows that our classification is extensible to TMs in configurations other than three states and three colors. From Figure 6, we can note a downward trend in the Silhouette value as we increase the number of states.

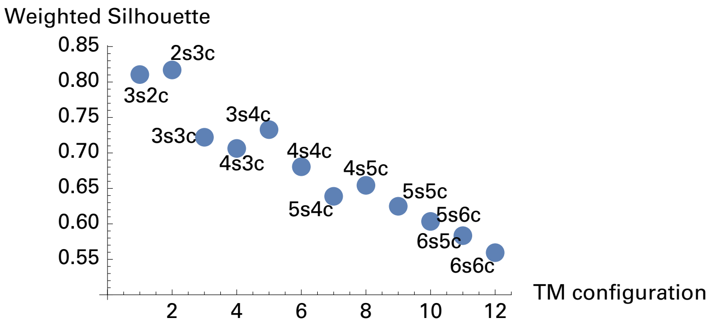

Figure 6. Weighted Silhouette for TMs on a small set of configurations. Labels identify *n* states with *m* colors (e.g., three states with two colors: 3s2c).

3.2.1 Analysis of 2-State, 3-Color Turing Machines

Turing machines with two states and three colors were analyzed in Chapter 11 of [4]. Wolfram concluded that there are just a few TMs with complex behavior. We classified the whole TM set (about three million) and found out that most TMs have simpler behavior (class I).

In order to get a better understanding of our results in the light of Wolfram's claims, we followed an approach based on TM 596 440, which has been proved as universal [10]. We selected TMs in cluster 6 (class IV), then we performed a subclustering at two levels with six and three clusters, respectively. The cluster that contains TM 596 440 has a size of 137, but just 17 different behaviors. Figure 7 shows an example of each kind of behavior. We can observe symmetries and growing patterns in the evolution for every TM. Although we could not reduce our list to the 14 referred to by Wolfram, we can say that our set is worth consideration.

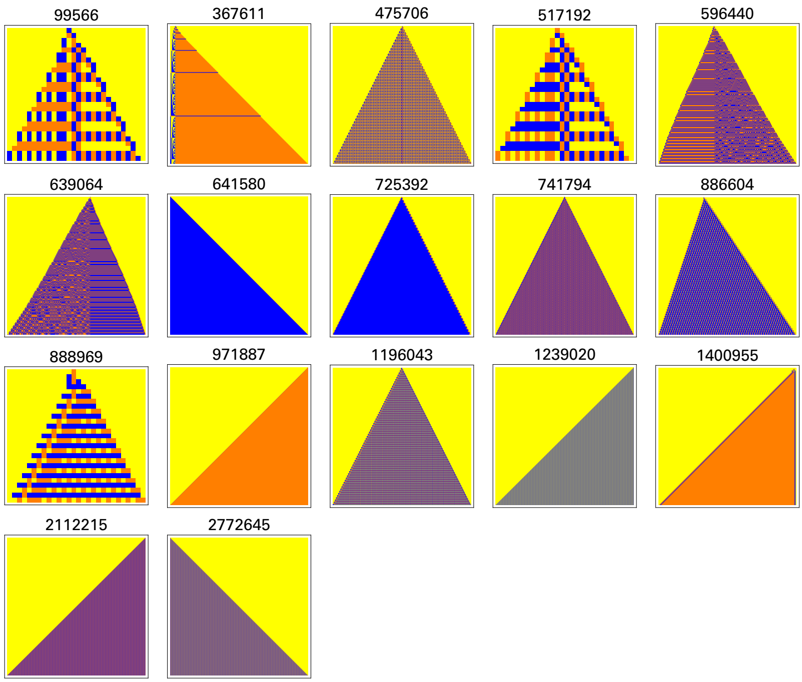

Figure 7. Sample of 2-state, 3-color TMs in the same class as the universal Turing machine 596 440.

4. Concluding Remarks

We followed a machine learning–based approach that successfully allowed us to build a classification for Turing machines (TMs). Because our problem was an unsupervised machine learning task, we had to rely on the Silhouette Validity Index as an assessment measure. We found that optimal clustering is achieved with nine clusters, which achieved the higher Silhouette Validity Index on all tests.

Mathematica's ClusterClassify function was the core tool for our approach. By providing a high-level interface for classifier building, ClusterClassify uses clustering to find a classification. In our case, classification was done over a set of features that describes TM evolution.

Using a dendrogram to capture relationships between clusters, we were able to analyze relationships between clusters. The dendrogram for the optimal clustering led us to propose a qualitative classification of four classes for TM behavior.

Although we focused our classifier building on 3-state, 3-color TMs, we demonstrated that our classification is extensible to other TM configurations, keeping an acceptable quality.

Acknowledgment

The development of this work was based on an idea from Stephen Wolfram, Todd Rowland and Christopher Wolfram, as part of the author's participation in Wolfram Summer School 2016.

References

[1] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1–2), 1984 pp. 1–35. doi:10.1016/0167-2789(84)90245-8.

[2] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, **s2-42**(1), 1937 pp. 230–265. doi:10.1112/plms/s2-42.1.230.

[3] H. Zenil, "Compression-Based Investigation of the Dynamical Properties of Cellular Automata and Other Systems," *Complex Systems*, **19**(1), 2010 pp. 1–28. www.complex-systems.com/pdf/19-1-1.pdf.

[4] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[5] H. Zenil and E. Villarreal-Zapata, "Asymptotic Behavior and Ratios of Complexity in Cellular Automata," *International Journal of Bifurcation and Chaos*, **23**(9), 2013 1350159. doi:10.1142/S0218127413501599.

[6] P. Letourneau. "Hunting for Turing Machines at the Wolfram Science Summer School," from *Wolfram Blog*—A Wolfram Web Resource. (Jul 19, 2017) blog.wolfram.com/2012/12/20/hunting-for-turing-machines-at-the-wolfram-science-summer-school.

[7] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, **20**(1), 1987 pp. 53–65. doi:10.1016/0377-0427(87)90125-7.

[8] C. N. S. Kumar, K. N. Rao, A. Govardhan and D. S. Reddy, "Imbalanced K-Means: An Algorithm to Cluster Imbalanced-Distributed Data," *International Journal of Engineering and Technical Research*, **2**(2), 2014 pp. 114–122.
www.erpublication.org/published_paper/IJETR021251.pdf.

[9] E. W. Weisstein. "Mersenne Number" from MathWorld—A Wolfram Web Resource. mathworld.wolfram.com/MersenneNumber.html.

[10] S. Wolfram. "The Prize Is Won; The Simplest Universal Turing Machine Is Proved" from *Wolfram Blog*—A Wolfram Web Resource. (Jul 19, 2017)
blog.wolfram.com/2007/10/24/the-prize-is-won-the-simplest-universal-turing-machine-is-proved/?year=2007&monthnum=10.