Predictive Collatz conjecture using classification and regressive translation models

†Gabriel Zenobi

The following study aims to make a predictive model on the patterns of the Collatz conjecture using some of the classical tools of Machine learning. Following the classification approach, a clustering analysis will be performed to code the Collatz decomposition of a sequence $\langle S(n)\rangle_{i,N\in N}^{N}$ arbitrary of order $N \to 10^{10}$.

For this purpose, we will evaluate three models, and in addition to this, a translation model is proposed with the objective of making transformations from the categorical space K to the discrete space N, and in this way to be able to carry out a regressive adjustment on the Collatz decomposition, that is, an approximation of the steps necessary to reach the cycle $4 \succ 2 \succ 1$.

Oversampling is also applied using the SMOTE technique to deal with class imbalance, and use is made of the coefficient of variation Cv in cluster analysis to complement this.

Finally, the evaluation of the models will deal with various metrics to give a broader vision of the best case to choose, and what to expect in projections on arbitrary sequences of orders of googols 10^{100}. It is important to understand that if you want to cover these large data sets with the use of multiple models, an immense computational power provided only by a supercomputer becomes mandatory.

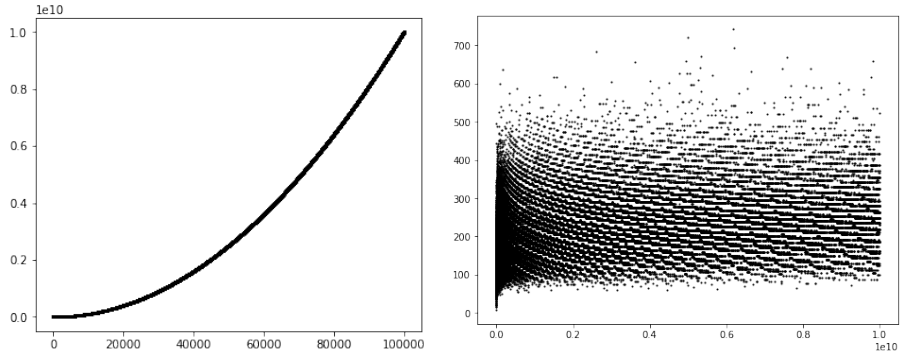
1 Introduction

Let the Collatz function be defined as

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \equiv 0 \pmod 2 \\ 3n+1 & \text{if } n \equiv 1 \pmod 2. \end{cases}$$

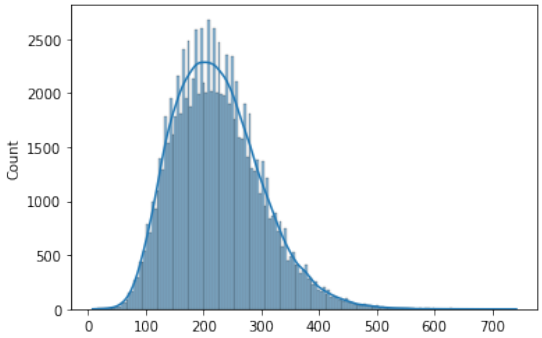
We propose the recurrence relation $F(n) \equiv F_{n+1} = F_n + i * \lambda$ where $i = 1, 2, 3, .., n$, λ is the slope or factor of increase and F_0 the initial conditional of the system. Thus, we generate the sequence $\langle F(i)\rangle_{i,N\in N}^{N}$ to study the behavior of the Collatz decomposition defined as the function $\zeta(S)$, on the proposed relation $\langle \zeta(F(i))\rangle_{i,N\in N}^{N}$ when $n \to \infty$.

Starting with the parameters $F_0 = 19$, $\lambda = 1$ the behavior of F_{n+1} resembles that of an exponential function for $n \to 10^{10}$, this It can be seen in the figure on the left.



In the image on the right we see the pattern of the Collatz decomposition provided by $\langle \zeta(F(i)) \rangle_i^N$.

On the other hand, the distribution of this pattern can be seen in the following figure.



In this way D is the dataset generated by the sequence, the Collatz decomposition ζ and the vector of categorical variables (or classes). Let's define this as

$$D = \left[\langle F(i) \rangle_{i,N\in N}^N \quad \langle \zeta(F(i)) \rangle_{i,N\in N}^N \quad |\mathrm{C}| \right] .$$

2 Cluster analysis

Making use of the K-Means algorithm we will iteratively vary the hyperparameter K to study the behavior of the objective function.

$$J = \sum_{i=1}^{K} \sum_{j=1}^{C_j} ||x_i - \mu_j||^2$$

–Also called function of sum of squared errors (SEE) or Distortion– In this way, I will define the following algorithm as VCIG (variational cluster inertia generator), which we will use to achieve an optimal variation in the SEE and the proposed metric Cv or coefficient of variation to support a lower class imbalance given a parameter K_{max}.

We must not lose sight of the fact that as more centroids are added, the amplitude of the data around these centroids will also decrease, therefore there is a natural trade-off between Cv and SEE that must be chosen carefully.

Algorithm 1: Inertia generator to study the variability in Collatz decomposition patterns

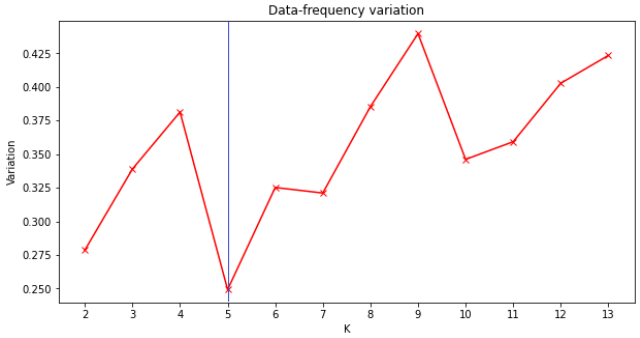
Input : D is the dataset that contains the sequence in X and the Collatz decomposition in Y.
K_{max} maximum size of clusters to iterate

Output: $S, V, L,$
$\langle min_{cv}, min_{k-cv}, min_{k-index} \rangle$ Vector of information about the best hyperparameter according to the minimum variation

1 VCIG (D, K_{max});
$D' = MinMaxScaler(D)$ ▷ We first do a scale transformation of the data.
for $k = 2$ *to* K_{max} **do**
  $model \leftarrow KMeans(n_{clusters} = k)$
  $D'_y = model.labels$ ▷ At each step we modify the classes of the Dataset to compute new calculations.
  $S \leftarrow model.inertia$ ▷ Inertia of K Means with hyperparameter k
  $L \leftarrow model.labels$ ▷ Space of classes or categories
  $V \leftarrow Cv_f(D')$ **if** $k > 2$ **then**
    $min_{cv} = V$
    $min_{k-cv} = k$
    $min_{k-index} = k - 2$
  end
end

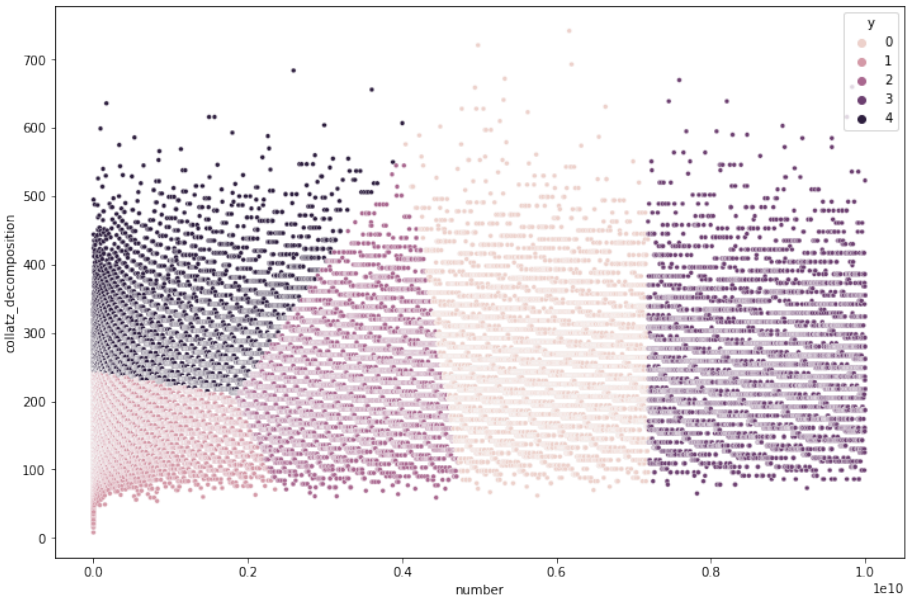
The function $Cv_f(D)$ is in charge of computing in each iteration of the K-means algorithm the frequency of the data in each cluster, and saving this information in a vector $\vec{V}$, of K components, where this number represents the total number of clusters. To know the variability of $\vec{V}$, the function must calculate $Cv = \frac{\sigma}{\mu}$, which is the coefficient of variation proposed to determine the imbalance between classes.

On the other hand, the tri-vector $\vec{Cv_{min}} = \langle min_{cv}, min_{k-cv}, min_{k-index} \rangle$ is useful to automatically obtain the index of minor variation and its associated hyperparameter K. In this way we will complement the Elbow selection method with the lowest coefficient of variation obtained in the simulation of the inertia function. The following image exemplifies the imbalance curve according to the increase of the hyperparameter K.



Keep in mind that this curve will vary depending on the sequence function used, its initial condition, and the size of the data used. For this reason, it is suggested to complement the analysis using a stochastic simulation algorithm, such as Monte Carlo or even an evolutionary-inspired method (Genetic algorithms, SWARM, differential evolution, among others...) to obtain a confidence interval on the values of $\vec{Cv_{min}}$. Also, as we take larger K the number of samples in each cluster will be smaller due to the reduction approach used by KMeans (so that if $K = N$ the number of data per cluster will be equal to 1 and therefore $Cv = 0$), so it is obvious to expect that when $K \to N$, then $Cv \to 0$.

† As an example of visual reference of the decomposition function ζ on the arbitrary sequence, the following figure provides the patterns obtained by the method and algorithm described (by taking larger K the color gradient will be greater, and less otherwise).

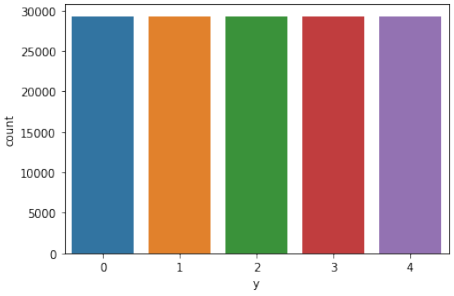


y represents the clusters and also the gradients.

3 SMOTE(Synthetic Minority Over-sampling Technique)

Through this technique we seek to balance the minority classes with the majority classes, in this way we achieve a necessary balance to avoid interpretation biases in the evaluation and validation of the models to be used.

This article does not seek to explain this technique in detail, since the objective is to make a predictive model using these tools, emphasizing the importance of a well-balanced training and validation set. Further development would involve using Monte Carlo methods to achieve a higher level of confidence over multiple generated data sets.



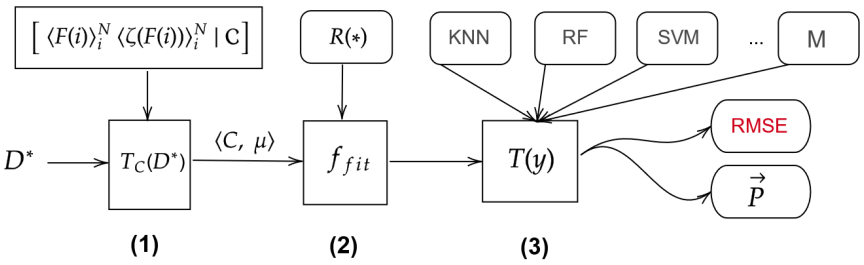
Class balance after applying SMOTE on dataset D.

The new dataset generated after applying the SMOTE technique is the matrix defined as

$$D^* = \left|\langle F(i)\rangle_{i,N\in N}^{N} \quad \langle \zeta(F(i))\rangle_{i,N\in N}^{N} \quad \mathrm{C}\right|\!|.$$

4 Collatz model-transformer

This is my association model proposed to "translate" the category space K to the discrete space N, and thus be able to perform a complementary predictive analysis on the error of the models used. Below is a diagram of the transformer model, followed by the steps to be executed.



Steps.

(1) It is the function $T_C(D^*)$ in charge of transforming the Dataset D^* into a new dataset $\langle C, \mu\rangle$. With which to adjust the model.
(2) F_{fit} regressive fit function that represents the internal model used to train the data set transformed in (1). It trains on the steps or Collatz decomposition of the arbitrary sequence.
(3) $T(y)$ translates the vector of predictions made with some other model (KNN, RF, SVM, ...), to the discrete space N, that is, the regressive space of the step (2).

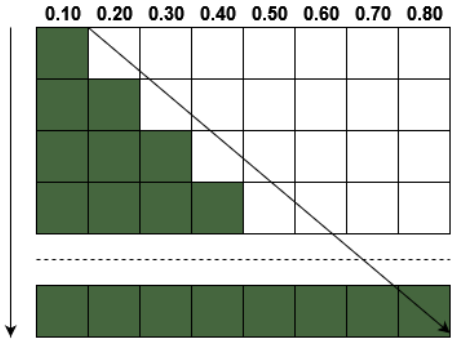
- In this study, the KNN was used in its regressive version, this would be the adjustment function that occupies or replaces step (2).
- RMSE(Root mean square error) $= \sqrt{\frac{\sum_{i=1}^{\infty} ||y_i - \hat{y}_i||^2}{N}}$ is the mean square error metric that computes the distance separating the predictions from the actual values measured using the Euclidean distance.
- $\vec{P}$ is the translation vector that represents the steps or Collatz decomposition.

From here, if a broader comparative analysis is desired, replacement models should be used (always in their continuous version).

Important! The data transformation performed by the model in regressive space will have the form: $\langle C, \mu \rangle = \begin{bmatrix} Cluster_0 & \mu_1 \\ Cluster_1 & \mu_2 \\ ... & .. \\ Cluster_K & \mu_K \end{bmatrix}$

5 Multi-modeling and assessment

Using the SMOTE D^* dataset from **(4)**, three models will be evaluated along with the proposed additional translation model. To do this we will take several instances of the data set D^* , and iteratively (and linearly) we will perform a training-validation to calculate multiple predictions in each block of different size. In this way we will be able to demonstrate the increase in the RMSE when $N \to \infty$ for ζ, that is, the Collatz function applied to an arbitrarily large sequence (according to our study, the recurrence relation with exponential behavior)



Variation in the training-validation split

The following algorithm summarizes this behavior.

Algorithm 2: Evaluation of multiple models using various metrics; Precision, Recall, Accuracy, F1 Matthews, RMSE...

Input : $\vec{x}, \vec{y}$ preferably balanced dataset, where $\vec{x}$ contains the arbitrary sequence, in our study provided by the recurrence relation $\langle F_{n+1} \rangle$, and $\vec{y}$ is the vector of class predictions.
I initial size of the training set.
$|step|$ size of the step function to split the set.
$|step_{max}|$ maximum size of the step function, or stopping criteria.

Output: $Metrics$ matrix of metrics of all the models provided throughout the evaluation.

1 MultimodelAssesment $(\vec{x}, \vec{y}, Models, I, |step|, |step_{max}|)$;
$\text{Init}_{matrix(\cdot)}(Metrics)$ ▷ **(0)**
$s = 0$
while $s \leq |step_{max}|$ **do**
  $\vec{x'}$, $\vec{y'}$ = Split($\vec{x}$, $\vec{y}$, $test_{size}$=$|step|$)
  ct = CollatzTransfomer($\vec{x'}, \zeta(\vec{x'}), \vec{y'}$) ▷ **(1)**
  $ct.fit(\dots)$
  $\text{Init}_{vector(\cdot)}(\vec{m})$ ▷ **(2)**
  for $j = 0$ *to* $|Models|$ **do**
    $\vec{s^*} = \rho(Models_m(\vec{x'}, \vec{y'}), ct)$ ▷ **(3)**
    $m_j \leftarrow \vec{s^*}$
  end
  $Metrics \leftarrow \vec{m}$ ▷ **(4)**
  $s = s + |step|$
end

The steps to explain in Algorithm 2 are the following:

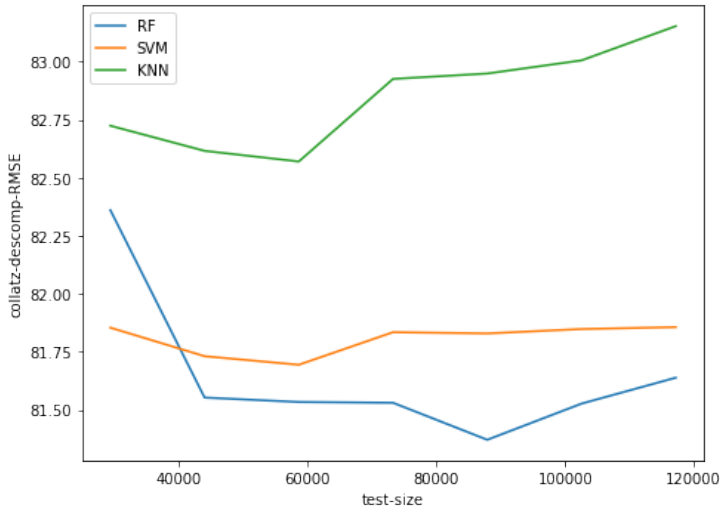
(0) Initialize the main matrix in charge of storing the metrics, an access to it will be of the form $Metrics_{\langle i,j \rangle}$, where the row i contains the vector of vectors of metrics, and j the metric vector of the j-th model. Let's take into account that in a language like python this is done dynamically, therefore we treat lists, or previously initialized numpy arrays if we want to perform space optimization.
(1) Train the translator model according to the selected regressive model, in this study we treat the KNN. If you want to change this dynamically, it is necessary to use replacement policies to adjust new models and compute their respective metrics.

(2) Initialize a new vector in each iteration, keep in mind that in a language like python this is done dynamically, therefore we treat lists, or previously initialized numpy arrays if we want to optimize space.
(3) Compute the vector of metrics of the m-model and additionally provide as argument, the translation model to internally evaluate the regression metrics (RMSE).
(4) Add a new row $i + 1$ to the metrics array, where i represents the index of the ith split; Although this variable does not appear in the algorithm, we can represent it theoretically to understand the level of evaluation we are at (this loop ends when s reaches $|step_{max}|$). Let's take into account that in a language like python this is done dynamically, therefore we treat lists, or previously initialized numpy arrays if we want to perform space optimization.

Other considerations to take into account are:
- $\text{Init}_{matrix(\cdot)}$ and $\text{Init}_{v}ector(\cdot)$ are theoretical functions that initialize a vector and/or matrix in a dynamic or pre-defined way, this depends on the language as previously indicated (in python we could define them using lists).
- Split is the function that divides the dataset in training-evaluation according to the parameter $test_{size}$.
- CollatzTransfomer is the previously described translation model.
- ρ is the theoretical function that computes a series of previously defined metrics (Precision, Recall, Accuracy, F1 Matthews, RMSE), the return of said function is a vector containing all this information in its components.

Finally the results obtained were

	accuracy	precision	recall	f1	matthews-coef	collatz-descomp-RMSE	test-size(%)	test-size	model
0	0.825021	0.823510	0.825021	0.823282	0.781829	82.360590	0.2	29295	RF
1	0.826867	0.822806	0.826867	0.824345	0.783841	81.553318	0.3	43943	RF
2	0.826318	0.822208	0.826318	0.823821	0.783126	81.534324	0.4	58590	RF
3	0.826552	0.822203	0.826552	0.823505	0.783673	81.530899	0.5	73238	RF
4	0.825602	0.821855	0.825602	0.822046	0.782990	81.371614	0.6	87885	RF
5	0.825871	0.821627	0.825871	0.822917	0.782797	81.527216	0.7	102533	RF
6	0.825781	0.822387	0.825781	0.823398	0.782599	81.638904	0.8	117180	RF
7	0.825021	0.822124	0.825021	0.823405	0.781364	81.854369	0.2	29295	SVM
8	0.826548	0.823637	0.826548	0.824919	0.783274	81.731356	0.3	43943	SVM
9	0.826318	0.823526	0.826318	0.824771	0.782973	81.694594	0.4	58590	SVM
10	0.826402	0.823478	0.826402	0.824792	0.783077	81.834751	0.5	73238	SVM
11	0.825875	0.823145	0.825875	0.824394	0.782403	81.829471	0.6	87885	SVM
12	0.825373	0.822571	0.825373	0.823855	0.781776	81.848334	0.7	102533	SVM
13	0.824953	0.822002	0.824953	0.823314	0.781278	81.856671	0.8	117180	SVM
14	0.827752	0.827174	0.827752	0.827392	0.784732	82.723841	0.2	29295	KNN
15	0.827504	0.826632	0.827504	0.827017	0.784408	82.615583	0.3	43943	KNN
16	0.826660	0.825833	0.826660	0.826215	0.783340	82.569754	0.4	58590	KNN
17	0.824545	0.823707	0.824545	0.824098	0.780693	82.925361	0.5	73238	KNN
18	0.821062	0.820056	0.821062	0.820541	0.776335	82.948558	0.6	87885	KNN
19	0.817668	0.816377	0.817668	0.816989	0.772101	83.005156	0.7	102533	KNN
20	0.814405	0.812645	0.814405	0.813460	0.768039	83.152574	0.8	117180	KNN



-As we see in the figure, the RMSE increases as we expand the validation set, or in other words; when our projections tend to infinity the Collatz decomposition ζ over the sequence increases, deviating more and more from the fitted regressive model.

Considerations; To establish how much the increase in RMSE is at infinity, sequences out of sample even larger than those proposed must be taken (of orders of googles, that is, from 10^{100} onwards), for this it is necessary an immense computational power since the complexity of time will grow as much as the complexity of space. Although in the first instance the increase of ζ is small, we cannot trust that the regressive translation model will remain stable forever, so this error will grow (let's say on a percentage scale) more or less depending on the model used, for This I suggest using a more sophisticated technique such as Stacking and/or the use of fuzzy logic to support the predictive consensus of multiple models.

The question with which this study concludes is to find out the form of the incremental error to delimit this measure in a more stable way and thus propose a general model that encompasses an immense amount of Collataz numbers.

References

[1] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, *Introduction to Data Mining*

[2] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The elements of statistical learning data mining, inference, and prediction. Second Edition*

[3] Tom M. Mitchell. *Machine Learning*

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*

[5] Dean Abbott. *Applied Predictive Analytics*