Classification of the behavior of Voxel-Based Soft Robots by means of Unsupervised Learning

Francesco Andreuzzi*, Luca Filippi†

January 4, 2022

1 Introduction

Voxel-based Soft Robots (VSR) are a form of robot made up of deformable soft cubes called *voxels* [4]. They are defined by a *morphology* and a *controller*. The morphology is the arrangement in space of the voxels; the controller determines the value of the control signal to be applied to each voxel [1]. We consider VSRs represented by a 2-D grid made up of a set of connected voxels [3].

We are going to use an unsupervised learning approach to find common behaviors in a set of VSRs. In particular we are going to employ the clustering algorithm KMeans [2] coupled with a tailored weighting approach [5].

2 Dataset

The first phase of the work consisted in the examination of the dataset and of the information at our disposal. We found works in literature regarding the usage of a particular set of features to determine a solution for our problem [4], even though the method used was different than what we are going to show in this report. Therefore, we decided to select the features containing the spectrum of the center of mass in both x and y axes, and the information about the *main gait* [4]. In particular we considered the feature `avg.touch.area`, which we believed to be important to discriminate between different gaits.

In order to be able to verify our results quickly, we dropped all the robots of the dataset except for the best individuals resulting from a set of iterations. This choice was determined by the fact that we already had a set of video files showing the behaviors of these VSRs. Therefore, the number of VSRs taken into account is $d := 10 \times 2 \times 4 = 80$ since the dataset comprised 10 seeds, 2 possible shapes of the VSR and 4 possible training terrains.

From now on we are going to refer to the spectrum of the center of mass of the VSRs in our dataset as $\mathbf{S} \in \mathbb{R}^{d \times s}$, where s is the number of columns in

*Solution design, Solution development, Data collection, Writing
†Problem statement, Solution design, Data collection, Writing

the dataset dedicated to those features; moreover, we are going to refer to the column vector containing the values of the feature `avg.touch.area` as $\mathbf{A}_t \in \mathbb{R}^d$.

Features in $\mathbf{S}$ are normalized (i.e. scaled and translated in orde to have mean zero and unit variance) in order to improve the results of our approach [2]. The transformation is carried out in a non-standard way, namely we transform the features as if they were one single (long) feature in order to preserve differences between frequency buckets, which may be important for the classification.

3 The proposed approach

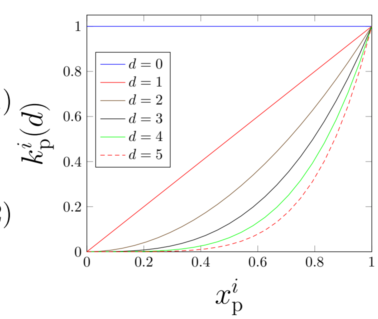
We try to solve the problem applying the algorithm KMeans [2] on the features $\mathbf{S}$ and $\mathbf{A}_t$. The central point of this work is discussed briefly in Section 3.1, and consists in using several coefficients (which are functions of features which we are not going to use in KMeans) to tailor the influence of each component of $\mathbf{A}_t$. In Section 3.2 we are going to show how we modified the dataset thereafter.

3.1 Coefficients

The discussion presented in this section is mostly based on the information extracted from the *main gait*, which collects both qualitative and quantiative data about the behavior of a VSR [4]. We motivate briefly each coefficient.

$$k^i_{\mathrm{mi}} = \left(\frac{x^i_{\mathrm{mode\_interval}}}{\max_j x^j_{\mathrm{mode.interval}}} \right)^{-1} \in [1, +\infty] \quad (1)$$

$$k^i_{\mathrm{p}}(d) = (x^i_{\mathrm{purity}})^d \in [0, 1] \quad (2)$$



The first coefficient taken into account is $\mathbf{k}_{\mathrm{mi}}$, whose d components are defined as in (1). The rationale behind (1) is the observation that as $x^i_{\mathrm{mode\_interval}}$ approaches zero the main footprints become increasingly important to reconstruct the behavior of the robot. The coefficient is also "normalized" against the maximum mode interval in the dataset.

If the mode of the time intervals between two consecutive occurrences of the most prevalent n-gram of footprints does not reoccur massively, the measure computed in (1) is not truly expressive of the importance of the main footprints. For this reason in (2) we take into account the *purity* of time intervals. Since $x^i_{\mathrm{purity}} \leq 1$, multiplying something by $k^i_{\mathrm{p}}(d)$ causes a non-increasing scaling which depends on how dominant (i.e. frequent) is the mode interval. We can experiment multiple relations as d varies (see the figure on the right).

We now look at the significance of the number of footprints in the main gait. If `n.footprints` is high, we could deduce that the main gait is very informative of the behavior of the robots. By contrast, if we were able to extract only a small number of footprints we deduce that the behavior of the robot is rhapsodic, therefore the main gait provides a low-quality approximation. Building on these considerations we compute the following coefficients:

$$k^i_{\text{nf}} = \frac{x^i_{\text{n.footprints}}}{\max_j x^j_{\text{n.footprints}}} \in [0,1] \quad (3) \qquad k^i_{\text{nuf}} = \frac{x^i_{\text{n.footprints}}}{x^i_{\text{n.unique.footprints}}} \in [1,+\infty] \quad (4)$$

The coefficient k^i_{nf} computed in (3) provides a measure of the quantity of information that the main gait of the i-th VSR provides about the behavior of the robot. We may take the considerations above even further, observing that the behavior of a robot may be considered rhapsodic, on a lower scale, if the number of *unique* footprints in the main gait is high. Equation (4) expresses this concern, quantified by k^i_{nuf}. This coefficient is high for robots whose main gait contain a small number of unique footprints, i.e. for robots whose behavior may be well approximated with a small number of (unique) footprints.

The reader may notice that our coefficients do not live on the same domain. This does not mean that we are promoting some coefficients more than others: in our experiments we considered multiple combination of multipliers for each coefficient in order to find the optimal one. Some of them may even vanish in the optimal configuration.

We define the vector of tailored weights as follows, $w_{\text{mi/nf/nuf}} \in \mathbb{R}$ in (5) are multipliers (or weights) used to tune the influence of $k^i_{\text{mi/nf/nuf}}$:

$$\mathbf{k}_{\text{gait}}(w_{\text{mi}}, w_{\text{nf}}, w_{\text{nuf}}) = \left[w_{\text{mi}} k^i_{\text{p}} k^i_{\text{mi}} + w_{\text{nf}} k^i_{\text{nf}} + w_{\text{nuf}} k^i_{\text{nuf}}\right]_{i=0}^{d} \tag{5}$$

3.2 Modified dataset

Employing (5) we modify the dataset as follows:

$$\widetilde{\mathbf{X}} := \left[\widetilde{\mathbf{A}}_t, \quad \mathbf{S}\right] \in \mathbb{R}^{d\times(s+1)} \tag{6}$$

where $\widetilde{\mathbf{A}_t} := \mathbf{A}_t \odot \mathbf{k}_{\text{gait}} \in \mathbb{R}^d$ (the symbol "$\odot$" denotes the Hadamard product, i.e. the element-wise multiplication of two vectors). Note that we dropped the dependency of $\mathbf{k}_{\text{gait}}$ on the weights $w_{\text{mi/nf/nuf}}$ to lighten the notation, however as we anticipated in Section 3.1 in our experiment we tested multiple values for those weights in order to find the best combination.

4 Results

We present briefly the experiments carried out in this work, before drawing our conclusions regarding the approach presented in Section 3.

4.1 Testing clusterization

In order to validate our approach we watched the video files mentioned in Section 2 and produced a testing clusterization *by hand*. We identified 4 clusters, which we labeled in the following way: *walking*, *crawling*, *jumping*, *rolling*. It might be interesting to experiment with different numbers of clusters though, in order to tune the results given by KMeans [2].

4.2 Analysis of the results

We compared the output of our algorithm with the testing dataset, using different combinations of weights for the coefficient as we mentioned in Section 3.1. Then we took the combination that gave the lowest number of misclassifications. As we can see in Figure 3 we improved slightly the results, in particular inside the cluster *jumping*. This confirms our guess that some clusters are more sensible than others on the feature `avg.touch.area`. We observed that the optimal values of the weigthts mentioned in (5) are:

$$w_{\text{mi}} \approx 1.5 \qquad w_{\text{nf}} \in [0, 0.5] \qquad w_{\text{nf}} \in [0.1, 0.2] \qquad d = 1$$

Therefore we infer that the most influential coefficient among those presented in Section 3.1 is k^i_{mi}, with a preferable linear dependency on `purity`.

5 Conclusions

The results produced by our classifier are not very accurate as evidenced by the high number of classification errors. However, the effect of the approach proposed in Section 3 is quite evident as we highlighted in Section 4. We may explain the classification errors with likely mistakes in the testing dataset (e.g. deciding if a VSR it walking or crawling is usually not trivial), and with the fact that two different shapes of robots are taken into account.

The code for the experiments is available in this GitHub repository. We used Python, scikit-learn, Pandas and NumPy for the implementation.

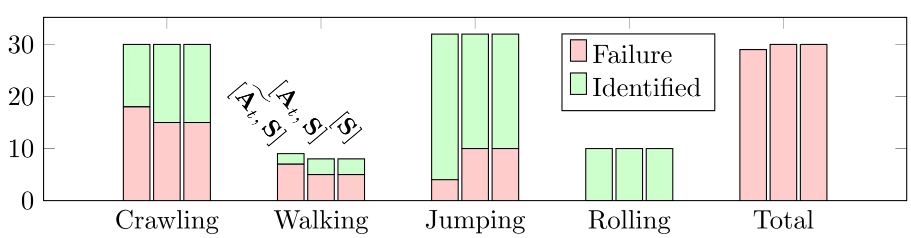


Figure 3: Number of failures/successes for KMeans applied on three different sets of features. The leftmost of the three columns is obtained by means of the method presented in Section 3; the center column is obtained using the features $\mathbf{S}$ and $\mathbf{A}_t$ (i.e. no tailored weighting for `avg.touch.area`); the rightmost column is obtained using only the features inside $\mathbf{S}$.

References

[1] Andrea Ferigo, Giovanni Iacca, and Eric Medvet. Beyond body shape and brain: evolving the sensory apparatus of voxel-based soft robots. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pages 210–226. Springer, 2021.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. An introduction to statistical learning. 2009.

[3] Eric Medvet, Alberto Bartoli, Andrea De Lorenzo, and Stefano Seriani. Design, validation, and case studies of 2d-vsr-sim, an optimization-friendly simulator of 2-d voxel-based soft robots. *arXiv preprint arXiv:2001.08617*, 2020.

[4] Eric Medvet, Alberto Bartoli, Federico Pigozzi, and Marco Rochelli. Biodiversity in evolved voxel-based soft robots. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 129–137, 2021.

[5] Deepak Panday, Renato Cordeiro de Amorim, and Peter Lane. Feature weighting as a tool for unsupervised feature selection. *Information processing letters*, 129:44–52, 2018.