Automatic Demonstration and Feature Selection for Robot Learning

Santiago Morante, Juan G. Victores and Carlos Balaguer

***Abstract*— Robot learning frameworks, such as Programming by Demonstration, are based on learning tasks from sets of user demonstrations. These frameworks, in their naïve implementation, assume that all the data from the user demonstrations has been correctly sensed and can be relevant to the task. Analogous to feature selection, which is the process of selecting a subset of relevant features for use in model construction, this paper presents a demonstration selection process, which is additionally applied for feature selection for further data filtering.**

The demonstration and feature selection process presented is called Dissimilarity Mapping Filtering (DMF). DMF involves three steps: obtaining a measurement of dissimilarity (e.g. Dynamic Time Warping, etc.), reducing dimensions through a mapping algorithm (e.g. sum of dissimilarities, Multidimensional Scaling, etc.) and a filtering method (z-score based, DBSCAN, etc.). As a demonstration selector, DMF discards outlying demonstrations in terms of all the features considered simultaneously. As a feature selector, DMF discards features that present high inconsistency among demonstrations. We apply DMF to our Continuous Goal-Directed Actions (CGDA) robot learning framework presented in previous works.

I. INTRODUCTION

Robot learning frameworks provide a way for non-expert users to teach robots new actions. These frameworks share two key factors: (1) the user demonstrations are the main input to the learning framework, and (2) they internally generate a generalized model of the task. A demonstration selection process to filter bogus performances or incorrectly sensed user demonstrations can be incorporated as an initial step in any of these frameworks.

Regarding the generalized model of the task, Programming by Demonstration [1] conventionally encodes a generalized action as a probabilistic model in the robot joint or operational space. Dynamic Movement Primitives [2] encode a generalized action as a control policy, generally in the operational space. Continuous Goal-Directed Actions (CGDA) [3] encodes a generalized action as a feature trajectory preserving all the scalars that can be extracted from the sensor data at each instant. Attending to the space where the generalized model is stored, a handcrafted feature selection process is implicitly performed when defining the structure of the generalized model of the task.

While reducing an action to the joint or operational space is a clear over-simplification for many use cases (e.g. filling a glass depends on the layout of the environment), preserving all the scalar features that can be extracted from the sensor data can lead to not knowing which feature is relevant for the task. For instance, a person fills a glass with water by pouring it from a bottle. Which are the features that are relevant for the task? The area of the glass that is perceived with a slightly different color due to the new refraction index? The absolute or relative position and orientation of the bottle? Reproducing the sound of a motorcycle that was passing by during one of the demonstrations?

The Dissimilarity Mapping Filtering (DMF) process presented in this paper is used both as a demonstration selector and as a feature selector. We work in a robotics learning scenario, using a humanoid robot equipped with simple 3D machine vision, learning a block-moving task (see Fig. 1).

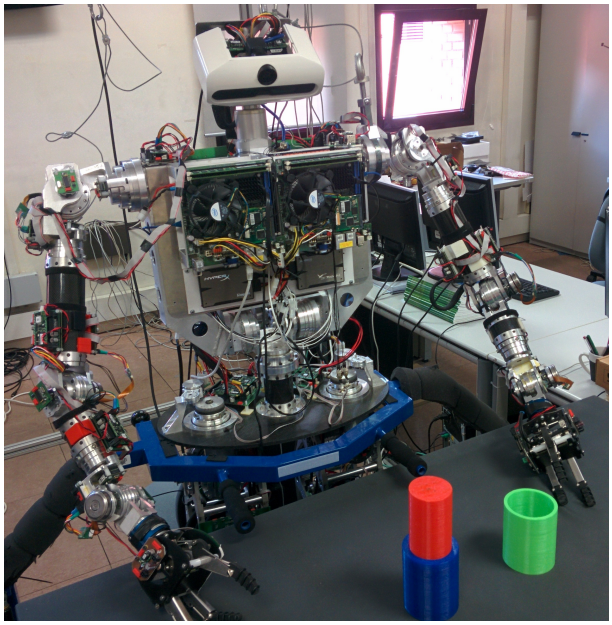

Fig. 1. Humanoid robotics learning scenario: block-moving task.

All of the experiments are completely reproducible, as we are aware of the importance and concerned about the reproducibility of research. Our project code[1] and its associated tools, experiments and results are publicly available, and have been open-sourced as a further guarantee on reproducibility.

II. STATE OF THE ART

The general problem of selection of demonstrations and features can be decomposed into three subparts: different duration of user demonstrations, demonstration selection, and feature selection.

All of the authors are members of the Robotics Lab research group within the Department of Systems Engineering and Automation, Universidad Carlos III de Madrid (UC3M). smorante@ing.uc3m.es

[1] https://github.com/smorante/continuous-goal-directed-actions

A. Different Duration of User Demonstrations

One of the problems that arises when comparing user demonstrations is their different duration. It is tricky to compare sets of multidimensional signals of different durations without losing too much information, as scaling and warping can lead to desynchronize demonstrations and feature trajectories within demonstrations. Literature has faced this problem in different ways. Some machine learning authors force the user to perform all the demonstrations with a fixed duration [4]. Jetchev also fixes the duration [5] but claims that different duration demonstrations can also be handled. Mühlig uses Dynamic Time Warping (DTW) on each individual feature when comparing the signals [6][7]. Statisticians face similar problems when comparing temporal series. Some techniques they use are [8]: padding (filling the shortest with zeros), upsampling (inserting values), or interpolating (interpolate timestamps).

B. Demonstration Selection

When recording user demonstrations, there is no guarantee that all the demonstrations will be perfectly executed. There may be many reasons for this: human fallibility, sensor error, network latency, etc. Sometimes it is difficult and time-consuming to manually check each demonstration and each feature to find anomalies. Additionally, when recording many features at a high rate sampling, the data generated can easily overwhelm the human capacity to find deviations from the correct demonstration. Chernova [9] proposes a method to filter discrete choices in a human-robot interaction reinforcement learning framework. However, to the author's knowledge, there has been no work on automatic demonstration selection (understanding demonstrations as full trajectories to complete an action), where incorrectly performed or sensed user demonstrations are discarded.

C. Feature Selection

Many possible features may be extracted from sensor data for each task, but not all of them may be relevant for a specific task. A feature selector can automatically discard features that are irrelevant for a given task. While most robot learning frameworks are provided only the relevant features considered by their designer (i.e. only joint angle values or operational space coordinates), in certain literature more features are fed to the algorithms, some of which are automatically selected.

In [10], the features used for encoding the task are the robot joint angles, the user hand coordinates, the location of the objects at which actions are directed, and the laterality of the motion (which hand is used). They encode the trajectories into a Hidden Markov Model (HMM) of the task for each demonstration. For feature selection, they discard features that present a high variance among HMM states. Variance is also the discarding factor in [7], where the observed movement is projected into a task-specific space and the correspondence problem is avoided by solely focussing on the object trajectories without making any assumption on the teacher's postures during the demonstration. They encode relative object positions and orientation. They use DTW to avoid the different duration problem, and they discard features by variance. The same author extended the discarding possibilities by adding and attentional factor or a energy saving (called kinetic) one [6]. They use what they call task spaces. Observed movements are mapped into a pool of task spaces and they present methods that analyze this task space pool in order to acquire task space descriptors. A selection method named task space selector analyzes the observed object trajectories and acquires task space descriptors that match the observation best. Several criteria are incorporated, such as a psychologically inspired criterion that is based on the robot's attention to the objects in the scene and a kinetic criterion that estimates effort and discomfort of the human teacher. Concerning the learning of object movements, task spaces may be composed of absolute object positions and orientations, relations between objects, additional constraints such as the restraint to only planar movements, and additional joint-level constraints. In [5] they encode the center of the target object, three fingertips, the three lower digits, the palm center and the relative distances. They remove the redundant features using correlation as a measure.

III. DISSIMILARITY MAPPING FILTERING

The Dissimilarity Mapping Filtering (DMF) process, applied as a demonstration selector and as a feature selector, is introduced in this section. An additional preprocessing step is incorporated in the descriptions prior to the dissimilarity, mapping, and filtering steps.

A. Demonstration Selector

Using DMF as a demonstration selector, the goal of the process is to automatically discard incorrectly sensed or performed demonstrations. Assuming each feature is a time-varying scalar value (e.g. joint angles, centroid position coordinate, or more arbitary features supported by the Continuous Goal-Directed Actions (CGDA) framework such as percentage of area of a certian color [11][12]), the set of user demonstrations that is fed to DMF can be treated as a set of multidimensional signals.

1) Preprocessing: First, the input data is normalized. While normalization may not be required if all the features are presented in the same units, if e.g. joint angles are used in conjunction with operational space measurements, these values must be weighted or somehow be comparable. We propose four different types of normalization, as this choice can affect the final demonstration selection results.

- MinMax: Each feature, for each demonstration is normalized within the limits of the feature:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

- Standardized: Each feature is standardized:

$$X_{norm} = \frac{X - X_{mean}}{X_{stddev}} \tag{2}$$

- Whole Experiment Normalization: Each feature is scaled by the maximum value for this feature, among all the demonstrations:

$$X_{norm} = \frac{X}{X_{maxExperiment}} \tag{3}$$

- Physical Limits: Each feature is normalized within the physical limits of the sensor that provide the information for this feature:

$$X_{norm} = \frac{X - X_{PhysicalMin}}{X_{PhysicalMax} - X_{PhysicalMin}} \tag{4}$$

Depending on the field of study where DMF is applied, it may be convenient to use one of these normalizations, or none. The next step after the normalization is to reduce the multidimensional complexity.

We consider two different approaches for reducing each multidimensional signal into a one dimensional signal. The first approach is to sum all the features of the same demonstration into a single signal by summing all the discrete points of each feature signal, see Fig. 2. Performing the same operation on all demonstrations, we obtain a single signal per demonstration of size $1 \times M$ where M is the total number of features.

The second approach sums the demonstration into a single signal by summing all the features for each point, obtaining a signal of size $N \times 1$ where N is the total number of points of the demonstrations.

Summing rows (1xM)

t	f_1	f_2	f_3
0	56	65	9
1	55	65	3
2	57	62	2
3	53	60	0
4	54	59	1
5	52	59	1
Σ	327	370	16

Summing columns (Nx1)

t	f_1	f_2	f_3	Σ
0	56	65	9	130
1	55	65	3	123
2	57	62	2	121
3	53	60	0	113
4	54	59	1	114
5	52	59	1	112

Fig. 2. Different preprocessing methods for reducing multidimensional information.

Whichever the method selected, we have converted each user demonstration, originally a multidimensional signal, into a one dimensional signal that can be used in the next step.

2) Dissimilarity: We must obtain an absolute measurement of dissimilarity regarding each demonstration, or a relative one. This absolute or relative measurement will be analysed in the next step. DMF is a flexible process where one can select a different algorithm for each of the steps. We have selected to use Dynamic Time Warping (DTW) to obtain a distance matrix [13], but other measurements can be used (e.g. Euclidean distance, uniform scaling, etc.). DTW allows demonstrations of different durations. Synchronization issues are avoided when columns are summed in the previous step.

Let $X = \{x_1, .., x_p\}$ and $Y = \{y_1, .., y_q\}$ be two generic signals. To compare two elements, a local cost measure (a distance $dist(x, y)$ e.g. Euclidean) is needed. A lower cost represents a bigger similarity of the sequences. Evaluating all pairs of points between the signals, we obtain a cost matrix CM:

$$CM = \begin{pmatrix} dist(x_0, y_0) & \cdots & dist(x_p, y_0) \\ \vdots & \vdots & \vdots \\ dist(x_0, y_q) & \cdots & dist(x_p, y_q) \end{pmatrix} \tag{5}$$

The goal now is to find the lowest cost alignment path between the signals. This path is usually calculated in an accumulated cost matrix derived from CM, where each cell represents the distance of the correspondent pair $dist(x, y)$ plus the cost to reach this cell. In this accumulated matrix, the cost of alignment $C(X, Y)$ of the optimal path is:

$$C(X, Y) = \frac{C(x_p, y_q)}{P + Q} \tag{6}$$

We apply DTW for the comparison of each pair of demonstrations, obtaining one cost C in each comparison. With these costs, we create the distance matrix DM for all demonstrations d:

$$DM = \begin{pmatrix} C(d_0, d_0) & \cdots & C(d_R, d_0) \\ \vdots & \vdots & \vdots \\ C(d_0, d_R) & \cdots & C(d_R, d_R) \end{pmatrix} \tag{7}$$

Where R is the total number of demonstrations of the experiment. This matrix DM is symmetrical and can be used to obtain the dissimilarity among demonstrations.

3) Mapping: The main idea in this step is to reduce the information provided by the previous step into a single value per demonstration. We have selected to map the distance matrix DM into a single dimension by summing the value in the columns, but other algorithms can be used: multidimensional scaling, ISOMAP, etc. With our method, we obtain a value V for each demonstration i:

$$V_i = \sum_{j=0}^{R} C(d_i, d_j) \tag{8}$$

Each value V represents a measurement of the dissimilarity of a demonstration with respect to all the others.

4) Filtering: In this last step, the aim is to screen the demonstrations to discard the outlying ones. We have chosen to filter them by using the z-score (called standard score) because it standardizes the results, but other algorithms can be used: DBSCAN, t-test, etc. Each value V is converted into a z-score Z:

$$Z = \frac{V - V_{mean}}{V_{stddev}} \tag{9}$$

Finally, we discard the demonstrations with a z-score Z higher than a threshold α. This α is the only tunable parameter of the whole DMF process as explained.

This preprocessing, dissimilarity, mapping and filtering, can discard the incorrect demonstrations. In the next section, we will explain how to discard features that are irrelevant for a given task.

B. Feature Selector

The goal of this section is to discard those features not relevant for a specific task. We start from the raw data, using only the demonstrations selected in the previous step. What we propose in this case, is to use the same DMF process as in the previous situation.

1) Preprocessing: As in the previous case, there may be the need for normalization. The possibilities are the same as those in the previous case. However, there is no need to reduce the multidimensional complexity in this case.

2) Dissimilarity: Again we have chosen DTW for measuring dissimilarity. In this case, we aim to obtain a cost value for each dimension (each which corresponds to a time-varying scalar value of a feature). For each dimension, we calculate the dissimilarity among all demonstrations. We obtain one cost C in each comparison. With these costs, we create the distance matrix DM for all demonstrations d for each dimension similar to Eq. (7). One distance matrix is obtained per feature, as in Fig. 3.

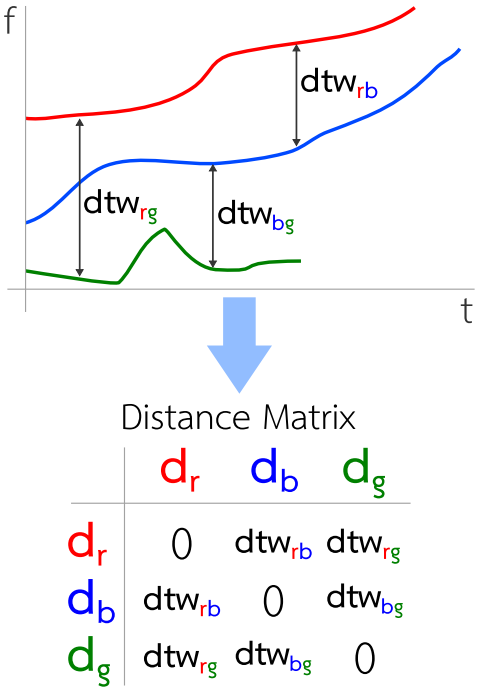

Fig. 3. DTW cost between demonstrations results in one DM per feature.

3) Mapping: The mapping in this case is the sum of all the elements of the distance matrix, resulting in a single value V_{total} that represents a measurement of the dissimilarity of the demonstrations for a given feature:

$$V_{total} = \sum_{i=0}^{R} \sum_{j=0}^{R} C(d_i, d_j) \tag{10}$$

4) Filtering: The filtering process is again performed using z-scores. Each value V_{total} is converted to a z-score Z, and the algorithm discards the features with a z-score Z higher than an α threshold.

IV. EXPERIMENTS

The first goal of the experiment is for the robot to distinguish correct and incorrect demonstrations of a given task: putting the green object on top of the red object. The second goal is for it to distinguish relevant and irrelevant features for the task.

We have used the full-size humanoid robot TEO [14]. TEO's head is equipped with an ASUS Xtion PRO LIVE set to provide 640×480 RGB and depth streams at 30 fps. The red and the green object are color segmented, as in Fig. 4. The following 13 scalar features are extracted in a periodic 40 ms loop: centroid absolute position (red object x_1, y_1, z_1 and green object x_2, y_2, z_2), centroid relative position (the difference between the centroid absolute positions $x_1 - x_2$, $y_1 - y_2$, $z_1 - z_2$), absolute values of the previous values ($|x_1 - x_2|$, $|y_1 - y_2|$, $|z_1 - z_2|$), and Euclidean distance between the red and the green object $dist(X_1, X_2)$.

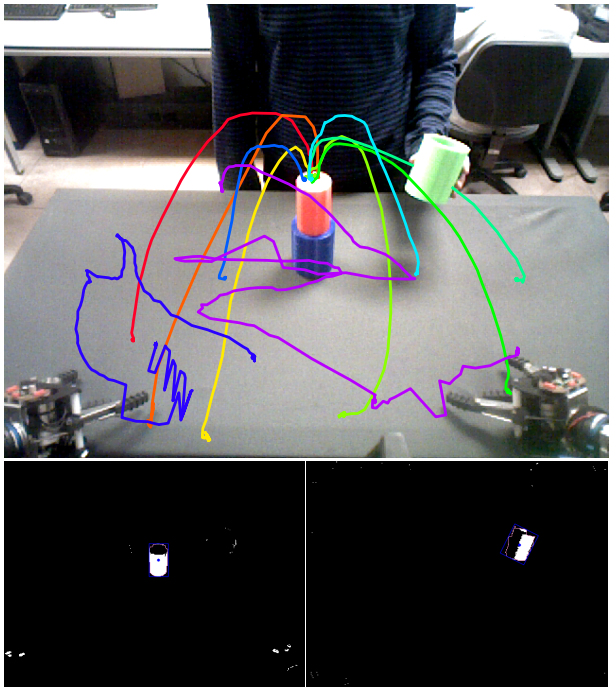

Fig. 4. User demonstrations from TEO's perspective; red segmentation on the bottom left, and green segmentation on the bottom right.

We recorded 10 demonstrations of different durations, performing 8 of them correctly, and performing the last 2 incorrectly. The red object is not moved in any of the correct demonstrations, but it is moved in the incorrect ones. The green object approaches the red object from different angles in the correct demonstrations, and is moved randomly in the incorrect ones. As humans, with this context information, we consider that the relevant demonstrations (those to be

kept), are those which are similar to most of the other, so we would discard the last two demonstrations. Regarding the features, we consider that the features that must be discarded are: x_2, y_2, $x_1 - x_2$ and $y_1 - y_2$, which are those dependent on the initial position of the green object. The rest of the variables should not be discarded: variables involving z remain constant across demonstrations, absolute value differences of the objects tend to the same values when sign is removed, and finally, the Euclidean distance always decreases for all demonstrations, as the green object is always moved closer to the red one.

A. *Results*

The spatial movements of the red object for all demonstration can be seen in Fig. 5.

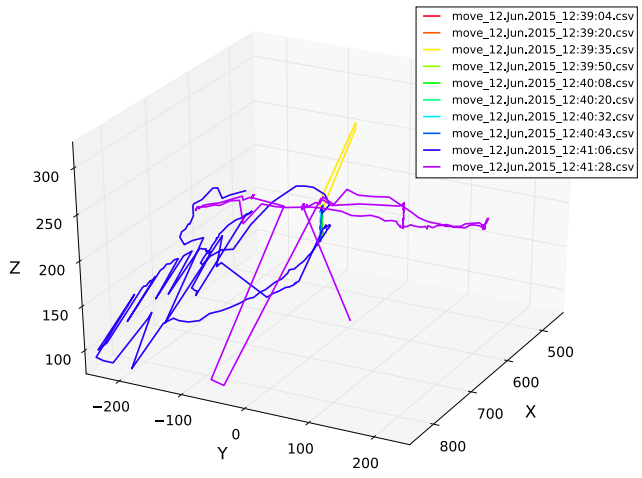

Fig. 5. Red object centroid coordinates throughout time.

There is a concentration of lines in the middle (the object remains unmoved). Additionally, there are two lines (blue and purple) which deviate from the rest. These are the incorrectly performed demonstrations. Fig. 6 depicts the spatial movements of the red object for all demonstrations.

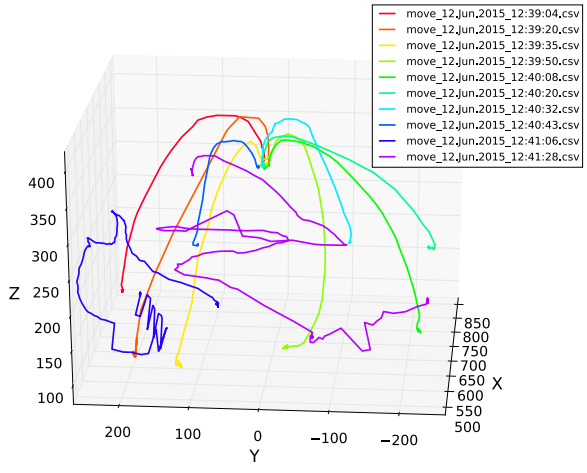

Fig. 6. Green object centroid coordinates throughout time.

In this case, for the correct demonstrations, the green object converges to the top of the red object. In the incorrect demonstrations (blue and purple), it does not converge to this point. We have overlayed the green object movement into the robot point of view, to give the reader an understanding of the plots (Fig. 4).

Once DMF is applied on the demonstrations, we plot the z-scores for the demonstrations. In the first case (Fig. 7) we apply the summed columns preprocessing, and treat the data without normalization. We also tested the summed rows preprocessing (Fig. 8), also without normalization.

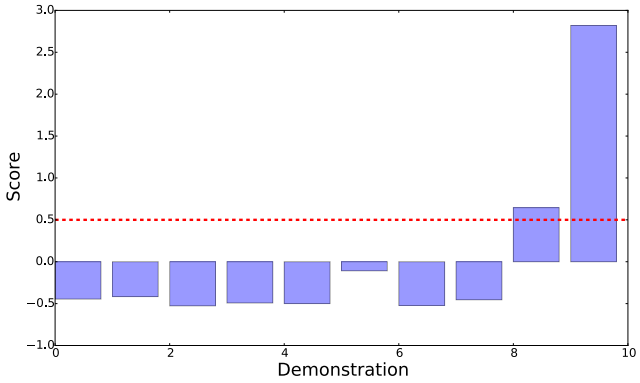

Fig. 7. Demonstration selection's z-scores without data normalization, preprocessed by summing columns, setting the threshold to $\alpha = 0.5$.

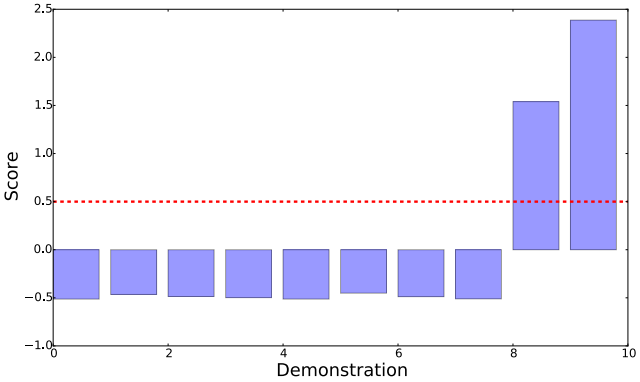

Fig. 8. Demonstration selection's z-scores without data normalization, preprocessed by summing rows, setting the threshold to $\alpha = 0.5$.

In both cases, the discarded demonstrations are those that were incorrectly performed. Once these two incorrect demonstrations were discarded, we applied the feature selection algorithm on the remaining ones. The z-scores results for the feature selection can be seen on Fig. 9. The discarded features are x_2, y_2, $x_1 - x_2$ and $y_1 - y_2$. This result agrees with our previous expectation. The data was treated without normalization. As a summary, we have gathered the correspondent z-scores, when data is normalized with several techniques, in Tab. I.

V. DISCUSSION

In our experiment, all the features were measured in the same distance units. This may not be case for other exper-

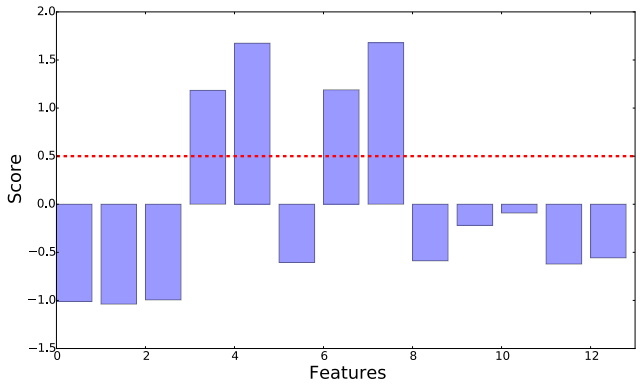

Fig. 9. Feature selection's z-scores without data normalization, setting the threshold to $\alpha = 0.5$.

TABLE I

Z-SCORES WITH NORMALIZED DATA. GREEN MEANS ACCEPTED FEATURE, RED MEANS DISCARDED ($\alpha = 0.5$).

	MinMax	Standardize	Whole Exp.	Phys. Limits
x_1	-0.004	0.301	-0.956	-1.030
y_1	0.743	0.650	-0.889	-1.077
z_1	-0.657	-1.080	-0.929	-0.985
x_2	1.223	1.209	-0.37	2.396
y_2	1.439	1.334	1.874	1.570
z_2	-1.080	-1.010	-0.728	-0.226
$x_1 - x_2$	1.166	1.201	1.670	0.662
$y_1 - y_2$	1.435	1.334	1.679	0.246
$z_1 - z_2$	-1.088	-1.032	-0.257	-0.637
$\|x_1 - x_2\|$	-0.782	-0.891	0.004	0.204
$\|y_1 - y_2\|$	-0.917	-0.977	-0.041	-0.152
$\|z_1 - z_2\|$	-0.484	-0.034	-0.440	-0.258
$Eucl.Dist$	-0.992	-0.996	-0.609	-0.711

iments, so that data normalization may be a requirement. While it may be necessary, normalization may dangerously distort the data. The following is a description derived from of our own experience upon using different methods.

- MinMax: When normalizing each feature within its empirical limits, the noise in the signal may be amplified. For instance, a flat signal having sensor noise may result in a signal where the noise has been greatly amplified.
- Standardized: Outliers may attenuate the signal. Imagine a sine wave signal with a far outlier (e.g. a sensor failure). Having outliers often produces a significant effect on the mean and standard deviation. When the signal is subtracted the mean and divided by the standard deviation, it may be completely flattened.
- Whole Experiment Normalization: The data may be distorted by both of the problems already explained.
- Physical Limits: This normalization implies that the designer has a previous knowledge of each feature limit. In some cases, the limits will be imposed be the sensor, while in other cases (e.g. derived features and combinations), mathematical limits must be determined. There is an additional issue when tasks do not expand over the whole space and are instead concentrated in a small region, and features limits have been set with large ranges. When the signal is normalized, it may be flattened, becoming no more relevant than noise.

It is an open discussion what technique is the most appropriate, and the authors believe it will depend on the learning context.

VI. CONCLUSIONS

In this paper we have presented Dissimilarity Mapping Filtering (DMF), a process for discarding irrelevant elements on a set. We have applied DMF to demonstration and feature selection in the context of a humanoid robot goal-directed learning experiment. Results show the accuracy of DMF, allowing a great flexibility with the interchangeable algorithms. Robots may leverage of the presented process when performing complicated task involving many features, which is the realm of modern humanoid robot learning.

ACKNOWLEDGMENT

This work was supported by RoboCity2030-III-CM project (S2013/MIT-2748), funded by Programas de Actividades I+D in Comunidad de Madrid and EU.

REFERENCES

[1] S. Calinon, *Robot programming by demonstration: A probabilistic approach*. EPFL Press, 2009.
[2] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*. Springer, 2006, pp. 261–280.
[3] S. Morante, J. G. Victores, A. Jardón, and C. Balaguer, "Action effect generalization, recognition and execution through continuous goal-directed actions," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1822–1827.
[4] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, vol. 97, 1997, pp. 12–20.
[5] N. Jetchev and M. Toussaint, "Task space retrieval using inverse feedback control," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 449–456.
[6] M. Muhlig, M. Gienger, J. J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 4996–5002.
[7] M. Muhlig, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 1177–1184.
[8] I. A.-A. Tarsitano, "Adjusting time series of possible unequal lengths," 2012.
[9] S. Chernova and M. Veloso, "Multi-thresholded approach to demonstration selection for interactive robot learning," in *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*. IEEE, 2008, pp. 225–232.
[10] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 299–304.
[11] S. Morante, J. G. Victores, A. Jardón, and C. Balaguer, "On using guided motor primitives to execute continuous goal-directed actions," in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*. IEEE, 2014, pp. 613–618.
[12] ——, "Humanoid robot imitation through continuous goal-directed actions: an evolutionary approach," *Advanced Robotics*, vol. 29, no. 5, pp. 303–314, 2015.
[13] T. Albrecht, "Dynamic time warping," in *Information Retrieval for Music and Motion*. Springer, 2009, pp. 69–84.
[14] S. Martínez, C. Monje, A. Jardón, P. Pierro, C. Balaguer, and D. Muñoz, "Teo: Full-size humanoid robot design powered by a fuel cell system," *Cybernetics and Systems*, vol. 43, no. 3, pp. 163–180, 2012.