International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Evaluation of Similarity Metrics Under the Context of an Autonomous Reactive System

Guillermo Toledano Villalobos[a], Salvador Ibarra Martínez[b], José Castan Rocha[c], David Teran Villanueva[d*]

[a,b,c,d]Universidad Autónoma de Tamaulipas, Tampico 89109, México
[a]Email: a2113330259@alumnos.uat.edu.mx
[b]Email: sibarram@docentes.uat.edu.mx
[c]Email: jacastan@docentes.uat.edu.mx
[d]Email: jdteran@docentes.uat.edu.mx

Abstract

Currently, in the field of robotics, institutions and researchers are working on the design and development of autonomous navigation systems on robots for dynamic environments. The most advanced implementations of autonomous behaviors are found on vehicles or wheeled devices, allowing them to move on controlled environments and even on rough terrain. In this paper, it is presented the design of an autonomous reactive system for humanoid robots. This system requires to know the current state of the robot, during a specific activity, to make the right reactive action for a specific situation. In the context of inquiring the current state of the robot, we consider the implementation of a knowledge base populated with diverse states of the joints and their possible reactive actions. To recover the possible reactive actions from the knowledge base, it is required to search for the current state of the robot in the knowledge base. However, this process may incur in high computational cost depending on the size of the knowledge base. Therefore, in this work, we carried out a comparative study of six similarity metrics, with the objective of identifying the metric that offers the best computational time. In these studies, it is identified that the metrics with lower mathematical complexity showed the best results. Additionally, we used Wilcoxon and Friedman statistics tests to assess the performance of the similarity metrics. Finally, we included an analysis of the characteristics and functionality of several similarity metrics, which showed that some of them are not suitable in the context of our proposal. On the other hand, other metrics were identified as viable and with potential for future works.

Keywords: Autonomous reactive system; Dynamic environments; Similarity metrics; Behavior patterns; Humanoid robots.

----

* Corresponding author.

1. Introduction

Nowadays, society is constantly interacting with dynamic environments, ranging from low to high dynamicity like homes, streets, and factories among others. Regardless of the level of dynamicity in the environments, it is common that multiple objects are moving at the same time changing the environment. These changes demand from a person to produce an appropriate response to the environment in a reasonable time, which can vary depending on the dynamicity of each environment. Although the human brain learns from previous environment situations, the probability of such situations to repeat exactly as they were is extremely rare. It would mean that every element in the environment would be in the same position, moving at the same rate and that the person reacting to the environment would be in the same place altogether with the rest of the people. Although these conditions are rarely met, the human brain uses previous knowledge to react “instinctively” to the events presented in everyday life. Recently, with the evolution of the robotics, researchers have tried to implement these skills into humanoid robots. Currently, robots cannot process the environment in the same way as a human being. Instead, they create patterns and decompose a complex task into a set of subroutines designed to solve a specific problem; e.g., Flores and his colleagues [1] proposed a Fuzzy Logic Controller to produce a walking pattern to achieve stabilization for bipedal locomotion. The Fuzzy Logic Controller provides the simulation of human reasoning through knowledge from the way that a person should walk. Additionally, Luo and his colleagues [2] proposed a pattern generator for walking, based on the model of Five Masses with Angular Impulse for humanoid robots. The proposed model was composed of the Conventional Model of Three Masses and the Law of the Conservation of Angular Impulse; which describes the relationship between two oscillating arms and legs. In addition, they designed a walk compensator pattern to check whether the previous pattern was efficient or not for a specific time lapse. By increasing or reducing the responsibility of the joints, the compensator shares and distributes redundant time to another joint that might need more time. These researches aim to produce a human-like behavior, enabling the robot to move like a human being. However, to interact with the real world, a robot needs to process the environment and react to it as a human would. This task requires of multiple subsystems like computer vision, analysis of its sensors, processing the spatial location of the robot in the environment, and the knowledge of the position of its joints in a specific action. It is important to know the position of the joints in a specific action to react adequately to sudden or dangerous changes in the environment; that is, if the robot is walking, with the right foot forward, it should not turn rapidly to the right. Thus, it should consider other options to avoid the danger. Therefore, we must know the current state of each joint to know the possible actions to perform. As a human, a robot can store multiple previous states of its joints and the possible immediate actions to avoid a fall or danger. However, the amount of possible combinations for the state of the joints is hardly small, considering the number of unions that a humanoid robot may have. Therefore, finding a stored state that resembles the current state of the joints can be a computationally expensive process. Within the Artificial Intelligence, it exists similarity metrics which are designed to measure the likeness between two entities. Thus, the current state of the joints of a robot can be processed as one of these entities. Usually, these models use vectors as input information and the data contained are usually binary, that is, within the domain 0 and 1 [3], but models are flexible enough that they can be implemented in different contexts with other data types [4]. The following works implemented similarity measures for their proposals. It is interesting that the scope of their application is quite different among them. In

[5] proposed a set of functions, called Deep Perceptual Similarity Metrics (DeePSiM). These functions calculate distances among the characteristics of the image extracted by a Deep Neural Network instead of computing distances in the space of the image, generating sharper and more natural images. On the other hand, in [6] tackle the Linking Prediction problem. They mention that the most common implementation for Link Prediction is to calculate the next links for each link and unconnected outputs, using Similarity Measures, resulting in high percentages of similarity in the prediction. Thus, the authors proposed a new method based on Temporal Similarity Metrics and Continuous Action set Learning Automata, which takes advantages of the use of different Similarity Metrics. In [7] show repetitive patterns as meaningful visual signals for the matching and detecting objects in images. There are Similarity Metrics that do not consider repetitive patterns and therefore cannot handle images well with repeated patterns. In response, they propose a new Feature Repeatability Similarity Measurement (FRS), which allows the use of repetitive patterns of information to improve recovery performance. The proposed FRS framework detects repetitive patterns using a descriptor and geometric information of local features in the images. Finally, [8] present results from multiple evaluations for the performance of different Similarity Metrics, adapted to work with text strings, focusing on the task of matching toponyms. In addition, they report the use of supervised machine learning to bypass the manual adjustment of the Similarity Threshold. As we can see, similarity metrics are commonly used for different purposes and, depending on the context, some similarity metrics might yield better results than others. Therefore, it is advised to make a study on the similarity metrics considered for each work. As stated before, the robots need to react to the environment fast enough to reach the reaction level of a human, adapting its behavior to sudden changes that require immediate actions to avoid physical damage or even injure third parties. Naveau and his colleagues [9] proposed a walking patterns generator that considers the position and orientation of the feet to avoid obstacles. Therefore, the paper showed an extension of the pattern generator that directly chooses to avoid convex obstacles. The algorithm uses the whole-body dynamics to correct the trajectory of the center of mass of the underlying simplified model. Moreover, robots must have the ability to adapt their walking process to rough or mon-flat terrain, involving tasks such as balance control and inclination of the body or feet. In [10], Yu and his colleagues proposed a gait pattern generator method for an omnidirectional biped walking and a model that described the motion of biped walking over sloped ground. The trajectories of each foot were designed considering the walking speed, step length, and walking direction. Finally, the motion trajectory of the center of mass of the robot was planned through a linear inverted pendulum model in the sagittal and coronal planes. On the other hand, Hong and his colleagues in [11] proposed a modifiable walking pattern generation algorithm, which allows humanoids to handle dynamic walking commands by changing its walking period, step length, and direction independently. If the humanoid is given a command to perform an infeasible movement, the algorithm substitutes the infeasible command with a feasible one using binary search. The feasible navigational command is subsequently translated into the desired center of the mass state. Based on this algorithm, they created various complex walking patterns such as backward a sideways walking. Based on the phase and trajectory analysis, Zhang and his colleagues in [12] proposed a modified gait planning method based on the central pattern generator (CPG) for the sampling-based footstep planning. By adjusting the parameters of the CPG, it is possible to obtain different gaits of going forward, stepping side and swerving, which allows for smooth transitions among these gaits. In this context, we will study the execution time, for search and recognition the current state of the joints, of different similarity metrics while not considering the value of the

similarity; with the objective of supporting an autonomous reactive system in a humanoid robot. The autonomous reactive system (ARS) previously mentioned, will be implemented on the humanoid robot platform Robotis OP2 (see Figure 1), to generate reactive behaviors within an environment, allowing the robot to perform navigation tasks. This platform has 20 degrees of freedom that give it the ability of movement and displacement. For limitations of this work, as mentioned before, we are not considering the returned value for each metric, we will just evaluate the responsive time generated by each metric in several specific workload scenarios. Due to the different implementations or applications of the measures, they generate results with different ranges of values, even if the input data is normalized, the results obtained will not be the same. In robotics, the field of locomotion is of great relevance to the development of humanoid robots. It has led to modeling and implementation of human-like displacement actions. For this purpose, researchers use patterns for the representation of desired positions and angles on a specific time lapse; e.g., when performing a specific action within the process of walking in a bipedal entity.Human-like biped displacement involves synchronization and participation of multiple joints for the correct reaction to a demanding environment. This supposes that an ARS determines the possible actions, so it is very important that it has awareness about the current state of each joint at a specific time.

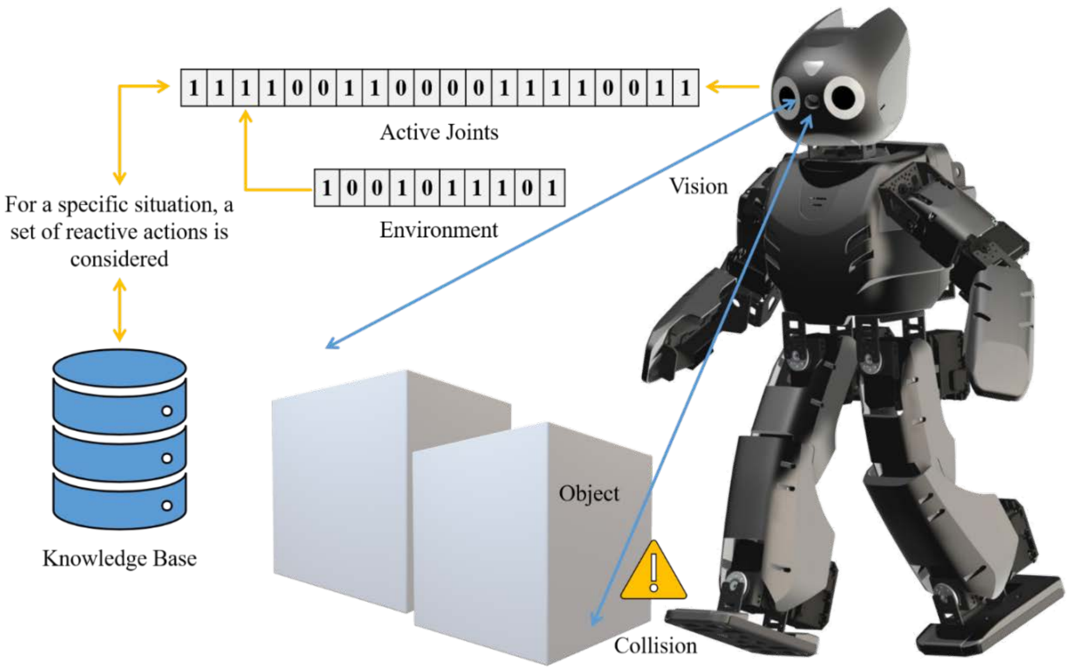

Figure 1: The robot must react in such a way that allows it to adapt to a constantly dynamic environment

In this context, the response time is crucial for decision-making, considering that the simple act of taking a step requires a set of states of the joints. Table 1 shows a single state during the walking process, considering only the activation of the joints. This state can be considered as one of the patterns stored in the robot, and a set of patterns represents the whole action of a step, one set of patterns for the left and other for the right leg. This ARS would involve taking the input pattern, considering the changes in the environment and if needed, reacting according to a set of possible actions that can be performed from the current pattern; by consulting the knowledge base and finding the current state or the most similar state to obtain the set of possible actions.

Table 1: Pattern sample from the walking action of the robot Robotis OP2

1	0	0	1	0	1	0	1	1	0	1	0	0	1	0	1	0	1	1	0

2. Similarity Metrics

Similarity techniques allow to find, and group objects that share features within a defined space. Based on their characteristics we can search, recognize, label and classify entities [13]. Although detecting whether a group of objects is similar or dissimilar may be a simple task for the human brain, it gets complicated when the number of variables or elements inside the entity reaches a certain limit. Bringing this ability to computers or robots might be more complicated because of the involvement of multiple areas of research such as computer vision, automatic learning or data analysis. A very important aspect that should be considered for the implementation of these techniques is to understand that the similarity or dissimilarity between two objects depends on the context. The equivalence between objects can be modeled through a similarity function, which produces a numerical value to indicate the degree of likeness or inequality between two entities [13]. In our context and given two binary vectors, we want to calculate the relationship between them. This can be achieved through functions that calculate a numerical value which specifies the level of similarity between them. For this paper, we considered the following similarity metrics.

A. **Manhattan Distance.** In [14] proposed this model, where the sum of the absolute differences is used to measure the distance between two points (see Equation 1). The Manhattan Distance will return the values of 0 when the vectors are identical and *n* if the vectors are completely different.

$$Manhattan(x, y) = \sum_{i=1}^{n} |x_i - y_i| \qquad (1)$$

B. **Euclidean Distance.** This concept is taken from the distance equation proposed by Pythagoras. Is defined as the straight-line distance between two points, which examines the root of square differences between the coordinates of a pair of objects [15] (see Equation 2). The result of the Euclidean distance generates values of 0 and *n*, where the value 0 would indicate that both cases (*x*,*y*) are identical while the value *n* indicates otherwise.

$$Euclidean(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \qquad (2)$$

C. **Chi-Square Distance.** This model calculates the distance between two points or samples $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$ similarly as the Euclidean distance but with the difference that it uses a vector of weights $W = (w_1, \dots, w_n)$ and does not apply the squared root (see Equation 3). The return values are of a wider range, ranging from 0, when samples are identical, to *n* indicating that there is no resemblance at all.

$$ChiSquared(x,y) = \sum_{i=1}^{n} w_i (y_i - x_i)^2 \qquad (3)$$

Sorensen-Dice Coefficient. This model measures the likeness between two sets, X and Y. $D(X,Y) \in [0,1]$, with $D(X,Y) = 0$ if, and only if the sets are disjoint and $D(X,Y) = 1$ if, and only if the sets are identical [16] (see Equation 4). Returned values range from 0 to 1, where 1 means that vectors are identical and 0 when vectors are completely different.

$$Dice(x,y) = \frac{2 * \sum_{i=1}^{n} x_i * y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2} \qquad (4)$$

D. **Jaccard Index.** For two sample sets, the model is defined as the proportion of intersection between samples X and Y divided by the proportion of their union [17] (see Equation 5). The values generated by the Jaccard index range from 0 to 1, a value of 1 is reached, indicates that total equality is found and 0 when vectors are completely different.

$$Jaccard(x,y) = \frac{\sum_{i=1}^{n} x_i * y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2 - \sum_{i=1}^{n} x_i * y_i} \qquad (5)$$

E. **Cosine similarity.** Let $X = (x1, x2, \dots, xn)$ and $Y = (y1, y2, \dots, yn)$ be two n-dimensional vectors with positive components. The cosine of two vectors X and Y is the inner product of X and Y divided by the products of their lengths [18] (see Equation 6). Cosine similarity provides values between 0 to $\sqrt{n}$, where $\sqrt{n}$ means that both objects are identical and 0 otherwise.

$$Cosine(x,y) = \frac{\sum_{i=1}^{n}(x_i * y_i)}{\sqrt{\sum_{i=1}^{n}(x_i^2) * \sum_{i=1}^{n}(y_i^2)}} \qquad (6)$$

The last three similarity metrics have a peculiar behavior that makes them little suitable for measuring binary data. We will address this issue in the Discussion Section.

3. Methodology

The similarity metrics used in this paper will be tested to identify the metric that has the best efficiency (computational cost). In order to use it inside a robot to perform tasks, like search and retrieve information, in the shortest time possible. For the purposes of measuring the similarity metrics we carried out the experimentation in a computer with an Intel Core i7-7700HQ @2.8GHz processor with 32GB of RAM @2400MHz running Windows 10 Pro 64 bits and the code of the similarity metrics were implemented in Python 3.6.2. For the experimentation, it was used a knowledge base that was generated randomly, which contains patterns of possible active joints combinations for a specific moment in the movement or position of the robot. A single movement, such as lifting the arm, involves a set or sequence of specific active joints. Thus, the patterns collection intends to represent several movements and their different patterns that represent the movement's sequence (Table 2). Then, an input pattern is compared with the whole knowledge base, in order to identify the current movement and its state. Once this information is known, then we can identify a set of possible immediate action that can be carried out to react to the changes in the environment properly.

Table 2: Stored Pattern samples for a certain action

0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	1	0	0	1	1
0	1	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1	0	0	1
1	0	1	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	1	1
1	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0
0	1	1	0	0	1	1	0	1	0	0	1	0	1	0	0	0	1	0	1
⋮																			
1	1	1	0	0	1	1	0	1	0	0	1	1	1	0	1	0	1	0	1

The patterns used for the experiments were generated randomly in the domain of 0 and 1, where 0 indicates that the joint is not in use at that moment and 1 otherwise. The vectors have a size of twenty, and each element represents a joint in the robot. In this study, we created nine different knowledge bases that range from 10,000 to 50,000 patterns, which are less than 0.05% of the number of possible combinations for a binary vector of twenty elements which is 2^20= 1,048,576. For each one of the similarity metrics and a number of patterns in the knowledge base, every *x* in Table 3, the process of applying the similarity metric from the input pattern to the whole knowledge base was repeated twenty-five times to calculate the average execution time for each similarity metric.

Table 3: Average time (x_n) for an input pattern matching against *n* samples in the knowledge base

Patterns	Dice	Cosine	Jaccard	Euclidean	Manhattan	Chi-Square
10000	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$x_{1,6}$
15000	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$
20000	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$
25000	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$x_{4,6}$
...	...	...	...	…	…	…
50000	$x_{9,1}$	$x_{9,2}$	$x_{9,3}$	$x_{9,4}$	$x_{9,5}$	$x_{9,6}$

4. Experimentation

For the experiment, an input pattern is given and matched against the *n* patterns, from 10,000 to 50,000, in the knowledge base using the six different similarity metrics. Table 4 shows the average time spent on each model for all cases. Also, we can observe that the values generated by the first three models are quite distant from those generated by the last three models.

Table 4: Averages time (*ms*) generated for each Similarity Metric

KB	Dice	Cosine	Jaccard	Euclidean	Manhattan	Chi-Square
10000	138	159	140	80	46	77
15000	206	212	212	115	66	118
20000	296	295	303	149	89	157
25000	358	355	357	190	111	190
30000	419	426	421	223	133	230
35000	492	503	490	266	155	271
40000	568	570	551	299	179	309
45000	629	645	621	333	210	356
50000	686	706	692	367	214	385

For better visualization we split the results into two groups, the first group contains the metrics Dice, Cosine and Jaccard. They all produce an average computational time above 130 milliseconds for the smallest test and above 680 milliseconds for the largest test (see Figure 2). We believe that their large computational time is related to the mathematical operations required to calculate their functions.

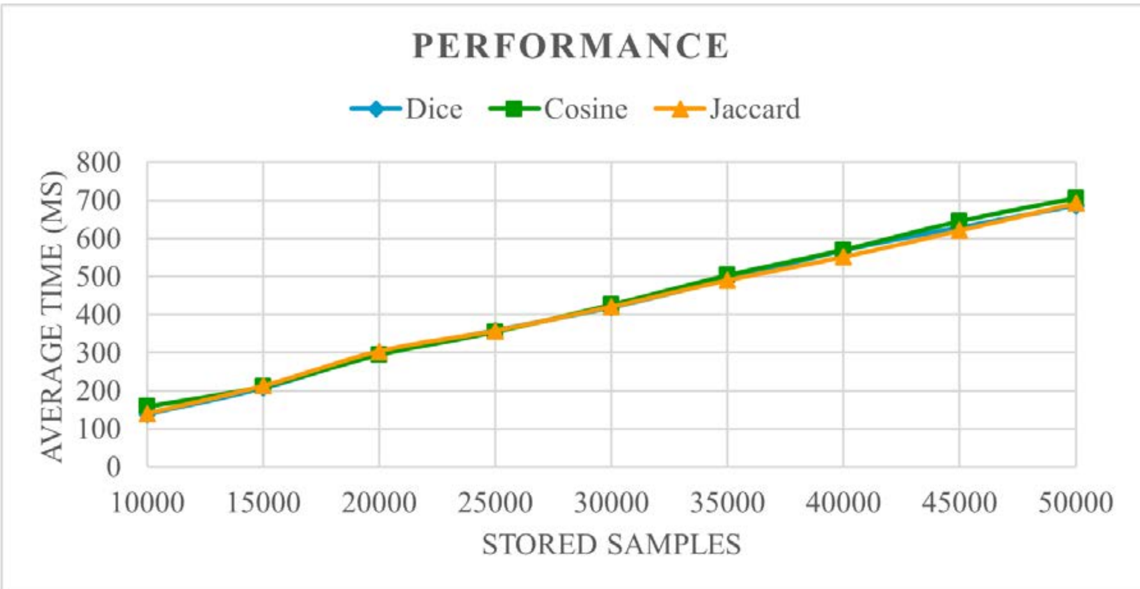

Figure 2: Performance for Dice, Cosine and Jaccard algorithms running under the same working conditions.

According to Figure 2, Cosine measure generates the highest runtime values above the other two algorithms in most cases, followed by Jaccard algorithm; while the metric Dice presents lower average values in several cases. In the following section, we show a statistical study for these metrics. On the other hand, the Euclidean, Manhattan and Chi-Square metrics have the best efficiency, producing a time between 45 and 80 milliseconds for the smallest test and between 210 and 390 for the largest test (see Figure 3). As we can see, the algorithms of the first group produce a larger computational cost, where the Cosine metric is the largest among them. On the other hand, for the second group, the Manhattan metric produces the lowest time consumption.

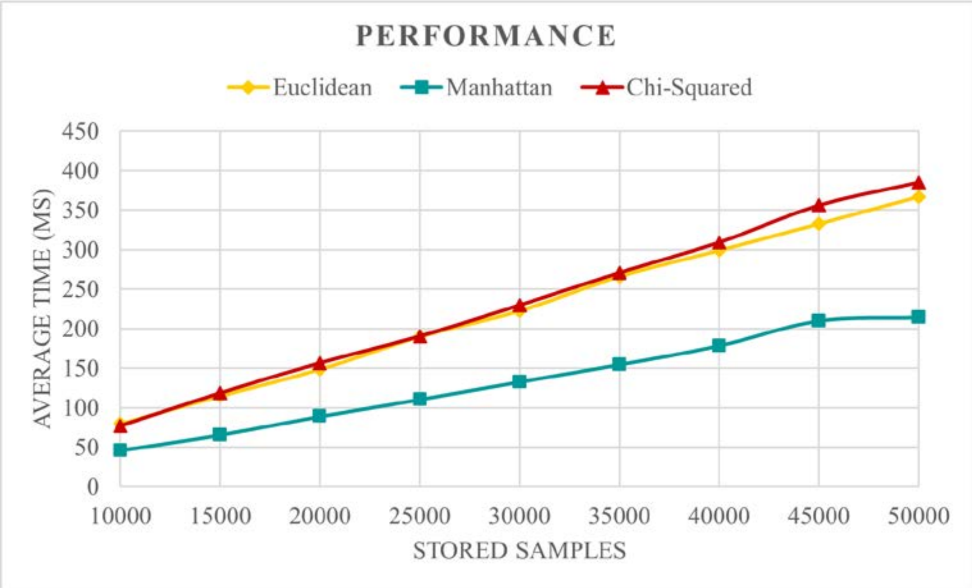

Figure 3: Performance for Euclidean, Manhattan and Chi-Squared algorithms running under the same working conditions.

5. Results

In Figure 4, we can see the performance of all the metrics tested. This figure shows the Cosine as the metric with the largest average time, while Manhattan Distance offers the shortest time compared to the other algorithms. Additionally, Euclidean Distance and Chi-Square can be considered as alternatives to the Manhattan Distance. To provide statistical support for the results and to assess the statistical difference between the average times collected, we used the Friedman test which is a nonparametric statistical test for multiple groups of measures [19]. Also, to determine which pairs of metrics are statistically different, we used the Wilcoxon Signed Rank test which is a nonparametric related-sample test [20]. For these tests, we considered as a null hypothesis H_0, which supports that there is no statistical difference in the performance among the efficiency of the models. While the hypothesis H_1, supports that there is a statistical difference in the efficiency of the model (Equation 9).

$$H_0: \mu_1 = \mu_2$$

$$\textit{There are No differences between average times} \tag{9}$$

$$H_1: \mu_1 < \mu_2 \vee \mu_1 > \mu_2$$

$$\textit{Average times present a significant difference}$$

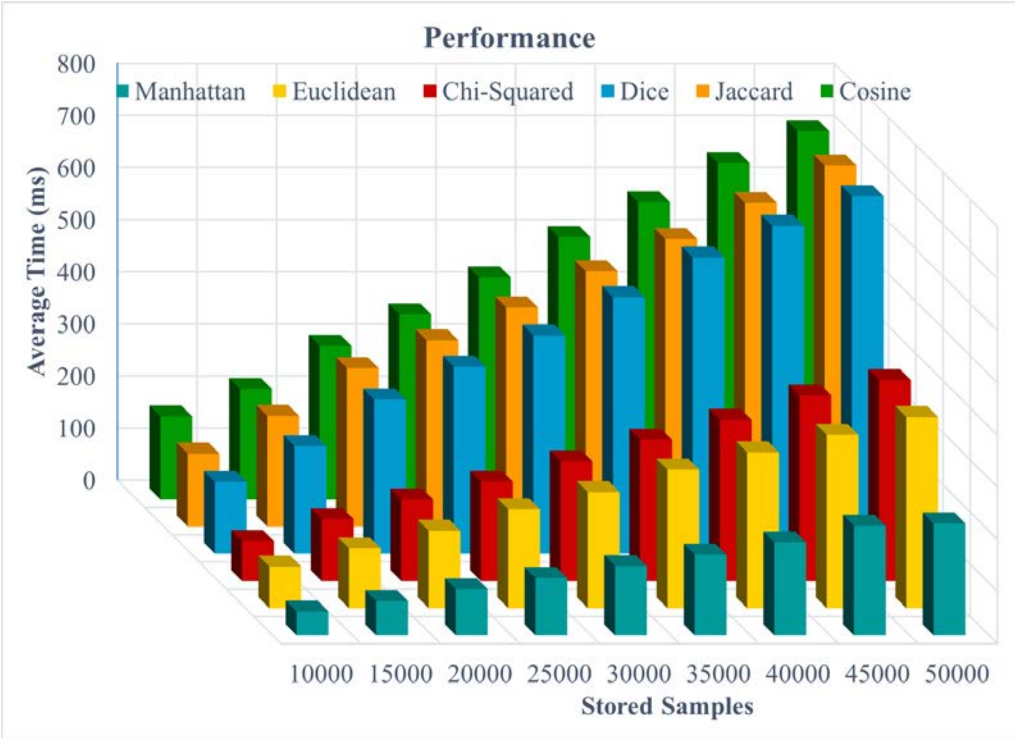

Figure 4: Model's performance in each test under the same working conditions.

As we stated before, we split the metrics into two groups, the first group with the metrics Dice, Cosine and Jaccard and the second group with the metrics Euclidean, Manhattan, and Chi-Square. We split these metrics according to a visual numeric difference; however, to identify if the metrics from the first group have a significative difference from the metrics of the second group, we need to know which metric from the first group has the best performance and which metric from the second group has the worst performance. Thus, to identify the metric that has the best performance from the first group, we used a Friedman test, which showed that there are no significant differences among the Dice, Jaccard, and Cosine, with a p-value of 0.165 which is larger than the alpha value of 0.05. This result accepts H_0 meaning that the three metrics from the first group does not have a statistical difference among them. However, the test also identifies the Dice metric as the one with the Ishigaki average time.On the other hand, we carried out the Friedman test for the second group which produces a p-value lower than 0.001 which is lower than the alpha value of 0.05 giving more than 95% of certainty of this statement. This result rejects the H_0 meaning that there is a statistical difference among the metrics of the second group. Additionally, Chi-Square was identified as the metric with the largest average time, whether if there is a significant difference when compared to the second worse metric, which is the Euclidean metric. Once we know the best metric from the first group (Dice) and the worst metric from the second group (Chi-Square), we carried out a Wilcoxon test which produces a p-value of 0.008 which is lower than the alpha value of 0.05. This specific value (0.008) rejects H_0 and produces a 99.2% of confidence that the Chi-Square metric has better efficiency than the Dice metric. As we can see, there is a significant difference between both groups. However, we wanted to know if there were a significant difference between the Chi-Square and Euclidean metrics, for this reason, we carried out a Wilcoxon test. The result showed that the Euclidean metric has statistically better efficiency, with a p-value of 0.021 which gives a certainty level of

97.9%. With these results, we conclude that the Chi-Square metric has the worst efficiency of the second group while performing better than any metric of the first group. Finally, a Wilcoxon test was carried out to measure the statistical performance of Manhattan when compared to the Euclidean metric. This test showed that the Manhattan metric have statistically better efficiency than the Euclidean metric with a p-value of 0.008 which produces a certainty level of 99.2% rejecting H_0.

6. Discussion

There is an interesting behavior for the Cosine, Jaccard, and Dice metrics, they all produce one when total equality is found; however, every element in the vector must be one. On the other hand, those metrics will produce zero when total inequality is found or when every element is zero even if both vectors are identical. According to this behavior, the output data can be normalized to adjust it within the other metrics output domain and then, the results of both groups can be compared but, this normalization process can undermine the performance of the metrics and generate the same results in terms of execution times. Additionally, further studies may include the angle of the joints; so, this new information may change the problem of measuring binary vectors to measuring vectors of real numbers. This change would require additional experimentation to identify the best metrics for the new structures.

7. Conclusions

In this paper, a comparison study was carried out to measure the computational cost of six similarity metrics, with the objective to determine which model offers the best alternative for the task of search and recovery information. This study is relevant in the context of robotics, particularly to support an autonomous reactive system. This system will allow the humanoid robot to move reactively according to a dynamic environment. This behavior will give the robot the ability to perform navigation tasks and even, interact with its environment in real time. Therefore, the decision-making process must decide with optimal responses in the shortest possible time. Based on the results obtained, we decided to separate the metrics into two groups. The first group, integrated by Cosine, Jaccard, and Dice metrics, which produced the worst average time for the tests. On the other hand, the second group integrated by Chi-Square, Euclidean and Manhattan metrics showed the best efficiency. Additionally, non-parametric tests (Friedman and Wilcoxon) showed that the Chi-Square metric, which was the worst metric from the second group, outperformed every metric from the first group. Besides, the Manhattan metric outperformed the other five metrics in this study.

8. Recommendations

Cosine, Jaccard, and Dice metrics do not seem useful for measuring binary vectors as in this case. Maybe working with another data type like real numbers, these metrics will result in more appropriate measuring tool according to its behavior and characteristics, this means that the application context plays an important role in the implementation of these metrics. According to its behavior, these metrics measure how similar an entity is to another, unlike Manhattan, Euclidean and Chi-Square metrics that determine how far an entity is from another within a specific coordinate space.

Finally, the Chi-Square metric uses weights, which in further studies may be useful to give priority to certain joints to measure the similarity when compared to other vectors while considering the importance of those specific joints.

Acknowledgments

ISC. Guillermo Toledano (CVU: **817596**), thanks to the Consejo Nacional de Ciencia y Tecnología (**CONACYT**) for the support and funding of his postgraduate studies in Master program in Computer Science. Likewise, thanks to the faculty of Engineering "Arturo Narro Siller" and the Division of Postgraduate Studies and Research of the Autonomous University of Tamaulipas, to facilitate the facilities and equipment necessary for conducting their research.

References

[1]. K. L. Flores Rodríguez and F. Trujillo Romero, "Desarrollo de un Sistema Lógico Difuso para el Control de la Locomoción Bípeda de un Robot Humanoide NAO," Research in Computing Science, no. 113, pp. 181-194, 2016.

[2]. R. C. Luo, K. C. Lee and A. Spallanzani, "Humanoid robot walking pattern generation based on five-mass with angular momentum model," IEEE 25th International Symposium on Industrial Electronics (ISIE), Vols. 2016-November, pp. 375-380, 2016.

[3]. J. Demey, J. L. Vicente Villardon, J. A. Di Rienzo and F. Casanoves, Valoración y Análisis de la Diversidad Funcional y su Relación con los Servicios Ecosistémicos, vol. V, CATIE, 2015, pp. 47-59.

[4]. J. Songlei, L. Cao, L. Kai, and H. Gao, "Unsupervised Coupled Metric Similarity," IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 9, pp. 1810-1823, 2018.

[5]. A. Dosovitskiy and T. Brox, "Generating Images with Perceptual Similarity Metrics based on Deep Networks," ArXiv Preprint ArXiv: 1602.02644, vol. 1, pp. 1-9, 2016.

[6]. B. Moradabadi and M. Reza Meybodi, "Link prediction based on temporal similarity metrics using continuous action set learning automata," Physica A: Statistical Mechanics and its Applications, vol. 460, pp. 361-373, 2016.

[7]. D. Manandhar and K. H. Yap, "Feature Repetitiveness Similarity Metrics in Visual Search," IEEE Signal Processing Letters, vol. 24, no. 9, p. 1368–1372, 2017.

[8]. R. Santos, B. Martins, and P. Murrieta-Flores, "Learning to combine multiple string similarity metrics for effective toponym matching," International Journal of Digital Earth, pp. 1-26, 2017.

[9]. M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A Reactive Walking Pattern Generator Based on Nonlinear Model Predictive Control," IEEE Robotics and Automation Letters, pp. 10-17, 2017.

[10]. Z. Yu, X. Chen, Q. Huang, W. Zhang, L. Meng, W. Zhang and J. Gao, "Gait Planning of Omnidirectional Walk on Inclined Ground for Biped Robots," IEEE Transactions On Systems, Man, And Cybernetics, p. 10, 2015.

[11]. Y.-D. Hong and B. Lee, "Experimental Study on Modifiable Walking Pattern Generation," Journal of Electrical Engineering & Technology, pp. 2368-2375, 2015.

[12]. X. Zhang, J. Xiong, S. Weng, H. Li, M. Chen, Y. Gan, Q. Zhao, and Z. Xia, "A Modified Gait Planning Method for Biped Robot Based On Central Pattern Generators," International Conference on Information and Automation, pp. 1551-1555, 2015.

[13]. E. López Iñesta, "Aprendizaje de similitudes entre pares de objetos mediante Clasificación Supervisada," Universidad de Valencia, Valencia, España, 2017.

[14]. E. Krause, "Taxicab Geometry: An Adventure in Non-Euclidean Geometry," Dover Publications Inc., New York, 1986.

[15]. N. Hazim, S. Mahmood, W. Esam, and M. Sahib, "Face Detection and Recognition Using Viola-Jones with PCA-LDA and Square Euclidean Distance," International Journal of Advanced Computer Science and Applications, vol. 7, no. 5, pp. 371-377, 2016.

[16]. A. Shawn and H. Ghassan, "Multi-Region Probabilistic Dice Similarity Coefficient using the Aitchison Distance and Bipartite Graph Matching," arXiv, p. 9, 2015.

[17]. D. Prokopenko, J. Hecker, E. Silverman, M. Pagano, M. Nothen, C. Dina, C. Lange, and H. Fier, "Utilizing the Jaccard index to reveal population stratification in sequencing data: a simulation study and an application to the 1000 Genomes Project," Bioinformatics, vol. 32, no. 9, pp. 1366-1372, 2016.

[18]. P. Biswas and S. Pramanik, "Cosine Similarity Measure based multi-attribute decision-making with trapezoidal fuzzy neutrosophic numbers," Neutrosophic Sets and Systems, pp. 47-57, 2015.

[19]. T. Ishigaki, T. Oda, Y. Liu, D. Elmazi, K. Matsuo and L. Barolli, "A Neural Network Based Intrusion Detection And User Identification System For TOR Networks: Performance Evaluation For Different Number Of Hidden Units Using Friedman Test," Journal of Mobile Multimedia, pp. 251-262, 2015.

[20]. J. McDonald, "Wilcoxon Signed Rank Test," in 2Handbook of Biological Statistics, John Wiley & Sons, Inc, 2015, pp. 186-189.