

Quantifying Features Using False Nearest Neighbors: An Unsupervised Approach

Jose Augusto Andrade Filho^{*†}, Andre C. P. L. F. Carvalho^{*}, Rodrigo F. Mello^{*}, Salem Alelyani[†] and Huan Liu[†]
^{*}ICMC/USP, Sao Carlos, SP, Brazil. {augustoa, andre, mello}@icmc.usp.br
[†]Arizona State University, Tempe, AZ, USA. {salelyan, huanliu}@asu.edu

Abstract—Real-world datasets commonly present high dimensional data, which means an increased amount of information. However, this does not always imply an improvement in learning technique performance. Furthermore, some features may be correlated or add unexpected noise, thereby reducing data clustering performance. This has motivated the development of feature selection methods to find the most relevant subset of features to describe data. In this work, we focus on the problem of unsupervised feature selection. The main goal is to define a method to identify the number of features to select after sorting them based on some criterion. This task is done by means of the False Nearest Neighbor technique, which is rooted in chaos theory. Results have shown that this technique gives a good approximate number of features to select. When compared to other techniques, in most of the analyzed cases, it maintains the quality of the generated partitions while selecting fewer features.

Keywords—Machine Learning, Clustering, Unsupervised Feature Selection, Chaos Theory.

I. INTRODUCTION

Real-world datasets commonly present high dimensional data that increases the amount of information [1]. However, that does not always mean an increase of performance on the learning technique. For instance, one may have a dataset containing the following features or variables: humidity/moisture, temperature, pressure, wind speed, etc; for a given region. Now, consider that a researcher wants to model this dataset (by clustering, classifying or predicting). In that scenario, some features can be correlated that may increase the time-complexity of the learning approach as well as reduce its performance. Therefore, a subset of features may be enough to model the dataset.

Unsupervised Feature selection incurs an additional challenge, as we do not know classes in advance; therefore, selection becomes difficult [2], [3], [1]. The main reason is that, without the class label, it is very difficult to identify which features are the most relevant [4], [5]. It is also hard to define how many of those are needed in order to reduce the dimensionality of the problem. With a single dataset, it is possible to find several clustering partitions according to the selected features [6].

We observed that chaos theory tools could be employed to help address the challenge of defining how many features to use when performing the selection. Such a conclusion is

based on Whitney's immersion theorems [7] extended by Takens [8].

In his extension, Takens observes that after mapping observations into the right number of dimensions, there is no need of increasing them, since the distance in between observations remains the same even after increasing the number of dimensions. In this high-dimensional space, information is, therefore, best represented. Using the same principle, given a ranked set of features, we propose a technique to identify the number of dimensions (features) that represent a dataset. In our work, observations are samples on a dataset.

This paper is organized as follows: chaos theory concepts considered in this work are presented in Section II; Section III details our proposed model; and Section IV presents the experiment design with results. Concluding remarks appears in Section V.

II. CHAOS THEORY

Chaos theory was developed to better understand dynamic systems (in areas such as meteorology, physics, chemistry, among countless others) with a *random* behavior when observed over time.

According to immersion theorem extension [8], a time series x_0, x_1, \dots, x_{n-1} can be reconstructed in a multidimensional space $x_n(m, \tau) = (x_n, x_{n+\tau}, \dots, x_{n+(m-1)\tau})$, or time-delay coordinate space, where m is the embedded dimension and τ represents the time delay (or separation dimension). This mapping or reconstruction technique allows transformation in dynamic system observations (or rule outputs) as a set of points in m -dimensional Euclidean space. This reconstruction allows the obtainment of dynamic systems rules, consequently simplifying the study of behaviors and their usages under different circumstances, such as the study of orbits, tendencies and prediction [9].

The embedded dimension defines the number of axes for the time-delay coordinate space. The separation dimension supports extraction of the periodicity of series' behavior. This dimension defines the time delay of historical observations to be modeled and analyzed in order to predict future events. It basically point out how far we should go to obtain causal relationships in the series. Based on such a dynamic, we find the time-delay which helps us unfold the most dissimilar behavior in the dynamic system. Dissimilarities

help us unfold different system states that are related by a set of equations.

The embedded and separation dimensions support the study of series; however, we need to find those dimensions for any series, including ones generated by experimental data.

Fraser and Swinney [10] studied and confirmed that the Auto-Mutual Information technique (AMI) presents good results when estimating the separation dimension.

For the embedded dimension, Kennel et al. [11] propose the False Nearest Neighbors method (FNN), which calculates the closest neighbors for every data point in the time-delay coordinate space, starting with an embedded dimension equal to 1. Afterwards, a new dimension is added and the distance among the closest neighbors is again calculated. When this distance increases, data points are considered to be false neighbors, which makes evident the need for more dimensions to unfold the series' behavior.

In the present work, we consider Takens' immersion theorem, which extends Whitney's, and then employ it to select features for clustering. In order to perform the selection, we rank the features according to a metric. In this work, we use the Mutual Information metric. Afterwards, we compute the fraction of False Nearest Neighbor for any dataset under study and evaluate the embedded dimension. This in turn indicates the number of features to be considered. Finally, we select the best m -ranked features to represent the dataset.

III. FEATURE QUANTIFICATION

Chaos theory provides tools that can be used to study the behavior of time series. In this sense, the False Nearest Neighbors (FNN) method determines the embedded dimension, that is, the number of dimensions or axes needed to unfold the full behavior of a series, making possible its study.

The embedded dimension identifies the number of dimensions that can unfold the behavior of a time series. This is done by means of the False Nearest Neighbor algorithm, which analyzes the relation between two points considering different numbers of dimensions.

Consider a dataset D , with N features and M samples. Suppose we have $M = \tau$ and each feature aligned one after the other. With this time-delay, applying the same principle described by Takens [8] to identify the embedded dimension (m), we would be comparing the relation among its features.

Thus, we can use the False Nearest Neighbor algorithm, which has the support from Kennel et al. [12] and Takens [8] to identify the number of features to select.

However, there are many different ways to rearrange the dataset in order to use the False Nearest Neighbor. To overcome this problem, as a first step, we need to rank the features according to their importance. The next two subsections describe the ranking method used (subsection III-A) and the use of the False Nearest Neighbor to identify the number of features (subsection III-B).

A. Feature Ranking

In this work, we use Mutual Information (MI) to evaluate the correlation of features. The lower the MI is, the more different the series are. In our approach, we employ MI to compute the degree of dissimilarity for every feature (by comparing it to all others).

The Mutual information is defined in Equation 1, where X and Y are the variables (features, in this case); $p(x, y)$ is the joint probability density function of X and Y ; and $p_1(x)$ and $p_2(y)$ are the marginal probability density functions of X and Y respectively.

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p_1(x)p_2(y)} \right) \quad (1)$$

After computing the Mutual Information of every feature f_i against f_j , $\forall i \neq j$, we rank them according to the mean value of Mutual Information for each feature.

It is important to note that any method can be used to rank the features, since the False Nearest Neighbor only relies on the generated ranking.

B. False Nearest Neighbor – FNN

With the features ranked by their relevance, it is possible to use the False Nearest Neighbor (FNN) algorithm to identify the number of features to select.

FNN produces the fraction of false neighbors which has a strong relationship with the distances in between samples reconstructed in m -dimensional spaces. The lower such fraction for a given m , the better the reconstruction in m dimensions [13]. From that, we are able to analyze how the dataset is reorganized and its information reflected according to the number of dimensions (or features). According to Kennel [12], [11], when the fraction of false neighbor reaches 0, no significant modification in the distance of the points can be identified. Thus, there is enough information to unfold and understand the behavior of what is being studied.

The fraction of false nearest neighbors (C^d) for a given dimension d is defined in equation 2, where τ is the number of features (or time-delay) and R_{tol} is the threshold that defines if two samples relative positions presented significant change. The distance variation $V_{i,j}^d$, defined in equation 3, is the variation in the distance between two points when using different numbers of dimensions. In this variation, $R_d^2(\cdot)$ and $R_{d+1}^2(\cdot)$ represent the euclidean distance between two points considering d and $d + 1$ dimensions.

$$C^d = \frac{\sum_{i=1}^{\tau-1} \sum_{j>i}^{\tau} \text{sign}[V_{i,j}^d - R_{tol}]}{\frac{\tau(\tau-1)}{2}} \quad (2)$$

$$V_{i,j}^d = \sqrt{\frac{R_{d+1}^2(x_i, x_j) - R_d^2(x_i, x_j)}{R_d^2(x_i, x_j)}} \quad (3)$$

To clarify, FNN firstly assumes the embedded dimension $m = 1$; therefore, it evaluates how far single points, defined in \mathbb{R}^1 , are to their neighbors by computing the distances in between $f_{i,1}$ and $f_{j,1} \forall i \neq j$. Thus, for the embedded dimension equal to $m = 2$, the FNN will compute distances in between pairs $(f_{i,1}, f_{i,2})$ and $(f_{j,1}, f_{j,2}) \forall i \neq j$. So, when the embedded dimension is τ' , the FNN computes the distances in between every pair $(f_{i,1}, \dots, f_{i,\tau'})$ and $(f_{j,1}, \dots, f_{j,\tau'}) \forall i \neq j$.

Simply comparing the mean distance among dimensions is disadvantageous in that the FNN that has the advantage of also taking into account how these points relate to each other. This relation among samples, which cannot be expressed by figuring the mean distance, is what makes FNN a powerful tool identifying the number of features to select.

The FNN technique applied to feature selection is hereafter defined as FQFNN: Feature Quantification by False Nearest Neighbor. FQFNN is able, based on a feature ranking, to identify the number of features $N' < N$ that can describe the dataset. It uses the False Nearest Neighbor algorithm to compute the relations among samples while considering different numbers of dimensions. FQFNN can be easily parallelized since most of its computation is independent.

The next section presents conducted experiments to show how FQFNN could perform in real world datasets and how these results compare with others in the literature.

IV. EXPERIMENTS

In order to evaluate the FQFNN, experiments using the datasets detailed in subsection IV-A were conducted, which is defined in subsection IV-B. In subsection IV-C we present the results.

A. Datasets

Datasets were selected from the UCI repository¹. These datasets and their main characteristics (number of classes, features and samples) are presented in Table I.

Table I
DATASETS USED IN THE EXPERIMENTS

Dataset	Classes	Features	Samples
Dermatology	6	34	358
Glass	7	9	214
Heart	2	13	270
Iris	3	4	150
Lung Cancer	3	56	27
Yeast	4	20	205

Although these datasets present a relatively small number of features when compared to image and microarray data, for example, their selection allows for a comparison in a reasonable amount of time. This constraint is addressed below.

¹<http://archive.ics.uci.edu/ml/>

B. Design

Studies were divided into two steps. The first one relates to the application of a technique under study to a dataset. A subset of features makes up the output for this phase, and m is the number of selected features (by MitraFS [14], FSSEM [6] and the proposed technique, FQFNN). As a baseline, the total number of features (τ) was also considered alongside the subsets generated by the techniques to serve as the input for the next step.

For the *Evaluation* step, we used K-means to evaluate the selected features. Here, the K-means algorithm has three input parameters: the number of classes K , the dataset (containing only the features previously selected), and the starting centroids. These centroids are constant for all the techniques under evaluation. A total of 30 centroid sets for each dataset was randomly generated, to make the comparison among techniques as fair as possible.

K-means produces a partition for each technique under evaluation, and we compare these with the true class label in order to evaluate the clustering quality. We use CR (Corrected Rand) [15] to compute how similar the resulting partition and the true class are. This is based on the CR result with which we compared the techniques. This metric has a value between 0 and 1; with 0 representing there is no agreement in any pair of points, and 1 indicating that the two partitions being compared are exactly the same.

The evaluation step is computationally costly since it needs to be repeated several times for each dataset. Due to this, smaller datasets were used to compare the techniques and to evaluate the performance of the proposed method. However, the evaluation does not require as much time to produce the selected features even for large datasets. Therefore, this analysis will be done on larger datasets in future work.

C. Results

The result of the experiment is shown on Table II. From that, it is possible to see that FQFNN usually selects fewer features than the other techniques. In *Iris* and *Glass* datasets, FQFNN selects the same number of features; in the *Swissbank* dataset it selects more features. It is also important to note that, by selecting fewer features, we were also able to increase the quality of the final result. Besides, when compared to the other techniques, the quality does not present much difference, even though less features were selected.

As an illustration, we have computed the average number of features which were selected and the average value of CR. It shows that FQFNN selects fewer features than the other techniques studied, while the average quality remains about the same. Furthermore, when considering all of the features, all techniques analyzed were able improve the average quality.

Table II
RESULTS - RANDOM CENTROIDS

Dataset	FQFNN		MitraFS[14]		FSSEM[6]		Full Set	
	F	CR	F	CR	F	CR	F	CR
Dermatology	11	0.66	18	0.70	14	0.68	34	0.65
Glass	4	0.23	4	0.21	5	0.29	9	0.45
Heart	4	0.16	8	0.09	5	0.14	13	0.03
Iris	2	0.88	2	0.88	2	0.88	4	0.68
Lung Cancer	10	0.26	21	0.31	12	0.28	56	0.20
Yeast	5	0.70	9	0.72	6	0.71	20	0.74
Average	6	0.48	11	0.48	8	0.48	23	0.46

FQFNN did not present the best result every time, which means that some features may have been ranked in error. For instance, in the *Glass* dataset, FQFNN and MitraFS had selected the same number of features but their results were not the same. This means that both techniques had selected different features. Alternatively, in the *Iris* dataset, all the methods had selected the same features; therefore, their results were the same (this is only possible due to how the centroids were chosen). Thus, in order to increase the final quality, we needed to address how the ranking would be generated. Improving the ranking means improving the selection of features for the FQFNN.

V. CONCLUSION AND FUTURE WORK

The main contribution of this work is document how the False Nearest Neighbor (FNN) can be applied to feature selection. By applying the concepts from Takens[8] and Kennel[11] to the feature selection problem, we came up with a technique that could be used to address the challenge of identifying the optimal number of features to be selected. This technique was named FQFNN: Feature Quantification by False Nearest Neighbors. FQFNN has only one parameter, which identifies if the change in distance is significant. In this work we use the same value that is defined in [12]. The experiments performed show that this technique can be applied to feature selection.

For future work, we plan to study how FQFNN behaves when dealing with high dimensional data. Early studies show that its performance is satisfactory to our expectations. We also plan to study how different ranking methods, such as SPEC [5] impact on FQFNN result (number of feature to be selected) and its effect on the quality of the generated partitions.

ACKNOWLEDGMENT

This work is, in part, supported by NSF Grant (0812551), CAPES Foundation (process 6831/10-9), and CNPq (process 142207/2008-0).

REFERENCES

- [1] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering - a filter solution," *Data Mining, 2002. ICDM 2002. Proceedings. 2002 IEEE International Conference on*, pp. 115–122, 2002.
- [2] M. Devaney and A. Ram, "Efficient feature selection in conceptual clustering," in *International Conference on Machine Learning*, 1997, pp. 92–97. [Online]. Available: citeseer.ist.psu.edu/devaney97efficient.html
- [3] J. G. Dy and C. E. Brodley, "Visualization and interactive feature selection for unsupervised data," in *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2000, pp. 360–364.
- [4] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.
- [5] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pp. 1151–1158, 2007.
- [6] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, pp. 845–889, 2004.
- [7] H. Whitney, "Differentiable manifolds," *The Annals of Mathematics*, vol. 37, no. 3, pp. 645–680, July 1936.
- [8] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*. Springer, 1980, pp. 366–381.
- [9] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*. Springer, 1997.
- [10] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A*, vol. 33, no. 2, pp. 1134–1140, Feb 1986.
- [11] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, vol. 45, no. 6, pp. 3403–3411, Mar 1992.
- [12] M. Kennel, R. Brown, and H. Abarbanel, "Determining embedding dimension for phase space reconstruction using the method of false nearest neighbors," Institute for Nonlinear Science and Department of Physics, University of California, San Diego, Mail Code R-002, La Jolla, CA, EUA, Tech. Rep., 1992.
- [13] R. F. de Mello, "Improving the performance and accuracy of time series modeling based on autonomic computing systems," *J. Ambient Intelligence and Humanized Computing*, vol. 2, no. 1, pp. 11–33, 2011.
- [14] P. Mitra, C. Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, 2002.
- [15] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, pp. 193–218, 1985.