Anomaly detection using unsupervised learning

By Mateo Guaman Castro, ECE '20

Abstract

This technical note presents an overview of unsupervised machine learning, focused on its applications to anomaly detection. This article also presents an application of an unsupervised approach to anomaly detection of gait patterns.

Introduction

In our current age, the availability of data from the massive and widespread use of smartphones and other devices worldwide has allowed us to develop inference methods that utilize large quantities of data to solve a variety of problems. Algorithms that use these data to learn to perform tasks, such as classification or linear regression, have been proposed for close to 70 years [1,2], yet the advancements in computational power and the large availability of data have catapulted the development of data-driven, rather than traditional rule-based, methods, under the better known term "machine learning."

Machine learning has been at the forefront of many of the latest technological breakthroughs and has already become a part of our daily life. Although heavily influenced by statistical inference and optimization, machine learning has become a field of its own in the last decades. Under this umbrella term, three of the most significant areas are supervised learning, unsupervised learning, and reinforcement learning. In general, machine learning seeks to use data samples, rather than hard-coded algorithms, to achieve a task.

Unsupervised Learning

Introduction

In unsupervised learning, the goal is to determine the underlying structure of a given set of unlabeled data. One example of a type of unsupervised learning method is clustering, where data points from the training data are used to find clusters that represent the different data categories represented in the training data. Another example of unsupervised learning is dimensionality reduction, to find smaller representations of the input training data that contain as much information as the input data, similar to compression.

Supervised vs Unsupervised Learning

It is important to understand the differences between supervised and unsupervised learning to determine which method to use to fit the specific needs of a problem. The two main differences in these two types of methods are availability of data and interpretability of results.

Supervised learning requires input data that consists of two parts, the data (usually called X), and the label for the data (usually called Y). For instance, in image classification, X could be the matrix of pixel values for the picture of a cat, and Y could be the index of the class that represents a cat, say, 2. With a large enough amount of data, a learning algorithm will learn the characteristics that an image of a cat has, so that it will be able to recognize a cat in other images that it has not been trained on. Yet, this requires large datasets that must be manually labeled by experts, usually humans, which is both costly and time consuming. This limits the applicability of supervised learning to problems where labeled data already exists or where labeled data is easy to obtain.

Unsupervised learning, like its name suggests, does not require labeled data. The goal of these tasks is to

determine the underlying structure of the input data. This has certain benefits over supervised learning. The most noticeable advantage is that the lack of need for labels means that these methods can be used for a broader range of problems. Additionally, since these unsupervised algorithms themselves are tasked with finding patterns in the input data, they can find interesting and previously unknown features in the input data that may be useful for categorizing new data.

Yet, the same reasons that make unsupervised learning appealing over supervised learning also cause some of the drawbacks, the main one of which is interpretability of results. Given that the algorithm is itself recognizing data patterns rather than a human providing them, the results obtained from an unsupervised method may be harder to interpret and less trustworthy. Additionally, the methods used may be more computationally expensive, given that the algorithms are solving more complex problems.

Unsupervised Learning Methods

Clustering

In clustering, the main goal is to organize data points into different groups that share similar characteristics [3]. Clustering is a general task, and there are many different algorithms that tackle this problem, ranging from statistical to graphical approaches. Given that clustering tries to find structure in unlabeled data, it can be thought of as an unsupervised method. In this report, we will briefly describe three main types of clustering: connectivity-based clustering, centroid-based clustering, and distribution-based clustering. Connectivity-based clustering, also called hierarchical clustering, is based on the idea of aggregating data points based on their distance to other data points. Centroid-based clustering is based on the idea of finding a vector that represents a cluster and optimize the location of the vector so that the distance from the data points to the vector is minimal. Finally, distribution-based clustering is based on the idea of using probability distribution models to aggregate data points into clusters of data that might belong to the same probability distributions. One particular application of clustering is anomaly detection. After finding structure in some data space, and finding the clusters that represent this data space, we can characterize an anomaly as a data point that is far away from all of the clusters [4].

Dimensionality Reduction

Another important problem in unsupervised learning is dimensionality reduction. However, this problem is not limited to unsupervised learning. For instance, in information theory, this problem is studied under the umbrella of source coding. Dimensionality reduction is the problem of reducing the number of random variables that are needed to represent the input data as closely as possible. A more familiar term for this problem is compression, where we want to characterize data as efficiently as possible. In this tech note we discuss two dimensionality reduction techniques: Principal Component Analysis (PCA), and autoencoders. Principal Component Analysis is a linear dimensionality reduction method [5]. PCA works by linearly mapping high-dimensional inputs to lower-dimensional representations, where the lower-dimensional representation consists of the principal components, or most representative dimensions, of the data. Autoencoders are non-linear dimensionality reduction methods that make use of neural networks [6]. The goal of an autoencoder is to learn the parameters of a neural network (weights and biases) that achieve an encoding and decoding scheme with as little reconstruction loss as possible. The network learns a compressed representation of the input (encoder part), which it later uses to reconstruct the input (decoder part), as shown in Figure 1.

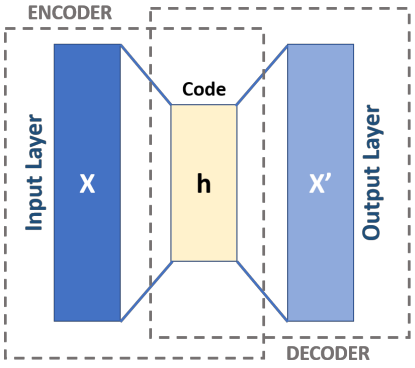

Figure 1. "Schema of a basic Autoencoder" by Michela Massi - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=80177333

Autoencoders used for Anomaly Detection

We are using autoencoders to learn a compressed representation for normal gait data, where the data comes from twelve pressure sensors placed on

different locations under the user's foot. The objective is to learn a compressed representation for these normal walking patterns that minimizes the error between the original input data and the output data reconstructed from the compressed representation. After learning what a normal walking pattern looks like, we then analyze whether a series of new data points come from anomalous walking behaviors, such as shuffling feet or using a ball-strike rather than a heel-strike gait, by looking at the reconstruction error at the output of the autoencoder. Since the reconstruction error will be lower for normal walking behaviors than for anomalous walking behaviors, we can determine a decision threshold to determine whether new input data correspond to anomalous walking behaviors or not. By making use of neural network architectures which keep track of temporal relations (recurrent neural networks, specifically, long-short term memory units), we can use this anomaly detection setup to predict anomalous behaviors with some anticipation. For testing purposes, we have collected normal walking data and anomalous walking data (shuffling, ball-strike gait) based on expert medical knowledge on walking patterns that precede a fall in the elderly. So far, our results show that we can use this anomaly detection method to detect anomalies in clean, short-term data, as shown in Figure 2. Our next steps are to collect longer term data and predict anomalies over longer periods of time.

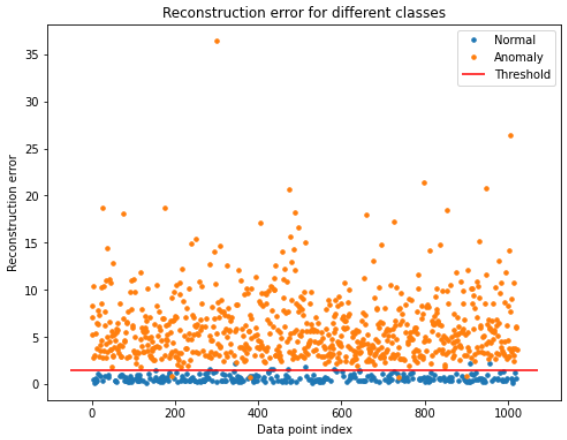

Figure 2. Reconstruction error for normal and anomalous data sequences, trained with inputs of 20 time steps. The threshold achieves good detection.

Conclusion

In this tech note, unsupervised learning methods are discussed, with an emphasis on unsupervised dimensionality reduction methods. Given the large availability of data nowadays, these methods can be applied to problems that can have a real impact on the daily lives of people. One such application is our implementation of an anomaly detection system using autoencoders to predict falls using footstep pressure data. One significant target for this application is the elderly, who are most at risk of suffering severe consequences after a fall, and who could benefit from knowing that their walking behavior is anomalous and may lead to a fall before the damage is done.

References

1. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

2. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

3. Rokach, L., & Maimon, O. (2005). Clustering methods. In *Data mining and knowledge discovery handbook* (pp. 321-352). Springer, Boston, MA.

4. He, Z., Xu, X., & Deng, S. (2003). Discovering cluster-based local outliers. *Pattern Recognition Letters*, *24*(9-10), 1641-1650.

5. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.

6. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.