Open World Classification With Gaussian Mixture Models

Md Abdul Aowal
maowal@umass.edu

Justin Clarke
jclarke@cs.umass.edu

1 Introduction

In traditional classification problems the data generating process has a fixed distribution. Samples are drawn from this distribution for both training and testing, resulting in two datasets with identical support. This approach is suitable for static environments where the possible inputs to the classifier are all fully known in advance. However, in many environments (and real-world applications) the ontology of possible classification targets is in flux. There may still be persistent patterns in the data generating process that are always relevant, but there also may be fluctuations that require an immediate change in the agent's behavior. Classification in an environment with a fluctuating data generating process is often known as Open World Classification. There are three critical steps in Open World Classification beyond those necessary in a typical classification problem:
1) Novelty detection - Identifying the appearance of objects from a previously unknown class.
2) Characterization - Learning the critical features of a novel class and how it differs from the existing classes.
3) Adaptation - Adjusting agent behavior to minimize additional risk posed by the appearance of a novel class, or to maximally exploit any benefit provided by a novel class.

In this case, our goal is to develop a classifier that can accurately classify objects known at training time, but can also accommodate new classes as they are discovered. We plan to use a Gaussian Mixture Model (GMM) where the join probability of membership in any class can serve as an inverse proxy for the probability of novelty. Once a sufficient number of novel objects are detected, we perform clustering to identify the parameters of any potential new mixture components (another step that is not necessary in typical classification problems). We then use AIC as our model selection criterion and determine which components, if any, to incorporate into our model.

2 Related work

The traditional classification methods have achieved significant success in several machine learning tasks under the closed-world assumption, that is, the training and testing data are drawn from the same label and feature spaces. However, the real world is open and dynamic, and in many applications such as self-driving cars, medical diagnosis, etc. where unseen situations can emerge unexpectedly and drastically weaken the robustness of these existing methods. Therefore, the model cannot expect it sees everything in training, which makes open-world learning or classification an interesting problem.

Open-Set Classification (OSC), proposed by Scheirer et al. (2012, 2014), has formalized processes for performing recognition in settings that require rejecting unknown objects during testing. They developed open set classifiers in a one-vs-rest setting to balance the performance and the risk of labeling a sample far from the known training examples (termed as open space risk). Later, Gunther et al. (2017) addresses this problem by thresholding confidence scores and assigning a single "unknown" label to all samples which do not meet the defined threshold. However, a limitation of this method is that it does not learn or otherwise take advantage of newly available data. While one can always train with an "other" class for uninteresting classes (known unknowns), it is impossible to train with all possible examples of unknown objects.

On the other hand, Open-World Classification (OWC), proposed by Bendale and Boult (2015), extends the concept of open-set recognition using Class Incremental Learning (CIL). Instead of assigning all unknown subjects to a single class, OWC distinguishes data from unknown identities and learns a new class for each unfamiliar subject. They presented the Nearest Non-Outlier (NNO) algorithm, an extension of the traditional Nearest Class Mean (NCM) approach that tackles open space risk while balancing accuracy. However, De Rosa et al. (2016) argued that several metric learning algorithms, like NNO and NCM, learn the model parameters on an initial closed set and do not change them as the problem evolves, contradicting the very own definition of OWC. Hence, they extended three algorithms, the Nearest Class Mean, the Nearest Non-Outlier, and the Nearest Ball Classifier, to update their metric and novelty threshold online. This study investigates a fast and straightforward approach of identifying new classes in open-world settings using Gaussian Mixture Models, with the aim for high performance and time efficiency.

2.1 Background

We will investigate whether GMMs can be adapted for open-world classification. Specifically, if they can accommodate new components who's parameters are learned at test time from limited samples. This will require an environment where we can explicitly control the data generating process to introduce novelty, or a naturally occurring process that has inherent variation in the distribution of values of the generated variables. The system we will study is a Gaussian mixture model and the combination of clustering, model selection, novelty detection, and classification. The task we will observe the system perform is classification, particularly in an environment with unknown classes present at test time.

2.2 Hypotheses or a research problem

Classification in open world environments requires addressing several interesting research problems. An immediate concern is effective novelty detection. If we use a minimum threshold on the unnormalized probabilities of class membership as an indicator of novelty, we face the problem that an unlikely sample from the non-novel distribution is prone to being misidentified as novel. We can raise the threshold very high to reduce the risk of false positive novelty identifications, but we are then likely to incur false negatives where we fail to identify novel samples that are relatively close to the known classes. There is a balance between sensitivity to novelty and the uncertainty in the novelty detections. Another research problem we will need to confront is the question of how many samples from a novel class are necessary for effective clustering. It is tempting to find the parameters of the new component as soon as possible and incorporate it into the model. However, collecting more samples of the novel class before clustering will lead to more accurate parameters. This also raises several related issues, such as how to recover from poor initial clustering when additional samples become available.

3 Data

We will require datasets that contain enough objects to learn the pre-novelty distribution of the data generating process, but also contain a post-novelty distribution where there has been a change to the data generating process. Ideally, we would be able to introduce novelty in a controlled manner where we could choose the timing and difficulty.

We plan to use data from ScienceBirds, an AI platform for Angry Birds. We are able to get ground truth data of the game state at various time steps that can be used for classification of the objects in the frame. There is a natural progression to the game where novel objects are introduced at each level, and we also plan to create our own novel objects by altering their features in different ways.

We can also use many popular datasets and make our own adjustments and perturbations to introduce novelty of varying difficulty. Similarly, by using a dataset with many known classes we can withhold samples of some classes during training, and then use them as novel classes at test time.

Finally, we plan to generate synthetic datasets with known distributions that we can make measurable and predictable changes to in order to simulate the presence of novelty.

4 Techniques and methods

We will rely heavily on many of the traditional methods and techniques in the classification literature, since the task we are undertaking is fundamentally similar. We can use existing tools and methods for clustering of novel samples, and for implementation of the GMM. The main changes will be the addition of novelty detection functionality, as well as coordinating the clustering process and updating the model for further classification.

We will also rely on many traditional performance metrics, such as accuracy, F1, etc., but we can apply them separately to novel, non-novel, and combined data. Additionally, we will analyze and compare the performance on the known classes before and after the introduction of novelty.

It will likely prove useful to define a basic method for characterizing the difficulty of a given novelty so we can make meaningful comparisons between environments and across different types of novelties. Even a simple ordinal scale with definitions may be sufficient.

5 Your planned analysis

We plan to experiment with different strategies for implementing the steps of the open world classification process. For each novelty detection method, clustering approach, model selection method, or model update, we will generate a set of standardized metrics that can be compared. Generally, they will be various metrics for accuracy.

In addition to varying difficulties of novelty, there are other aspects of the problem that we can vary that might help identify the best approach. For example, we will experiment with datasets with different numbers of classes, as well as different balances between classes.

6 Main challenges, schedule, and responsibilities

The main initial challenges relate to setting up the infrastructure to perform each step of the process and securing the necessary data. By dividing these early tasks we will be able to start running experiments sooner, and move collaborate on the analysis of the results. Below is our planned schedule:

1. Identify existing datasets and strategize about the generation of synthetic data (1 week)

2. Set up pipeline with existing tools and plan code updates. Begin data generation (2 weeks)

3. Implement code updates, complete data generation, and implement model testing framework (2 weeks)

4. Analyze results, improve approach and implement updates. Identify/generate and analyze any additional necessary data (2 weeks)

5. Produce final reports (1 weeks)

References

Bendale, A. and Boult, T. (2015). Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902.

De Rosa, R., Mensink, T., and Caputo, B. (2016). Online open world recognition. *arXiv preprint arXiv:1604.02275*.

Gunther, M., Cruz, S., Rudd, E. M., and Boult, T. E. (2017). Toward open-set face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 71–80.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., and Boult, T. E. (2012). Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772.

Scheirer, W. J., Jain, L. P., and Boult, T. E. (2014). Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324.