Survey: Open-set Classification

Alan Carvalho Neves*, Caio Cesar Viana da Silva†, Jéssica Soares‡ and Sérgio José de Sousa§
Department of Computer Science
Universidade Federal de Minas Gerais, Brazil
Email: *alan.neves@dcc.ufmg.br, †caiosilva@ufmg.br, ‡jessicasoares@dcc.ufmg.br, §sergiosjs@ufmg.br

I. Abstract

Classification is a important field of study in Machine Learning area. Traditional algorithms relay on closed-set scenarios, where the label classes are known in prior. Thus, in real world we eventually deal with unknown instances of data. To treat unseen instances in a proper way, a approach called Open-set classification was developed. This paper briefly surveys some of the recent works in Open-set Classification.

II. Introduction

In recent years, we experimented large advances in tasks that involves machine learning algorithms, mainly classification. In computer vision, impressive results were achieve in recognition and object detection. However, the traditional algorithms relay on closed-set scenarios, were the classifier was trained on a specified set of known classes. In real world there are dozens of possible classes and hierarchies - being impossible to recognize them properly in a closed-set system.

In Open-set classification, it should be possible to detect novel classes on data, performing incremental learning. This theme is challenging due it requires correct probability estimation of all known classes and prediction of unknown.

In this survey we will summarize five papers, published between 2015 and 2017, that use different methods to approach the open-set scenario.

III. Methods

A. Towards Open World Recognition [1]

Recently, databases for visual recognition systems have increased in size, using within a big variety. In this context, one of the challenges faced is to perform recognition tasks from a controlled environment to the real world.

In order to achieve a good performance in the "open world", a recognition system should update new object categories and be robust to these unseen groups, in addition, to have a minimum downtime.

Incremental learning algorithms handle new instances of known classes while open world requires a different and advanced approach. The first step is to continuously detect novel classes; The second is to update the system to include these new classes when novel inputs are found.

Distance-based classifiers like Nearest Class Mean (NCM) learn new classes incrementally incorporating new images or classes by adjusting existing means of a given class. However, this method uses close-set assumptions for probability normalization, since this method presumes all classes are known.

In their paper, [1] show how to extend Nearest Class Mean type Algorithms (NCM) to a Nearest Non-Outlier (NNO) algorithm that evolves model efficiently adding object categories incrementally while detecting outliers and managing open space risk. They also present a protocol for evaluation of the open world recognition in order to show the better performance of NNO algorithm in open world applications using ImageNet. The protocol is divided in two phases described bellow.

- *Training phase:* The training of NCM is divided in an initial metric learning/training phase and a growth/incremental learning phase. Both NCM and NNO add new categories, in the same way, hence [1] do not measure timing. In fact, NNO requires the estimation of which has to balance the classification errors between the known set of categories along with errors between the known and unknown set of categories. Therefore, [1] requirements are about high recall rate and optimization over F1-measure rather than accuracy.
- *Testing phase:* First, the data is divided into two sets: known and unknown sets. Afterwards, the open world evaluation test operates varying two variables: number of known categories in training (incremental learning) and number of unknown categories during testing (open set learning).

B. Extreme Value Machine [2]

An *Open world recognition* system must be able to detect unseen instances and incrementaly learn them [1]. The Nearest Non-Outlier (NNO) algorithm evolves the model efficiently adding object categories incrementally while detecting outliers and managing open space risk.

Unfortunately, NNO lacks strong theoretical grounding, using thresholded values for decision function and ignoring distribution information. To deal with that, a model called Extreme Value Machine (EVM) is presented [2]. The model is derived from *extreme value theory*, which calculates the radial probability of inclusion of a point in a class. A compact probabilist representation using extreme vectors (EV) is achieved using points and distributions that best summarize each class.

In EVM, a training set is represented by a set of extreme vectors, each of them associated with Probability of Sample Inclusion (Ψ) derived from EVT. The Ψ term is modeled in terms of distribution of sample half-distances relative to a reference point. For each positive reference point, we get half distances to the nearest negative samples - as in figure 1.

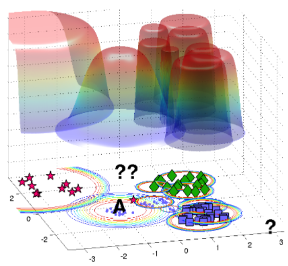

Fig. 1: EVM algorithm trained on four classes. The colors in the rings show a probability for each extreme vector chosen by the algorithm. EVM supports open set recognition and can reject the three "?" inputs that lie beyond the support of the training set as "unknown".

As EVT theorem states, the minimal values of margin for a given point is given by Weibull distribution. This way, the probability that a sample x' is included in the boundary estimated by x_i is defined as:

$$\Psi(x_i, x', k_i, \lambda_i) = exp^{-\left(\frac{\| x_i - x' \|}{\lambda_i}\right)^{k_i}} \quad (1)$$

where $\| x_i - x' \|$ is the distance of x' from sample x_i, x', k_i, λ_i are parameters of Weibull distribution.

Given a point x' in space, the probability that x' belongs to a class is defined as the max probability $\hat{P}(C_l|x')$ between the known classes compared to a threshold(δ). If $\hat{P}(C_l|x') \geq \delta$ then x' belongs to C_l class, otherwise the point is classified as "unknown".

As new data arrives, the model is incrementally updated by recalculation of the extreme vectors, using a greedy approximation - balancing desired accuracy and low computational resources.

The EVM was tested with Multi-Class Open Set Recognition on OLETTER dataset, achieving results comparable to W-SVM with lower training cost. On Open World Recognition test on ImageNet dataset, EVM is compared with state-of-art NNO algorithm, outperforming NNO in terms of accuracy and F1-measure, as shown in figure 2.

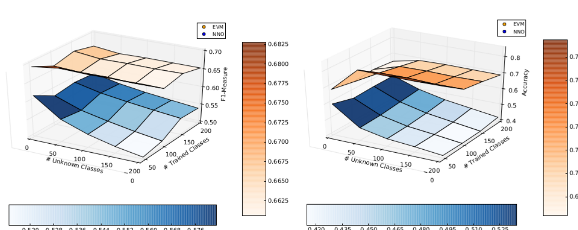

Fig. 2: Open world performance of the EVM and NNO on the ImageNet in terms of F1-mearsure (left) and accuracy (right).

C. Towards Open Set Deep Networks [3]

It is expected that images from unknown classes will all have low probabilities as the classification result. This is true only for a small fraction of unknown inputs. Thresholding uncertain inputs help but is still a relatively weak tool for open set recognition.

By dropping the restriction for the probability for known classes to sum to 1, and rejecting inputs far from known inputs, a new layer, called OpenMax (alternative to SoftMax) can formally handle unknown/unseen classes during operation.

This new layer uses the scores from the penultimate layer of deep networks (the fully connected layer before SoftMax) to estimate if the input is far from known training data. The approach is based on the fact that the values from this layer (Activation Vector (AV)), are not an independent per-class score estimate, but rather they provide a distribution of what classes are related.

It is estimated that for most categories, there is a relatively consistent pattern of related activations, i.e., if a great white shark is recognized it is also expected higher activations from classes as tigers and hammerhead sharks as well as whales, but very weak or no activations from birds or baseballs.

Observing the fooling images, while the artificial construction increases the class of interest's probability, the image generation process does not simultaneously adjust the scores of all related classes, resulting in an AV that is far from the model AV. To build a fooling image that is not rejected means not only getting a high score for the class of interest, it means maintaining the relative scores for the other classes.

As results, the experiments were able to show that OpenMax consistently obtains higher F-measure on open set testing, outperforming open set recognition accuracy of basic deep networks as well as deep networks with thresholding of SoftMax probabilities. Interestingly, it was also observed that the OpenMax rejection process often identifies and rejects the ImageNet images that the deep network incorrectly classified, especially images with multiple objects.

D. Nearest Neighbors Distance Ratio [4]

Solutions proposed for closed set applications in literature, as Nearest Neighbor (NN), are not going to be as successful for real-world recognition problems because in the latter situation new classes can me added. However, NN is widely used because of its simplicity,parameter independence and multiclass distinctiveness. For this reason, [4] proposes a method named Open-Set NN (OSNN), which is able to recognize samples from unknown classes during training time that outperform other approaches on the literature.

OSNN method verify if the test sample can be classified as unknown, using the ratio of similarity scores to the two most similar classes by applying a threshold on it. One of the advantages of this approach is that it is inherently multiclass, what means it is not affected as the number of classes for training increases. Moreover, the method is able to create a bounded KLOS (known labeled open space) in order to protect the classes of interest and reject unknown classes.

NN classifier is divided in two different open-set extensions according to [4]: Class Verification, also called $OSNN^{CV}$ and Nearest Neighbor Distance Ratio or simply OSNN.

- *Class Verification:* The training phase of $OSNN^{CV}$ is similar to NN. During the prediction phase, the test sample s is analyzes to check if both nearest neighbors have the same label. In case it is true, the label is assigned to s. Otherwise, s is assigned as an unknown class.
- *Nearest Neighbor Distance Ratio:* This method collects the nearest neighbor *t* and *u* from the test sample *s* such as $\theta(u) \neq \theta(t)$, which $\theta(x) \in \zeta = \{l_1, l_2, ..., l_n\}$. This expression represents the class of a sample *s* and ζ is the set of training labels. Given that, the ratio is calculated by the formula:

$$R = d(s,t)/d(s,u) \tag{2}$$

Where d(x,x') is the Euclidian distance between samples x and x' in the feature space. The value of R should be compared with a threshold T (0.0 < T < 1.0) for the purpose of classifying the sample according to the following rule: *s*.

$$\theta(s) = \begin{cases} \theta(t), & \text{if } R \leq T\text{, s has the same label of t} \\ l_0, & \text{if } R > T\text{, s is unknown} \end{cases}$$

Ultimately, if *s* is also faraway from the training sample, it is assigned as an unknown class.

In the interest of finding the best value for T in an open-set scenario, a *parameter optimization* is performed. For this purpose, a simulation of an open-set environment is established. Fot that, a training set *F* is created with half of known classes. In addition, a validation set *V* receives the other half of known classes and also all instances of unknown classes. After all, *T* is based is based in the accuracy of the validation set *V*. Details of this operation can be seen in figure 3.

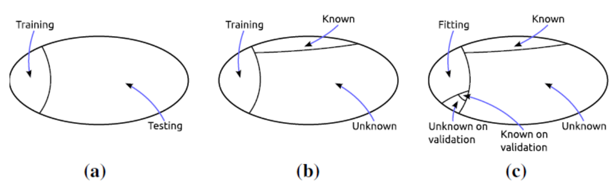

Fig. 3: Scheme of data partitioning for the experiments and the parameter optimization of the OSNN. **a** A dataset is divided into training and testing sets. **b** Most of the samples in testing set are unknown. **c** Partitioning of the training set by simulating an open-set scenario.

E. Generative OpenMax for Multi-Class Open Set Classification [5]

In proposed method Generative OpenMax, called G-OpenMax [5], extends OpenMax [3] by providing explicit probability estimation over unknown categories. This is done by using generative adversarial networks (GANs), which initially are a technique to estimate models via an adversarial process between two neural networks, for this paper [5], we use to generate the unknown classes. The synthetic samples are create by mixture distributions of known classes in space.

That is, while OpenMax estimate the pseudo probability of unknown class using aggregating calibrated score from known classes, the G-OpenMax which is an intuitive solution, directly estimate the probability of unknown class. Being performed through of synthetic images as an extra training label apart from known labels.

In G-OpenMax in parallel to the Open-Max, we train the network Net^{G} with extra class where the extra images come from our generator G to represent the unknown class, as can be seen in 4

The experiments proposal through comparison of G-OpenMax with vanilla OpenMax using two datasets: MNIST with digits of 10 classes from 0 to 9 and HASYv2 with handwritten symbol in 369 classes, including numbers, Arabic and Latin characters. HASYv2 is not a balanced dataset, then a preprocessing was done. For the MNIST, 6 classes are hold out in training and the rest 4 are used to varying "openness". For HASYv2, 60 classes for the training and the rest 35 are held out for testing unknown classes. The results are very positives for the two datasets.

In tests with datasets with different domain between the classes, like ImageNet, this method did not perform well because the generate images are not plausible with respect to the training classes in order to be good candidates to represent unknown classes from open space. That is, for datasets with all classes share common features, this method is a good option.

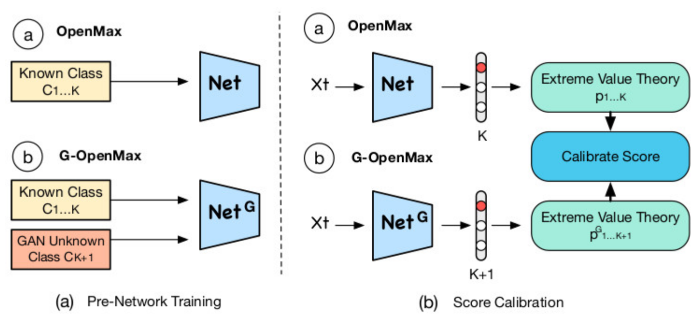

Fig. 4: a) Illustrates the pre-training process of Net and Net^{G}. GAN-based synthetic images are used as an extra training label. b) Explains the difference between score calibration in normal OpenMax and G-OpenMax.

IV. Conclusion

In the last few years open-set scenario has received the proper attention, experiments in literature that were performed considering that all classes of the problem are available for training, started to change their focus. This new methods are more accurate for real-world situations, where the amount of classes during test is many times larger than the known classes.

In this summary we presented five approaches that has some different techniques and concepts used to solve the open-set problem in closed-set state-of-art algorithms. Besides those, many other researches are looking into it, and the area still has a lot of room for future development.

References

[1] A. Bendale and T. Boult, "Towards open world recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[2] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult, "The extreme value machine," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[3] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1563–1572.

[4] P. R. M. Júnior, R. M. de Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha, "Nearest neighbors distance ratio open-set classifier," *Machine Learning*, vol. 106, no. 3, pp. 359–386, 2017.

[5] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," *CoRR*, vol. abs/1707.07418, 2017.