Few Sample Learning without Data Storage for Lifelong Stream Mining (Student Abstract)

Zhuoyi Wang, Yigong Wang, Yu Lin, Bo Dong, Hemeng Tao, Latifur Khan
Department of Computer Science
University of Texas at Dallas, Richardson TX
{zxw151030, yxw158830, yxl163430, Bo.Dong3, hxt160430, lkhan}@utdallas.edu

Abstract

Continuously mining complexity data stream has recently been attracting an increasing amount of attention, due to the rapid growth of real-world vision/signal applications such as self-driving cars and online social media messages. In this paper, we aim to address two significant problems in the lifelong/incremental stream mining scenario: first, how to make the learning algorithms generalize to the unseen classes only from a few labeled samples; second, is it possible to avoid storing instances from previously seen classes to solve the catastrophic forgetting problem? We introduce a novelty stream mining framework to classify the infinite stream of data with different categories that occurred during different times. We apply a few-sample learning strategy to make the model recognize the novel class with limited samples; at the same time, we implement an incremental generative model to maintain old knowledge when learning new coming categories, and also avoid the violation of data privacy and memory restrictions simultaneously. We evaluate our approach in the continual class-incremental setup on the classification tasks and ensure the sufficient model capacity to accommodate for learning the new incoming categories.

1 Introduction

Stream mining is a lifelong/incremental learning process(Silver, Yang, and Li 2013) in real-world AI systems, they acquire knowledge throughout the whole process to become more efficient and versatile facing new knowledge. Under this scenario, the instances with different classes would occur during different periods. While Deep Neural Network (DNN)(He et al. 2016) has achieved remarkable results on relative novelty detection tasks, there are several practical obstacles of DNNs as an effective lifelong system for stream mining: (1). Typically, when the new/novel class occurred, existing approaches tend to request human-annotated samples to learn new knowledge. However, the annotation cost of human is significantly limit the applicability of the stream mining approaches to acquire new knowledge more efficiently; (2) typically DNNs may fail to continually learn from a stream of data while maintaining old knowledge (also called the strophic forgetting problem), existing model update strategies typically need to store the seen classes' samples, which would violate data privacy and memory restrictions of the real-world applications. It is thus of great interest to propose a more powerful stream mining approach to learn new categories with only a few labeled samples, and maintain the previous knowledge with no access to previously samples simultaneously (without the storage of previous instances).

This paper proposes a few-sample learning and incremental generative model based stream mining framework, we aim to address the human annotation and memory restriction challenge in the previous description. The learning system should be continually trained from the non-stationary data stream, where new task/category samples become available at different times. Our approach applies the meta-learning paradigm that learn to generalize the new emerged classes from only a few labeled samples, the core mechanism is to extract the transferable knowledge from a collection of existing tasks to make the model fast adaption on the unseen data/domain; at the same time, we implement a trainable generative model through the attention masks for the network layer weights, incrementally learn new knowledge from the novelty categories without the necessary to replay previous knowledge to its generator.[1]

2 Approach

Given a training data set $D = \{(x_i, y_i)\}_{i=1}^{M}$, $y_i \in Y = \{1, 2, \ldots, c\}$ is the associated class label, for the lifelong streaming data $S = \{(x_t, y_t)\}_{t=1}^{\infty}$, here both $x_i, x_d \in \mathcal{R}^d$, and $y_t \in Y' = \{1, 2, \ldots, c, c+1, \ldots, c'\}$. The instances with different classes would occur at different period.

2.1 Few Sample Learning

This module is learning to recognize new classes during the stream process, with only a small amount of labeled novel class instances N. In the training step, we learn a DNN based feature extractor f_θ, a classifier with weight vector $W_D \in \mathcal{R}^{d\times c}$ (where c is the number of class in D). When the emerged novel classes are detected, the network parameters would be fixed to extract features from such categories, the new classifier with weight matrix W_N would be learned. Here we regard the W_D as $[w_1, w_2 \ldots w_c]$,

[1] We release the supplemental material and source code at: https://github.com/Vitvicky/Continual_Learning_For_Stream

each of the weight vector is a d-dimensional feature vector, and could be interpreted as prototypes(Snell, Swersky, and Zemel 2017). For the training samples $x_i \in D$, we compute the similarity scores $S(x_i, w_j)$ from x_i to each prototype $w_j, j \in [1, 2 \dots c]$ as following:

$$S_{x_i,w_j} = \frac{f_\theta(x_i)^\intercal w_j}{||f_\theta(x_i)|| \cdot ||w_j||} \tag{1}$$

The predicted probability for each prototype (class) could be obtained by normalizing similarity scores S_{x_i,w_j} with a softmax function, it based on the distance between the input feature and the prototype representing each class. Finally, we train the model with this distance-based classifier, through minimizing the margin-based distance loss: $L_{pred} = \sum_{y_i=y_j,y_i \neq y_k} \log(1+\exp(S_{x_i,w_j} - S_{x_i,w_k} + m))$, it could explicitly reduce intra-class variation during the learning step. Next, to recognize new classes in the fine-tuning stage, we fix f_θ and learn a new weight matrix W_N through the meta-learning algorithm(Vinyals et al. 2016). It constructs few-sample tasks from the new categories, each task comprised of the support and query sets (also called an episode). The meta-learning process would optimize the new model parameters on these tasks.

2.2 Incremental Generative Model

We introduce an incremental Generative Model G, which aims to memorize the distributions of previously seen data, also one type of the experience replay based approach(Shin et al. 2017). The instances from the seen classes are synthesized by G and then replayed to the discriminator D at each step of incremental learning. Our approach implements the generative model through the conditional generative adversarial network (GAN)(Goodfellow et al. 2014) consisted of L layers, we jointly train the G and a mask function for the weights of its generator. During the training of each task t (every task can be composed of a collection of items from larger or equals with single class), our approach would learn different binary mask functions M^t ($M^t = [m_1^t, \dots m_l^t]$) for each layer's weight value in the G. And the output of each connected layer l could be generated by combining the mask function M_l^t with the weights of different layers as following:

$$y_l^t = \sigma[(M_l^t \circ W_l)^\intercal x] \tag{2}$$

Here σ is some activation function, W_l is the weight matrix applied in G, it is between layer l and $l-1$, and $\circ$ means the Hadamard product. The learned mask function could be applied to connect the plasticity of neurons, thus avoiding overwriting of significant units by restricting SGD updates to the parameter segments of G that exhibit free capacity.

3 Experiment

We perform experiments to measure the classification accuracy of our approach on simulated class-incremental stream scenario, the benchmark datasets CUB200 and CIFAR10 are selected for the experiment, we randomly split the dataset into 50% training classes, and 50% novel classes, then simulate stream by randomly selecting 500 instances from 2 novel classes after a period of time until all available instances are selected to form the stream. In the meta-training stage for meta-learning methods, we pick k (k=1,5) instances from each novel class as the support set, the generator is based on a 3-layer conditional GAN, all approaches are trained from scratch and use the Adam optimizer with a four-layer convolution backbone (Conv-4), the initial learning rate is 0.001.

Table 1: Classification results (%) for CIFAR10 and CUB datasets, - means fail to execute.

Methods	**CUB**		**CIFAR10**	
	1-Sample	5-Sample	1-Sample	5-Sample
SAND	–	19.21±1.25	–	20.82±1.22
DNNs(fine-tuning)	46.93±0.72	63.85±0.67	50.12±0.69	69.55±0.61
P-Net(Vinyals et al. 2016)	51.37±0.88	70.17±0.72	53.07±0.83	71.48±0.69
Ours	**60.23±0.91**	**79.17±0.70**	**56.62±0.76**	**74.61±0.63**

We report an average classification accuracy (%) over the whole test sets of classes seen so far during the training for ten times, we show the result for 1-sample and 5-sample classification, (1 or 5 labeled samples are selected from each novel class), and show them in Table 1 for the CUB and CIFAR-10 dataset. It is obvious that our approach is better than other baselines (SAND(Haque, Khan, and Baron 2016), DNNs with fine-tuning and P-Net) by a large margin in the stream classification setting. Other relative results could be seen in the supplemental materials.

4 Conclusion and Future Work

In this work, we propose few-sample learning with a class-incremental generator approach for class-incremental stream mining. The results show that our approach could be adapted into a novel class with only a few request samples, and the catastrophic forgetting could also be solved through the generator used as a memory module with weight masking.

References

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Haque, A.; Khan, L.; and Baron, M. 2016. Sand: Semi-supervised adaptive novel class detection and classification over data stream. In *AAAI*, 1652–1658.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *NIPS*, 2990–2999.

Silver, D. L.; Yang, Q.; and Li, L. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NIPS*, 4077–4087.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NIPS*, 3630–3638.