

Central-diffused Instance Generation Method in Class **Incremental Learning**

Mingyu Liu, Yijie Wang*

National University of Defense Technology, China {liumingyu13, wangyijie}@nudt.edu.cn

Background

- ✓ In a real environment, new classes of the emerging instances may appear at any time while instances keep arriving. For this circumstance, class incremental learning is proposed as a machine learning framework to handle the new class instances incrementally.
- ✓ In many real incremental scenarios, the distribution between new class instances and the existing ones is not well balanced. Therefore, how to solve the class imbalance learning effectively during the process of class incremental learning is a meaningful research.

Motivations

- ✓ Many techniques have been proposed to deal with class imbalance problem, but they are poor at dealing with the emerging class scenario.
- ✓ Some research focuses on discovering emerging new classes which can be defined as anomaly classes in data stream, but they cannot solving the class imbalance problem simultaneously.

Class Incremental Learning via CdIGM

✓ Framework of Class Incremental Learning(Algorithm 1) The final goal of class incremental learning is to find a map $f: X \to Y$, so as to minimize the loss over S_K :

$$\sum_{j=1}^{K} I(f(x_i) \neq y_i), i \in \{1, 2, \dots, n_j\}$$

Algorithm 1. The Framework of Class Incremental Learning

Input: Sequence S_k ; number set N; the number of the existing classes M. **Output:** Weight W for prediction.

- 1: Receive supervised instances sets $\{\{(x_i, y_j)\}_{i=1}^{n_j}\}_{i=1}^{M}$ from sequence S_k ;
- 2: $T \leftarrow \left\{ \left\{ \left(x_i, y_j \right) \right\}_{i=1}^{n_j} \right\}_{i=1}^M$;
- $W \leftarrow \text{Train multi-classification model for } T$;
- Repeat:
- Receive a new instance set $\{\{(x_i, y_j)\}_{i=1}^{n_j}\}_{i=1}^M$ from sequence S_k ; 5:
- $S \leftarrow \left\{ \left\{ (x_i, y_{M+1}) \right\}_{i=1}^{n_j} \right\}_{i=1}^M;$
- $T \leftarrow CdIGM(W,T);$
- $M \leftarrow M + 1$;
- $W \leftarrow Update(W,T);$
- 10: Until M = K;

✓ Central-Diffused Instance Generation Method (CdIGM)

Receiving instances of a new emerging class Update the center vector set C. Generate the direction vectors randomly and normalize each one to the step Length L. The center falls into any existing class? Yes No Generate instances for each Loop to generate instances for vector within the distribution each direction vector evenly of class K until the synthetic until the synthetic instance q instance falls out of it to get falls into any known class. new instance set G. Iterate to merge the new synthetic instance set with the old one to form a final synthetic instance set.

Experiments

Table 1. Overall Accuracy on Data Streams in Experiments.

	SMOTE	OPCIL	OB	SDCIL
94.23±0.084	88.11 <u>+</u> 0.108	82.25 <u>+</u> 0.486	84.49 <u>+</u> 0.393	79.18 <u>+</u> 0.296
00.84 <u>+</u> 0.035	83.89 <u>+</u> 0.075	76.26 <u>+</u> 0.349	80.34 <u>+</u> 0.486	72.67 <u>+</u> 0.834
36.64±0.079	80.19±0.234	72.34±0.495	69.01±0.199	65.81±0.474
91.33 <u>+</u> 0.934	85.29±0.278	87.52 <u>±</u> 1.134	84.19±0.157	82.58 <u>+</u> 0.837
93.24 <u>+</u> 0.259	90.58 <u>±</u> 1.147	92.11±0.824	89.99±0.938	85.46 <u>+</u> 0.630
74.13 <u>+</u> 1.597	78.43±0.392	73.90 <u>+</u> 0.957	63.65±0.295	71.96 <u>+</u> 0.301
88.15 <u>+</u> 0.949	84.36 <u>+</u> 0.395	80.65 <u>+</u> 0.284	74.99±0.851	80.03 <u>+</u> 0.047
37.21 <u>+</u> 0.385	82.11 <u>±</u> 0.884	78.45 <u>+</u> 0.955	80.34±0.751	76.03 <u>+</u> 0.937
79.31±0.753	68.48±0.438	74.57 <u>±</u> 0.418	64.23±0.281	58.92 <u>+</u> 0.431
55.45±0.413	61.44±0.294	60.17 <u>±</u> 0.538	59.75±0.369	55.51 <u>+</u> 0.199
3(1) 3(1) 3(1) 3(1)	0.84±0.035 6.64±0.079 1.33±0.934 3.24±0.259 4.13±1.597 8.15±0.949 7.21±0.385 9.31±0.753	0.84±0.035 83.89±0.075 6.64±0.079 80.19±0.234 1.33±0.934 85.29±0.278 3.24±0.259 90.58±1.147 4.13±1.597 78.43±0.392 8.15±0.949 84.36±0.395 7.21±0.385 82.11±0.884 9.31±0.753 68.48±0.438	0.84±0.035 83.89 ± 0.075 76.26 ± 0.349 6.64±0.079 80.19 ± 0.234 72.34 ± 0.495 1.33±0.934 85.29 ± 0.278 87.52 ± 1.134 3.24±0.259 90.58 ± 1.147 92.11 ± 0.824 4.13±1.597 78.43 ± 0.392 73.90 ± 0.957 8.15±0.949 84.36 ± 0.395 80.65 ± 0.284 7.21±0.385 82.11 ± 0.884 78.45 ± 0.955 9.31±0.753 68.48 ± 0.438 74.57 ± 0.418	0.84 \pm 0.03583.89 \pm 0.07576.26 \pm 0.34980.34 \pm 0.4866.64 \pm 0.07980.19 \pm 0.23472.34 \pm 0.49569.01 \pm 0.1991.33 \pm 0.93485.29 \pm 0.27887.52 \pm 1.13484.19 \pm 0.1573.24 \pm 0.25990.58 \pm 1.14792.11 \pm 0.82489.99 \pm 0.9384.13 \pm 1.59778.43 \pm 0.39273.90 \pm 0.95763.65 \pm 0.2958.15 \pm 0.94984.36 \pm 0.39580.65 \pm 0.28474.99 \pm 0.8517.21 \pm 0.38582.11 \pm 0.88478.45 \pm 0.95580.34 \pm 0.7519.31 \pm 0.75368.48 \pm 0.43874.57 \pm 0.41864.23 \pm 0.281

✓ The results show that CdIGM averagely achieves 4.01%, 4.49%, 8.81% and 9.76% performance improvement over SMOTE, OPCIL, OB and SDCIL, respectively. It is proved to possess the strength of class incremental learning and class imbalance learning with good accuracy and robustness.

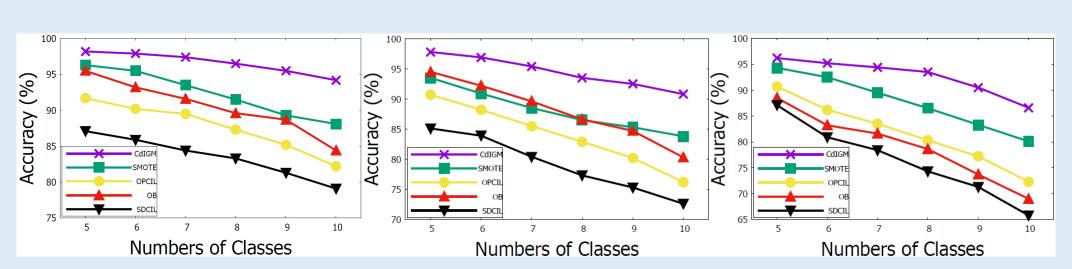


Fig. 1. Real-time Accuracy on Artificial Data Streams with Different Imbalance Rate