ICPRAI'22
3rd International Conference on Pattern Recognition and Artificial Intelligence - 2022

KLETech CEVI Center of Excellence in Visual Inteligence

Progressive Clustering: An Unsupervised Approach Towards Continual Knowledge Acquisition of Incremental Data



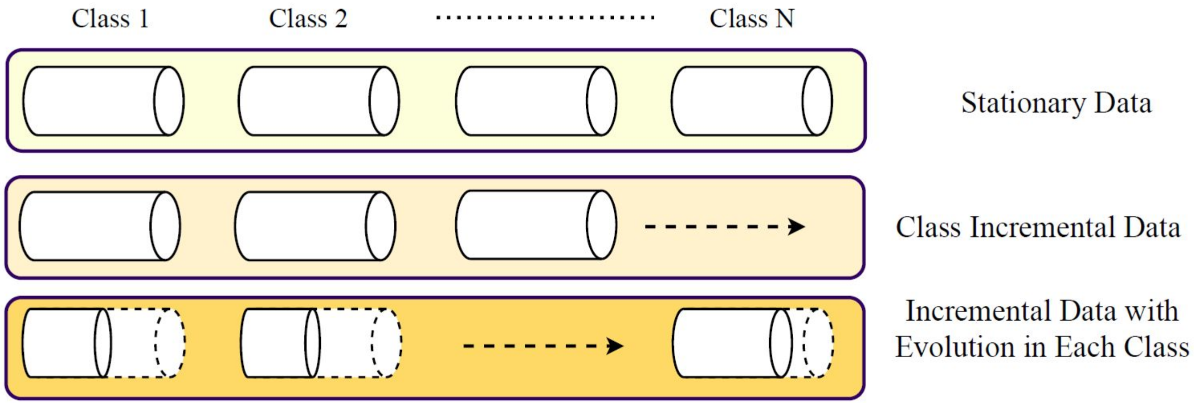

Akshaykumar Gunari | Shashidhar Kudari | Ramesh Ashok Tabib | Uma Mudenagudi

Center of Excellence in Visual Intelligence (CEVI)
KLE Technological University, Hubli, India

Contents:

Data Acquisition in Real World

- Various fields of data mining and machine learning applications involves clustering as their principal component, considering non incremental nature of the data.
- Existing algorithms lack to capture temporal dependencies in a natural, data-driven manner.
- In addition, the model needs to acquaint to the continuous change in the distribution of the input data.
- Dynamically growing data requires models preservation of previously learnt knowledge and acquire new knowledge.
- Towards this, we design algorithm to generate data that has different number of classes in each phase with varied sample size from each class.

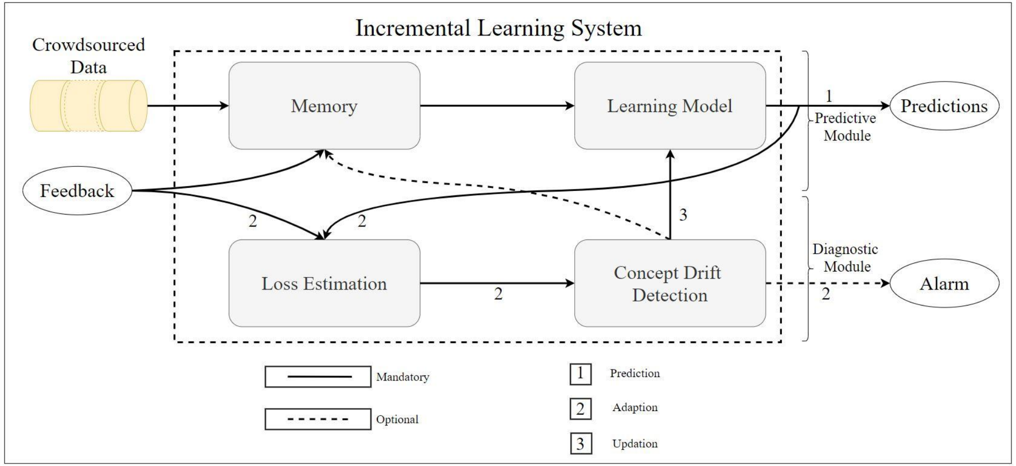


Data Acquisition in Real World

- Class Incremental Learning:
  - the number of classes across different phases is fixed;
  - classes appearing in earlier phases will not appear in later phases again;
  - training samples are well balanced across different classes in each phase.

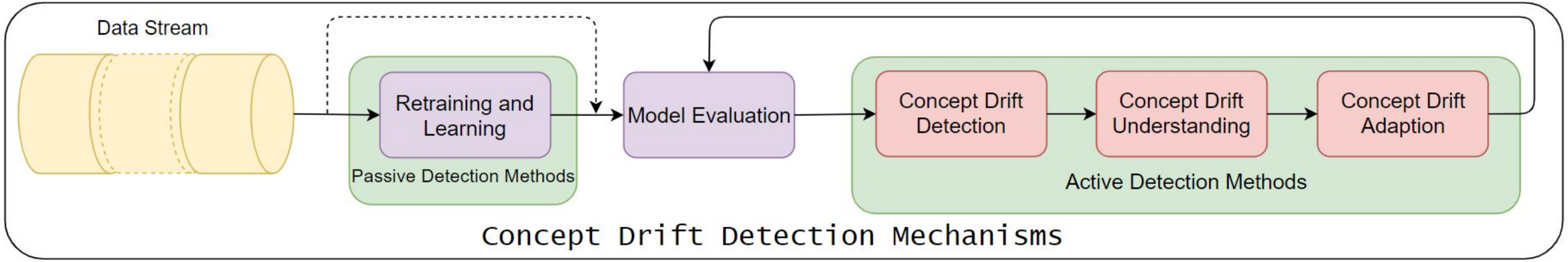
- Objectives:
  - To address incremental data to identify evolving clusters.
  - Concept Drift detection and handling.

- There is a need of strategies to handle the incremental nature of the data
  - that clusters the current data chunk from antecedent models knowledge.
  - that adapt to the change in the the behaviour of the data over time.
  - to design deep dynamically growing models that adjust itself with the distribution of the dataset.



Concept Drift

- In predictive analytics and machine learning, the concept drift means that the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes.
- To prevent deterioration in prediction accuracy because of concept drift:
  - Passive Methods
    - The model is continuously updated: By retraining the model on the most recently observed samples or enforcing an ensemble of classifiers.
  - Active Methods
    - Rely on triggering mechanisms: To explicitly detect concept drift as a change in the statistics of the data-generating process.



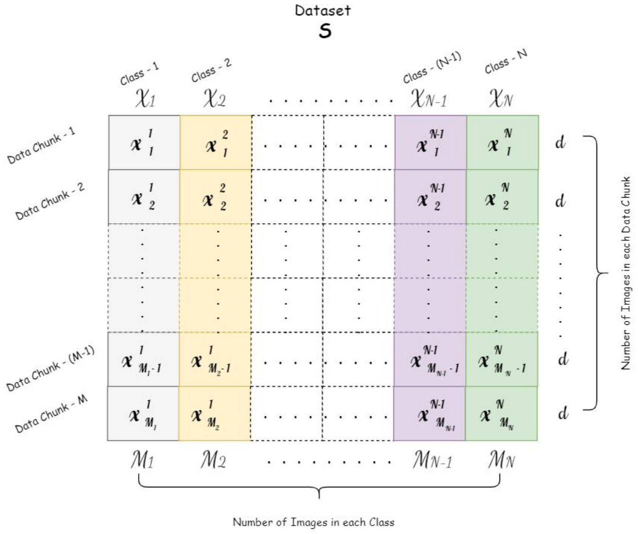
Incremental Data Generation

Algorithm 2: Incremental Data Generation for Progressive Clustering

Input : Dataset S as $S = \sum_{i=1}^{N}\{X_i\}$, $X_i = \sum_{j=1}^{M_i}\{x_j^i\}$ where N=Number of Classes, X_i = Class i, $n = \sum_{k=1}^{N}\{M_k\}$, M_k=Number of Images in class k, m=Number of Images in each data Chunk.

Output: Incremental Dataset $D = \sum_{i=1}^{N_c}\{D_{c_i}\}$, where N_c = Number of Data Chunks, D_{c_i} is data chunk i.

$D = \varnothing$ ▷ Initialize an empty set to store data chunks.
$c = m$
for $i \leftarrow 1$ **to** N_c **do**
    $d = \varnothing$ ▷ For each data chunk
    $t = random(1, N)$ ▷ Number of classes to choose in data chunk D_{c_i}
    for $j \leftarrow 1$ **to** t **do**
        $s = random(10,\ c)$ ▷ Number of samples to choose in class j
        $d_c = random\_collection(X_j, s)$ ▷ Choose s samples from class j
        $updateDataChunk(d)$ ▷ Add d_c to d
        $X_i = X_i - d_c$
        $c = m - c$
    end for
    $addDataChunk(d, D)$ ▷ Add data chunk d to D
end for
return D



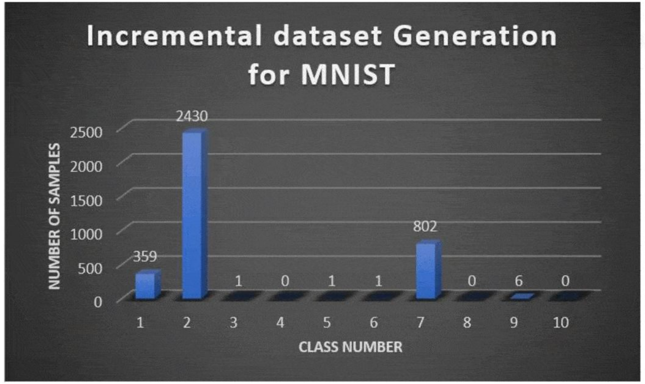
Incremental Data Generation

Algorithm 2: Incremental Data Generation for Progressive Clustering

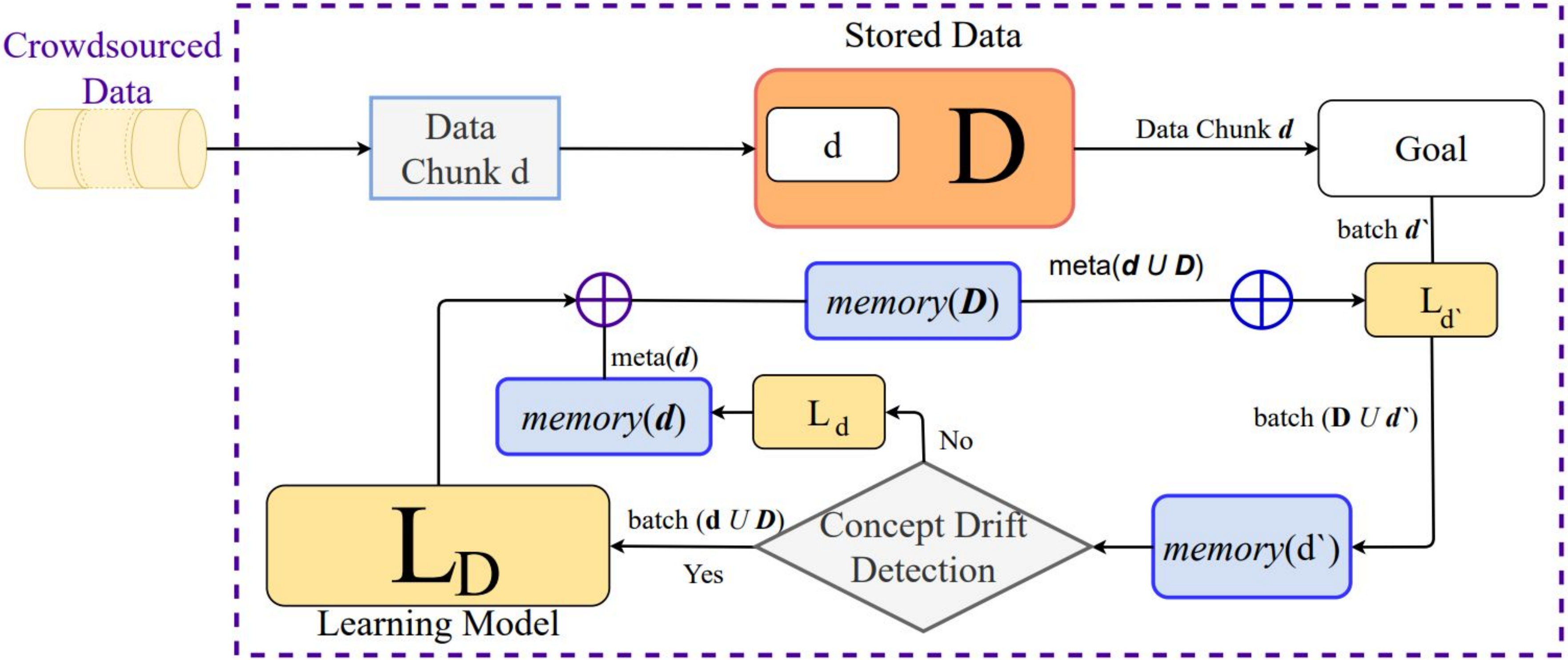
Input : Dataset S as $S = \sum_{i=1}^{N}\{X_i\}$, $X_i = \sum_{j=1}^{M_i}\{x_j^i\}$ where N=Number of Classes, X_i = Class i, $n = \sum_{k=1}^{N}\{M_k\}$, M_k=Number of Images in class k, m=Number of Images in each data Chunk.

Output: Incremental Dataset $D = \sum_{i=1}^{N_c}\{D_{c_i}\}$, where N_c = Number of Data Chunks, D_{c_i} is data chunk i.

$D = \varnothing$ ▷ Initialize an empty set to store data chunks.
$c = m$
for $i \leftarrow 1$ **to** N_c **do**
    $d = \varnothing$ ▷ For each data chunk
    $t = random(1, N)$ ▷ Number of classes to choose in data chunk D_{ci}
    for $j \leftarrow 1$ **to** t **do**
        $s = random(10,\ c)$ ▷ Number of samples to choose in class j
        $d_c = random\_collection(X_j, s)$ ▷ Choose s samples from class j
        $updateDataChunk(d)$ ▷ Add d_c to d
        $X_i = X_i - d_c$
        $c = m - c$
    end for
    $addDataChunk(d, D)$ ▷ Add data chunk d to D
end for
return D



Data Chunk 01


Crowdsourced Data
Stored Data
Data Chunk d
d
D
Data Chunk d
Goal
batch d`
meta(d U D)
memory(D)
L_{d`}
meta(d)
memory(d)
L_d
No
batch (D U d`)
L_D
batch (d U D)
Concept Drift Detection
memory(d`)
Yes
Learning Model

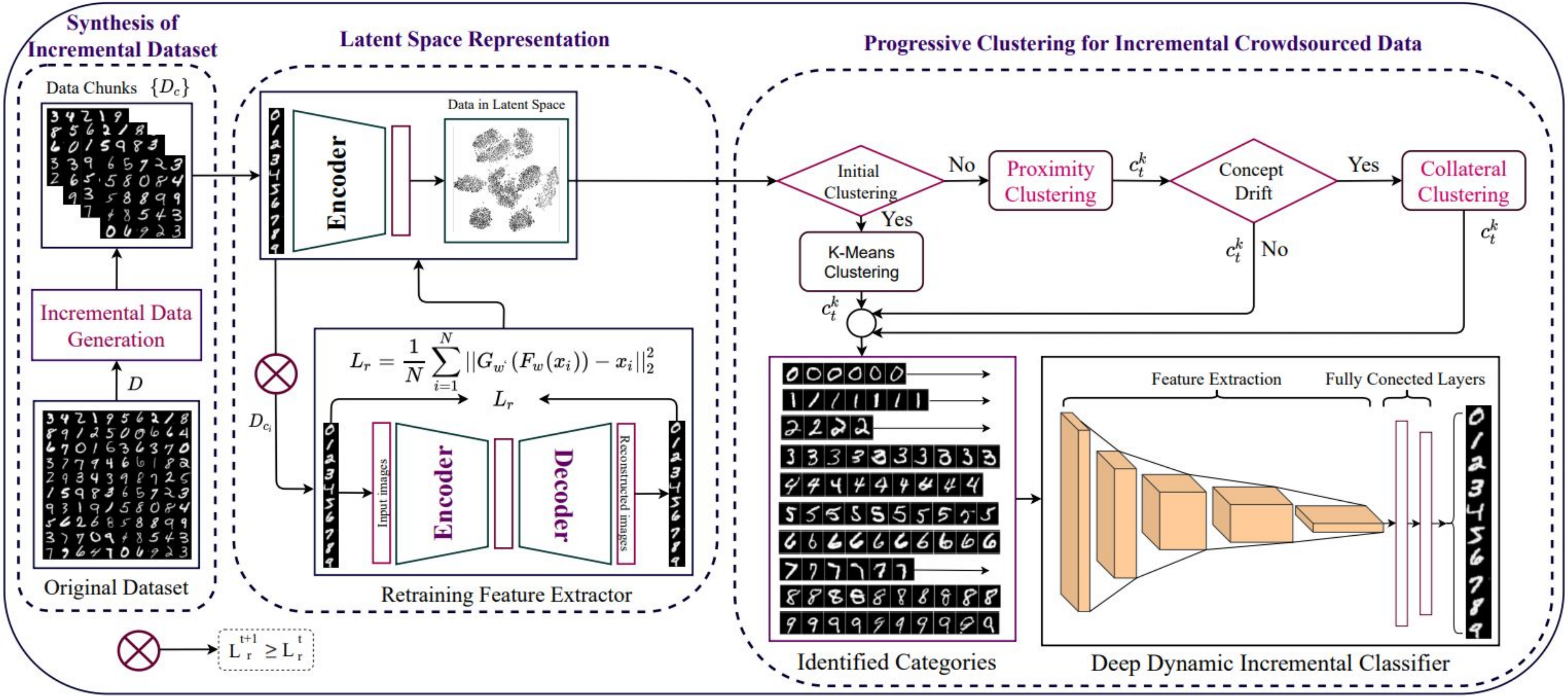
Progressive Clustering

Algorithm 1: Progressive Clustering

Input : Incremental Dataset $D = \sum_{i=1}^{N_c} \{D_{c_i}\}$, where N_c = Number of Data Chunks.

Output: Clusters $C = \{c_1, c_2, c_3 \dots c_k\}$

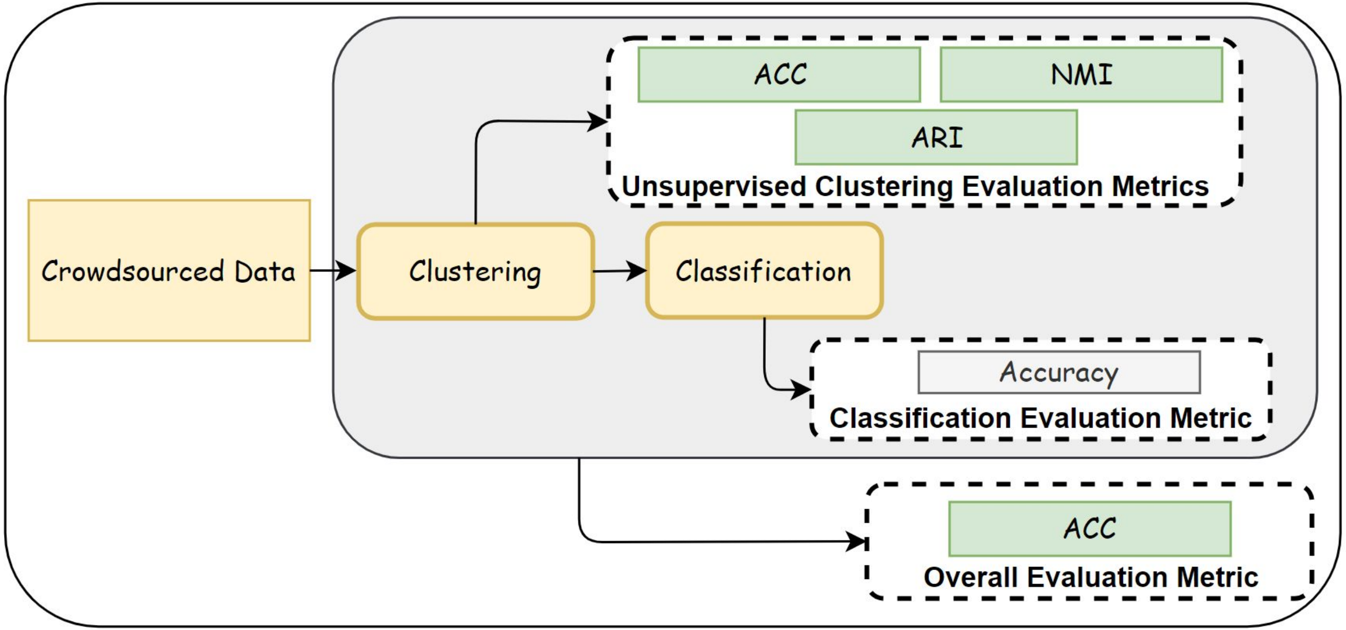
$update(\phi_{\theta_i}, D_{c_i})$
$C_t \leftarrow kmeans(D_{c_i}, k)$
for $i \leftarrow 2$ **to** N_c **do**
    if $(L_{r+1}) \leqslant \eta \cdot L_r$ **then**
        $update(\phi_{\theta_i}, D_{c_i})$
    end if
    $d_{c_i} \leftarrow getEmbeddings(D_{c_i}, \phi_{\theta_i})$
    if $ConceptDrift(d_{c_i})$ **then**
        $C_t \leftarrow kmeans(D_{c_i}, k)$
    end if
    $C_t \leftarrow proximityClustering(C_{t-1}, d_{c_i})$
end for
$C \leftarrow C_t$
return C


Synthesis of Incremental Dataset
Data Chunks $\{D_c\}$
Incremental Data Generation
D
Original Dataset
Latent Space Representation
Data in Latent Space
Encoder
D_{c_i}
$L_r = \frac{1}{N}\sum_{i=1}^{N} ||G_{w'}(F_w(x_i)) - x_i||_2^2$
L_r
Input images
Encoder
Decoder
Reconstructed images
Retraining Feature Extractor
$L_r^{t+1} \geq L_r^t$
Progressive Clustering for Incremental Crowdsourced Data
Initial Clustering
No
Yes
Proximity Clustering
c_t^k
Concept Drift
Yes
No
Collateral Clustering
K-Means Clustering
Identified Categories
Feature Extraction
Fully Conected Layers
Deep Dynamic Incremental Classifier

Dataset and Training Details

- Dataset:
  - MNIST and Fashion-MNIST.
  - Consists of 70k grayscale images of resolution (28 x 28) pixels belonging to 10 classes.
  - We divide the datasets into the data chunks of 7k images as discussed in Algorithm 2 from each dataset which serves as the incremental data.
  - We generated incremental dataset containing 10 data chunks for both MNIST and Fashion-MNIST dataset. Each data chunk contained different number of samples from each class.
- Training:
  - Runtime Environment: Nvidia GetForce GTX 1060.
  - Architecture: Autoencoder.
  - Learning Rate: 0:01.
    -
  - Numb $Conv^{5}_{32} \longrightarrow Conv^{5}_{64} \longrightarrow Conv^{3}_{128} \longrightarrow F_{C_{10}} \longrightarrow Conv^{3}_{128} \longrightarrow Conv^{5}_{64} \longrightarrow Conv^{5}_{32}$
  - Optimizer: Adam

Evaluation Methodology



Results of Progressive Clustering

Methodology	Number of times reclustered	Dataset							
		MNIST				Fashion-MNIST			
		ACC	NMI	ARI	Accuracy	ACC	NMI	ARI	Accuracy
K-Means	10	0.5385	0.4680	0.3229	0.9524	0.4737	0.5116	0.3473	0.9498
SEC	10	0.8037	0.7547	0.6542	0.9587	0.5124	0.5008	0.4245	0.9587
SAE+k-means	10	0.7817	0.7146	0.8658	0.9564	0.5370	0.5563	0.5474	0.9429
CAE+k-means	10	0.8490	0.7927	0.8798	0.9584	0.5833	0.6084	0.4449	0.9587
DEC	10	0.8408	0.8128	0.7831	0.9655	0.518	0.546	0.5139	0.9327
IDEC	10	0.8421	0.8381	0.5406	0.9587	0.529	0.557	0.4098	0.9547
DEC-DA	10	0.9861	0.9622	0.9447	0.9651	0.586	0.636	0.5484	0.9645
DCEC	10	0.8897	0.8849	0.5319	0.9691	0.584	0.638	0.5156	0.9459
Progressive k-means	**8**	0.5221	0.4631	0.2965	0.9324	0.4583	0.4587	0.3327	0.9149
Progressive SEC	**8**	0.7424	0.7021	0.6954	0.9234	0.4464	0.3984	0.3547	0.9258
Progressive SAE+k-means	**8**	0.7124	0.6857	0.7894	0.9132	0.4865	0.5132	0.5474	0.9174
Progressive CAE+k-means	**6**	0.8126	0.6972	0.8123	0.9129	0.5514	0.5127	0.4415	0.9107
Progressive DEC	**6**	0.7792	0.7462	0.7536	0.9097	0.4997	0.5165	0.4741	0.9057
Progressive IDEC	**6**	0.8056	0.7862	0.5125	0.9234	0.5174	0.5475	0.3687	0.9157
Progressive DEC-DA	**6**	0.9157	0.9165	0.9014	0.9324	0.5547	0.6234	0.5654	0.9268
Progressive DCEC	**6**	0.8265	0.7896	0.5123	0.9557	0.5844	0.5672	0.5074	0.9215

Contributions and Conclusion

- In this paper, we proposed a categorization strategy to handle the incremental nature of the data by identifying concept drift in the data stream.
- Our method automatically discovers newly occurring object categories in unlabelled data and is used to train a classifier that can be used for various downstream tasks such as content based image retrieval systems, image data segregation etc.
- We proposed an algorithm to alleviate the problem of concept drift by designing progressive clustering algorithm capable of handling continually arriving data.
- We demonstrated our results on standard MNIST and Fashion-MNIST datasets to show our methodology shows comparable performance to state-of-the-art clustering algorithms which will have to be trained from scratch on the arrival of each data chunk.
- Deploying incremental learning algorithms for critical applications warrants circumspection and is still a work in progress and we believe our work is a step in this direction.

References

[1] Belouadah, E., Popescu, A.: Il2m: Class incremental learning with dual memory. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp.583–592 (2019). https://doi.org/10.1109/ICCV.2019.00067

[2] Castro, F.M., Mar´ın-Jim´enez, M.J., Mata, N.G., Schmid, C., Karteek, A.: End-to end incremental learning. ArXiv abs/1807.09536 (2018)

[3] Chen, J., Zhang, L., Liang, Y.: Exploiting gaussian mixture model clustering for full-duplex transceiver design. IEEE Transactions on Communications 67(8), 5802-5816 (2019). https://doi.org/10.1109/TCOMM.2019.2915225

[4] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Computing Surveys 46(4) (2014). https://doi.org/10.1145/2523813.

[5] Kuncheva, L.: Classifier ensembles for changing environments. vol. 3077, pp. 1–15 (06 2004). https://doi.org/10.1007/978-3-540-25966-41

[6] Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(12), 2935–2947 (2018). https://doi.org/10.1109/TPAMI.2017.2773081.

[7] Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. pp. 7765–7773 (06 2018). https://doi.org/10.1109/CVPR.2018.00810.

References

[8] Moulton, R.H., Viktor, H.L., Japkowicz, N., Gama, J.: Clustering in the presence of concept drift. In: Berlingerio, M., Bonchi, F., G¨artner, T., Hurley, N., Ifrim, G. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 339–355. Springer International Publishing, Cham (2019)

[9] Palacio-Ni˜no, J.O., Galiano, F.: Evaluation metrics for unsupervised learning algorithms. ArXiv abs/1905.05667 (2019)

[10] Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5533–5542 (2017). https://doi.org/10.1109/CVPR.2017.587.

[11] Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks (2016)

[12] Sammut, C., Webb, G.I. (eds.): Mean Squared Error, pp. 653–653. Springer US, Boston, MA (2010). https://doi.org/10.1007/978-0-387-30164-8528, https://doi.org/10.1007/978-0-387-30164-8_528.

[13] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis (2016).

[14] Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks (2018).

Thank You