Message Passing Adaptive Resonance Theory for Online Active Semi-supervised Learning

Taehyeong Kim [1 2] **Injune Hwang** [1] **Hyundo Lee** [2] **Hyunseo Kim** [2]
Won-Seok Choi [2] **Joseph J. Lim** [1 3] **Byoung-Tak Zhang** [2]

Abstract

Active learning is widely used to reduce labeling effort and training time by repeatedly querying only the most beneficial samples from unlabeled data. In real-world problems where data cannot be stored indefinitely due to limited storage or privacy issues, the query selection and the model update should be performed as soon as a new data sample is observed. Various online active learning methods have been studied to deal with these challenges; however, there are difficulties in selecting representative query samples and updating the model efficiently without forgetting. In this study, we propose *Message Passing Adaptive Resonance Theory* (MPART) that learns the distribution and topology of input data online. Through message passing on the topological graph, MPART actively queries informative and representative samples, and continuously improves the classification performance using both labeled and unlabeled data. We evaluate our model in stream-based selective sampling scenarios with comparable query selection strategies, showing that MPART significantly outperforms competitive models.

1. Introduction

The recent success of deep learning in the field of visual object recognition and speech recognition is largely attributed to the massive amount of labeled data. However, most of the data samples in the real-world are not labeled, so it takes a lot of time and effort to label and use them for deep learning. In addition, in many countries, the collection and storage of medical data or data from personal robots are prohibited due to privacy concerns. This burden impedes the widespread use of deep learning in real-world applications such as medical AI, home appliances and robotics, and separates the best solutions from becoming the best real-world solutions.

From this point of view, active learning is a promising field of machine learning. The goal of active learning is to reduce the labeling cost by querying labels for only a small portion of the entire unlabeled dataset. The query selection is made according to which samples are most beneficial, i.e. both informative and representative (Settles, 2009). In most active learning algorithms, samples are selected from a large pool of data for querying (Settles, 2011; Gal et al., 2017; Sener & Savarese, 2017; Beluch et al., 2018). Once the queried data is labeled, it is accumulated in the training dataset and the entire model is trained again. This process is repeated, typically assuming that all the data can be stored and the model can access the data again at any time. In the aforementioned real-world problems, however, it is impossible to store a massive amount of data due to limited storage or privacy issues. In addition, it is highly inefficient to repeatedly train the entire model with large data and review all the unlabeled data for querying whenever a new label is acquired.

In contrast, the active learning paradigm in an online manner does not assume that data samples can be accessed repeatedly but instead, the input data is given as a continuous stream (Lughofer, 2017). This entails that uncertainty estimation of input samples and the decision of whether to query or not should be made online. The result of querying and training with one sample affects the uncertainty of subsequent input samples. Therefore, the model update should be done on-the-fly so that the new uncertainty distribution is estimated for the next query. In such scenarios, it is difficult to select a query sample that is representative as well as informative because all data samples cannot be reviewed at the same time. Moreover, since the labeling cost is expensive and the oracle is often unavailable, the number of queries may be limited (Hao et al., 2017; Zhang et al., 2018; Serrao & Spiliopoulou, 2018). Catastrophic forgetting is another issue to overcome when learning data online (Lee et al., 2017). This online active learning process, while more suitable for real-world problems, is therefore more challenging than offline active learning.

[1] AI Lab, CTO Division, LG Electronics, Seoul, Republic of Korea [2] Seoul National University, Seoul, Republic of Korea [3] University of Southern California, California, USA. Correspondence to: Byoung-Tak Zhang <btzhang@bi.snu.ac.kr>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

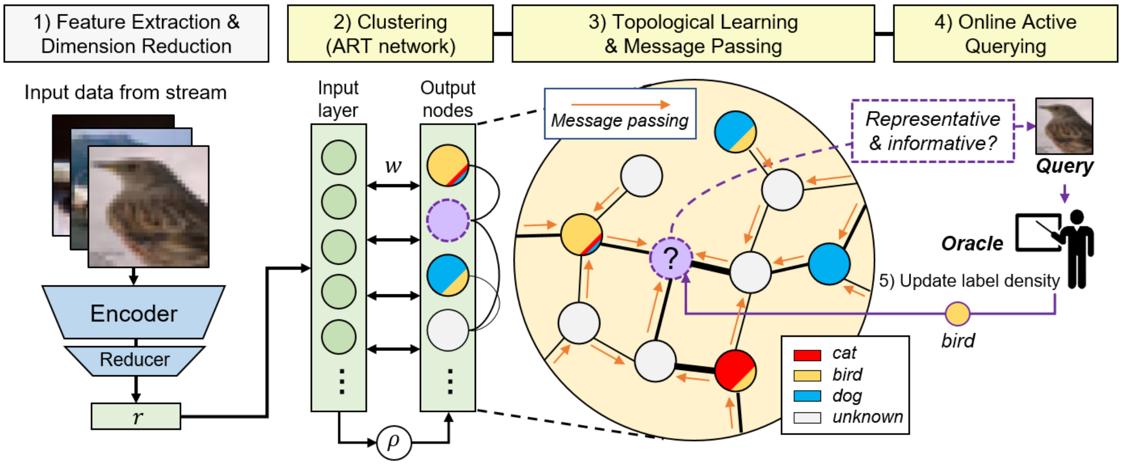

1) Feature Extraction & Dimension Reduction
2) Clustering (ART network)
3) Topological Learning & Message Passing
4) Online Active Querying
Input data from stream
Encoder
Reducer
r
Input layer
Output nodes
w
ρ
Message passing
Representative & informative?
Query
Oracle
?
5) Update label density
bird
cat
bird
dog
unknown

Figure 1. Overview of Message Passing Adaptive Resonance Theory (MPART). The feature is extracted from the input sample, and then MPART continuously constructs a weighted graph based on ART network by learning distributions and topology of the input data. It uses a message passing method to infer the class label and estimate the uncertainty of the input sample for querying. A sample is queried according to the query selection strategy, and the label density of the topological graph is updated using the collected labels.

In this regard, we propose Message Passing Adaptive Resonance Theory (MPART) that addresses these problems in online active learning. MPART learns the distribution of the input data without forgetting based on ART (Grossberg, 1987) which keeps the existing knowledge stable when learning new data. By utilizing a novel topology learning method on the learned distribution, MPART forms a topological graph using both labeled and unlabeled data. In order to exploit the limited amount of labels acquired, our model uses a message passing method which compensates for the lack of class information by propagating the information within the topological graph. Based on the learned topology, MPART can also select beneficial samples to query. The entire learning process, depicted in Figure 1, is performed in an online manner. We evaluate our model in the online active learning environments, showing that MPART significantly outperforms competitive models.

The main contributions can be summarized as follows.

- We propose a novel method that learns the distribution of input data by continuously constructing a weighted topological graph based on the ART network.
- We propose a message passing method for the graph to estimate the class labels in a semi-supervised manner and to select beneficial samples for querying.
- We design an online active learning task where the frequency of query is limited and the total number of classes is unknown. We validate the proposed model with various query selection strategies and datasets. The results of the experiment show that our model significantly outperforms the competitive models in online active learning environments.

2. Related Work

Topology Learning. Topology learning aims to learn the structural properties and topological relations underlying the input data. Additional knowledge such as class label or uncertainty can be obtained from the structural relationship using the topological information. SOM (Kohonen, 1990), a type of artificial neural network, learns the topological properties of the input data using neighborhood function and competitive Hebbian learning (CHL). This method is useful for dimension reduction of high-dimensional input data, but has a disadvantage in that it requires *a priori* selection of a subspace. GNG (Fritzke, 1995) and its derived models such as SOINN (Shen & Hasegawa, 2006) and E-SOINN (Shen et al., 2007) are incremental network models which can learn the topological relations of input data using the CHL. There are also variants of ART networks (see Section 3 for details) which integrate new knowledge into its existing knowledge so that what has already been learned is not forgotten by the new learning. Fuzzy ART-GL (Isawa et al., 2007) uses Group Learning that creates connections between similar categories. TopoART (Tscherepanow, 2010) combines incremental clustering with topology learning, which enables stable online clustering of non-stationary input data.

Semi-supervised Learning. Semi-supervised learning uses a small amount of labeled data with a large amount of unlabeled data to improve the performance of the model. (Iscen et al., 2019) and (Douze et al., 2018) use the label propagation method to assign labels for unlabeled data using the nearby labeled data. GAM (Stretcu et al., 2019) uses an agreement model that calculates the probability of two nodes sharing the same label on the graph. EGNN

(Kim et al., 2019) predicts the edge-labels of a graph to estimate the node-labels which are hard to be inferred directly in few-shot classification tasks. However, most of these methods are not suitable for online learning, because they need predefined topological information or repeated usage of whole training data. LPART (Kim et al., 2020) uses online label propagation on the ART network trained in a semi-supervised manner to overcome this issue, but the information conveyed between the nodes is limited due to insufficient topological information. MPART, on the other hand, addresses these problems by explicitly learning the topology of the input data.

Online Active Learning. The goal of online active learning is not only to reduce annotation costs, but also to continuously expand existing knowledge by exploring new information. OASIS (Goldberg et al., 2011) is a Bayesian model that combines online active learning and semi-supervised learning methods. SOAL (Hao et al., 2017) and OA3 (Zhang et al., 2018) utilize second-order information for online binary classification under limited query budgets. While these methods are successful in binary classification, they cannot handle the increasing diversity of classes due to the assumption of a fixed number of classes. Other methods (Loy et al., 2012; Weigl et al., 2016) employ consensus of models to better estimate uncertainty. However, using multiple models incurs large computational costs, making them less scalable for complex tasks. The most closely related work to ours is (Shen et al., 2011), enhancing the SOINN to enable online active semi-supervised learning. The proposed A-SOINN, however, selects the queries among representative values of clusters, which might be difficult for the oracle to recognize. Moreover, since this model stores only a single label in a cluster, it cannot handle the mixture of classes within the cluster and is susceptible to noise. Our model solves these problems by querying the input itself and using a distribution for labels in each node.

3. Background

Each related work shows a high potential to solve real-world problems in various domains, and motivates our idea of online active semi-supervised learning. In this section, we briefly introduce the ART networks and our motivations.

3.1. ART Networks

Adaptive Resonance Theory (ART), inspired by brain information processing mechanisms, is an unsupervised learning method for pattern recognition (Grossberg, 1987). The basic principle of ART is that object recognition is achieved by the interaction of ‘bottom-up’ sensory information and ‘top-down’ expectations. We refer to any neural network model based on ART as *ART network*. ART networks integrate new knowledge into the entire knowledge so that what has already been learned is not forgotten by new learning.

A basic ART network consists of two layers fully-connected to each other: an input layer and an output layer. As an input vector enters the input layer, it activates each node in the output layer according to the connection weight. Using the choice function with the activation values, the closest matching node is selected as a winner. Then, the fitness of the input to the winner node is calculated using the match function. If the fitness is greater than a vigilance parameter ρ, the input vector is integrated into the winner, updating the associated weight. Otherwise, a new node is created with a weight set equal to the input vector. With these mechanisms, ART networks can categorize incoming data online without forgetting. There are various ART networks such as Fuzzy ART (Carpenter et al., 1991), Gaussian ART (Williamson, 1996) and Hypersphere ART (Anagnostopoulos & Georgiopulos, 2000). We use Fuzzy ART as our backbone model, which incorporates fuzzy logic to enhance generalizability. Fuzzy ART will be described in Section 4.1 along with MPART’s topology learning method.

3.2. Motivations

Due to the nature of ART, the ART networks can efficiently learn new data by updating only a part of the model. In addition, it does not need to predefine the structure and size of the model, which allows flexible self-organization of irregular and complex data distributions. These properties play an important role in online learning, so we chose an ART network as a backbone model of MPART.

Moreover, in online active learning scenarios where data cannot be stored, it is difficult to select a representative sample for querying because all data samples cannot be reviewed simultaneously. We address this problem by learning the topology of input data. There is also an ART network called TopoART that can learn the topology. However, TopoART, in which only the winner and the runner-up node are connected, cannot effectively represent the relationships between nodes. In TopoART, the strength of the edge continuously increases as the data is learned, so a proper normalization method is required to use the edge information for message passing on the topological graph. Therefore, we propose a novel method to learn the topology and utilize it for message passing to select queries and predict labels.

4. Methods

MPART learns the distribution and topology of input data to construct a topological graph online. Then, class prediction and uncertainty estimation of the input data are performed using a message passing method on the graph. The estimated uncertainty is used to select and query useful samples. The entire process of MPART is described in Algorithm 1. In the following sections, we describe the topology learning, message passing, and active querying methods.

4.1. Topology Learning

Node Formation. We first extract representation vector $r_t \in [0,1]^{n_r}$ from the input data x_t using the pre-trained BYOL (Grill et al., 2020) and Parametric UMAP (Sainburg et al., 2020) (see Section 5.3 for details). As in Fuzzy ART, the representation vector r_t is complement coded to $I_t = [r_t, \vec{1} - r_t]$ in order to avoid proliferation of prototypes (Carpenter et al., 1991). With this, we can measure the similarity between the input and the category node j using the match function M_j and the choice function T_j as follows.

$$M_j(I_t) = \frac{\|I_t \wedge w_j\|_1}{\|I_t\|_1}, \qquad T_j(I_t) = \frac{\|I_t \wedge w_j\|_1}{\alpha + \|w_j\|_1} \tag{1}$$

Here, $\wedge$ is the element-wise minimum operator, $\|\cdot\|_1$ is the L1 norm, and $\alpha > 0$ is a choice parameter. A node j becomes *activated* by I_t if $M_j(I_t)$ is greater than or equal to a vigilance parameter $\rho \in (0, 1)$. When there are multiple activated nodes, or *co-activated nodes*, one with the highest $T_j(I_t)$ is chosen as a *winner* denoted by J_t. In case only one node is activated, we also refer to that node as the winner J_t. We update w_{J_t} with a learning rate $\beta \in (0, 1]$, and increase d_{J_t} by 1 as follows.

$$\begin{aligned} w_{J_t}^{new} &= \beta(I_t \wedge w_{J_t}^{old}) + (1-\beta) w_{J_t}^{old} \\ d_{J_t}^{new} &= d_{J_t}^{old} + 1 \end{aligned} \tag{2}$$

If none is activated, a new node J_t is created with an initial weight $w_{J_t} = I_t$ and a winning count $d_{J_t} = 1$.

Edge Formation. In addition to the node formation, edges between the co-activated nodes are developed. Unlike SOINN or TopoART in which only the winner and the runner-up node are connected, we connect the winner to every co-activated node to better represent more complex topology of multiple nodes. For an edge connecting nodes i and j, its count c_{ij} is set as the number of times i and j have been co-activated. The edge weight e_{ij} is defined as a ratio of c_{ij} to the sum of winning counts of incident nodes i and j as shown in Equation 3.

$$e_{ij} = \frac{c_{ij}}{d_i + d_j} \tag{3}$$

The weight e_{ij} well describes the correlation between i and j. If the two nodes have never been co-activated, e_{ij} is 0. As they get co-activated more frequently, e_{ij} grows larger; it reaches the maximum value of 1 if i and j have been co-activated whenever one of them was the winner. Since e_{ij} is bounded to $[0, 1]$, it can be used to adjust the degree of message passing without normalization. The edge formation does not interrupt the node formation, so the properties of the underlying ART network are also preserved.

Algorithm 1 The MPART algorithm

```
$V \leftarrow \{\}, C \leftarrow \{\}$
for $x_t$ in input data stream do
    $r_t \leftarrow \texttt{DimensionReduction}(x_t)$
    $I_t \leftarrow [r_t, \overrightarrow{1} - r_t]$
    $A \leftarrow \{\}$
    for $j$ in $1, \ldots, |V|$ do
        $M_j \leftarrow \|I_t \wedge w_j\|_1 \ / \ \|I_t\|_1$
        $T_j \leftarrow \|I_t \wedge w_j\|_1 \ / \ (\alpha + \|w_j\|_1)$
        if $M_j \geq \rho$ then
            $A \leftarrow A \cup \{j\}$
        end if
    end for
    if $A$ is empty then
        $J_t \leftarrow |V| + 1, \quad V \leftarrow V \cup \{J_t\}$
        $c_{J_t v} \leftarrow 0,\ c_{v J_t} \leftarrow 0 \quad \forall v \in V - \{J_t\}$
        $q_{J_t}(y) \leftarrow 0 \quad \forall y \in C$
        $w_{J_t} \leftarrow I_t, \quad d_J \leftarrow 1$
    else
        $J_t \leftarrow \arg\max_{j \in A}(T_j)$
        $c_{J_t v} \leftarrow c_{J_t v} + 1,\ c_{v J_t} \leftarrow c_{v J_t} + 1 \quad \forall v \in A - \{J_t\}$
        $w_{J_t} \leftarrow \beta(I_t \wedge w_{J_t}) + (1-\beta) w_{J_t}$
        $d_{J_t} \leftarrow d_{J_t} + 1$
    end if
    $q_{J_t}^{(L)}, d_{J_t}^{(L)} \leftarrow \texttt{MessagePassing}(J_t, c, d, q)$
    $p_t, \hat{y} \leftarrow \texttt{NodeClassification}(q_{J_t}^{(L)})$
    $s_t \leftarrow \texttt{UncertaintyEstimation}(p_t, q_{J_t}^{(L)}, d_{J_t}^{(L)})$
    if $s_t$ satisfies query condition then
        $y_t \leftarrow \texttt{QueryLabel}(x_t)$
        $q_{J_t}(y_t) \leftarrow q_{J_t}(y_t) + 1$
        $C \leftarrow C \cup \{y_t\}$
    end if
end for
```

4.2. Message Passing

Nodes formed in MPART possess additional information vectors such as class probabilities or uncertainty measures for better inference and querying. To compensate for the lack of information, we use message passing to aggregate the information within the topological graph. Before describing the procedure, we first define a notion of *neighbors*. Given nodes i and j, if the shortest path from i to j exists and its length is l, then we say j is an l-hop neighbor of i. Any node is a 0-hop neighbor to itself. A set of all l-hop neighbors of i is denoted as $\mathcal{N}_i^{(l)}$; a set of direct neighbors $\mathcal{N}_i^{(1)}$ will also be written as $\mathcal{N}_i$. We further denote an l-hop neighborhood $\mathcal{N}_i^{(0:l)}$ as a union of $\mathcal{N}_i^{(0)}, \ldots, \mathcal{N}_i^{(l)}$.

To illustrate message passing, let X denote an information vector (or scalar) of interest such as winning count d or label density q (will be described later). A single-layer message passing on the target node i updates X_i by adding X_j

of every neighboring node $j \in \mathcal{N}_i$, weighted by e_{ij} and discounted by a propagation rate $\delta \in [0, 1]$. Multi-layer message passing is achieved by performing the single-layer message passing recursively, aggregating the information from broader neighboring area. The recursive procedure for L-layer message passing targeted at the winner J_t can be formalized as Equation 4, where $X_i^{(l)}$ denotes the l-layer aggregated information and $X_i^{(0)}$ is set as X_i.

$$X_i^{(l)} = X_i^{(l-1)} + \delta \sum_{j \in \mathcal{N}_i} e_{ij} X_j^{(l-1)}, \ \forall i \in \mathcal{N}_{J_t}^{(0:L-l)} \tag{4}$$

We finally obtain the L-layer aggregated information $X_{J_t}^{(L)}$ by applying this update for l from 1 to L. Note that we only have to consider nodes in $\mathcal{N}_{J_t}^{(0:L-l)}$ when performing the l-th layer update since the information outside this set will not be propagated to J_t within the remaining $(L-l)$ updates. This reduces the computational cost compared to when reviewing all the nodes in the graph.

Node Classification. MPART infers the class of input x_t by aggregating the class information near the winner J_t. The class information q_i of a node i, called the label density, is a distribution over a set of known class labels C and indicates how probable the node belongs to each class. We use distributions rather than single values to cope with situations where one node represents samples of different classes.

Before receiving any label, each q is set to an empty distribution, i.e. an empty vector. As a new label is received and added to the label set C, every q is expanded by one dimension (zero-valued) which will be responsible for that label. For new nodes, label densities are set to zero vectors of dimension $|C|$. When a label y_t is provided with an input x_t, the corresponding density value in the winner J_t, i.e. $q_{J_t}(y_t)$, increases by 1. This increment also applies if the label was obtained via active querying. Since labels are provided rarely, we perform L-layer message passing for q to alleviate the deficiency. The aggregated label density $q_{J_t}^{(L)}$ contains class information of neighboring nodes, so it is representative and robust to noise. We obtain the class probability distribution p_t by normalizing it as Equation 5.

$$p_t(y) = \frac{q_{J_t}^{(L)}(y)}{\sum_{y' \in C} q_{J_t}^{(L)}(y')} \tag{5}$$

If $q_{J_t}^{(L)}$ is a zero vector, we set p_t to a uniform distribution. The label of the input is inferred as one with the highest probability. This distribution is also used to estimate the label uncertainty of a node, as described in the next section.

4.3. Active Querying

We combine the uncertainty-based sampling with density-weighted method for query selection. The uncertainty is measured per node using the label density and weighted by distribution density of data to produce the selection score. This score can be used with various selection strategies, including those suggested in Section 5.2.

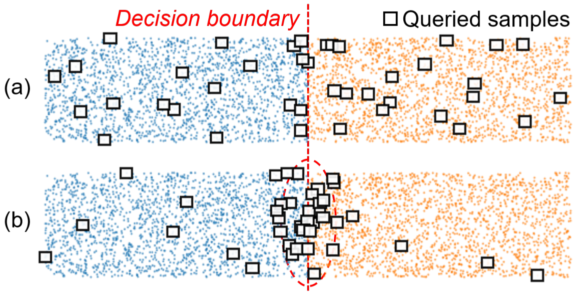


Figure 2. Example queries (black boxes) selected online according to the query selection score, for a simple binary classification of 2,000 samples each (blue and orange). (a) The first 40 queries are distributed relatively evenly, while (b) the latter 40 queries are concentrated around the decision boundary.

Uncertainty Estimation. We use two kinds of uncertainty measures as in (Kim et al., 2020). The first uncertainty u_e indicates the insufficiency of class information accumulated in each node. It is determined by the sum of label density values over all known classes. As more labels are input or more times of message passing are performed, u_e decreases as the sum of densities increases. For an input sample x_t, u_e is defined as that of the winner J_t as in Equation 6, with a sensitivity constant $k_e > 0$ and $\tanh$ for clamping.

$$u_e = 1 - \tanh\left(k_e \sum_{y \in C} q_{J_t}^{(L)}(y)\right) \tag{6}$$

The second uncertainty u_a measures the class impurity, i.e. how many classes are mixed within a node. We can use the normalized entropy of class probability, while setting u_a to 0 when no more than one label is observed.

$$u_a = \begin{cases} \frac{-\sum_{y \in C} p_t(y) \log p_t(y)}{\log(|C|)}, & \text{if } |C| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The two quantities, u_e and u_a, can be viewed as epistemic and aleatoric uncertainties respectively, in that u_e results from insufficient observation while u_a arises due to ambiguity in the input data. These uncertainties are complementary when used for query selection, since u_e plays an important role in the early stage of learning while u_a is more crucial after enough number of labels have been acquired. We combine them with a weight $\tau \in [0, 1]$ to get a query selection score u_t as shown in Equation 8. Figure 2 illustrates that the distribution of queries selected using u_t tends to be sparse at first, but concentrated near the decision boundary as the learning progresses.

$$u_t = \tau u_e + (1-\tau) u_a \tag{8}$$

Density-Weighted Method. Beneficial samples should be representative as well as informative (Huang et al., 2014). Both uncertainties u_a and u_e are considered representative because they are derived from label density $q_{J_t}^{(L)}$ aggregated from surrounding nodes. Additionally, we aggregate the winning count d_i to select representative samples, which represents the distribution density of the input data. Finally, the density-weighted query selection score s_t using data density $d_{J_t}^{(L)}$ is defined as in Equation 9.

$$\begin{gathered} d_i^{(l)} = d_i^{(l-1)} + \delta \sum_{j \in \mathcal{N}_i} e_{ij} d_j^{(l-1)}, \; \forall i \in \mathcal{N}_{J_t}^{(0:L-l)} \\ s_t = \tanh\left(k_d \cdot d_{J_t}^{(L)}\right) \cdot u_t \end{gathered} \tag{9}$$

Here, k_d is a positive constant for sensitivity. If the density-weighted query selection score s_t of the input x_t satisfies the condition according to the query selection strategy (see Section 5.2 for details), the model immediately queries the oracle to get a label y_t. An example of training results using the CIFAR-10 dataset is visualized in Figure 3.

5. Experiments

We investigated the effectiveness of MPART in online active learning scenarios. To do this, we designed a task described in Section 5.1 and evaluated the performance of the model with various settings. We also compared the performance to that of two competitive models: LPART (Kim et al., 2020) and A-SOINN (Shen et al., 2011). For statistical significance, we repeated each experiment 30 times.

The propagation rate δ for message passing was set to 0.1, and the parameters k_e, τ and k_d used for the score calculation were set to 1.0, 0.7 and 0.01, respectively. For other parameter settings, please refer to the Appendix.

5.1. Task

We set up an online active learning task to imitate real-world scenarios where only a few number of annotations are possible and the data cannot be stored. There is a previous study that has proposed this kind of task for stream-based selective sampling (Attenberg & Provost, 2011). In this previous work, the query budget is set as the maximum number of queries allowed in a fixed input period instead of the total number of queries throughout the training. Similarly, we constrained the model to query for certain number of times within each period, assuming that the oracle can annotate only a few samples in a period.

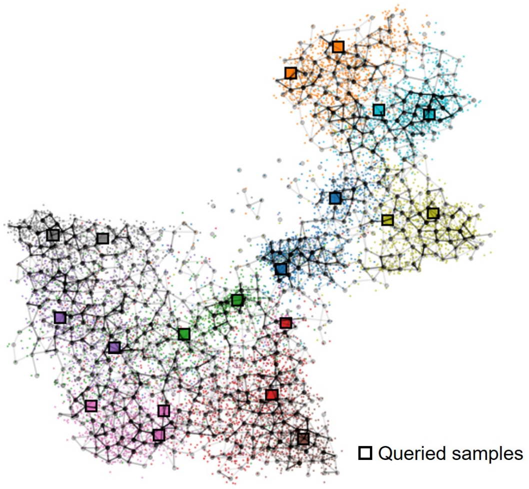


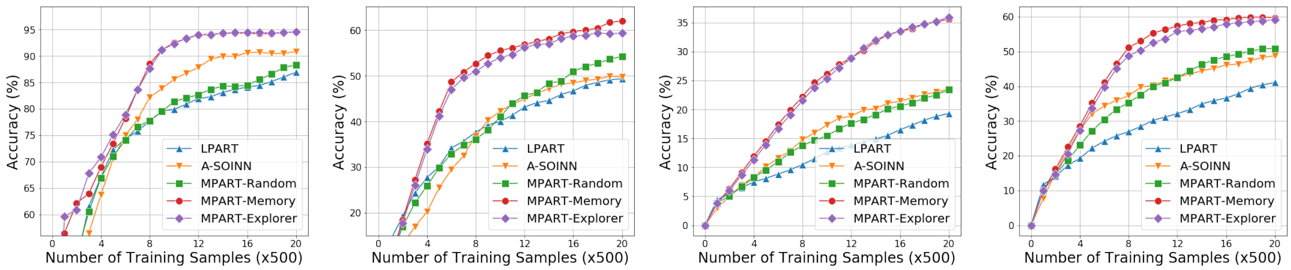
Figure 3. The visualization of result on the CIFAR-10 dataset for 1/500 query frequency and $L = 3$ using 'Explorer' strategy. The color of the dots represent the label of input data, and the intensity of the topological graph represents the density of nodes and edges. The distribution of the queried samples is spread evenly.

The basic goal of this task is to perform multi-class classification, where the number of classes is unknown in advance. The input samples are provided sequentially as a stream, i.e. one after another, and cannot be stored or reused. The whole training dataset is initially unlabeled and the model can inquire the oracle about the labels of a small number of inputs. Specifically, the number of queries is limited to a fixed budget B within a period of W consecutive inputs. We denote such constraint as a query frequency B/W. The final classification performance is evaluated on the hold-out test dataset.

For experiments, we used four kinds of datasets with different distributions: Mouse retina transcriptomes (Macosko et al., 2015; Poličar et al., 2019), Fashion MNIST (Xiao et al., 2017), EMNIST Letters (Cohen et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009). The datasets consist of 12, 10, 26, and 10 classes, respectively. The budget B was set to 1 or 2, and the period W was set to 100, 500, 1,000 or 2,000. For model training, we only used 10,000 randomly sampled data from the training split per trial. This is to push the situation to an extreme; with 1/500 or less query frequency, the model will not be able to receive labels for some classes unless it successfully selects and queries all the samples from different classes.

5.2. Query Selection Strategy

Random. A random query is selected from a sequence of inputs. This is a baseline and works as an ablated version of active selection strategies. It is equivalent to randomly providing labels for some inputs, so MPART learns in an online semi-supervised manner when using this strategy.



(a) Mouse retina transcriptomes (b) Fashion MNIST (c) EMNIST Letters (d) CIFAR-10

Figure 4. Classification accuracy according to the number of training samples. The number of layer $L = 3$ and 1 / 500 query frequency were used for MPART. The order of input samples was shuffled regardless of the class. Note that, for the EMNIST Letters dataset, the models can collect up to 20 class labels out of a total of 26 class labels after all training is complete.

Memory. This can only be used at $1/W$ query frequency, assuming the model has a memory that can store at most one sample. During each query period, one sample with the maximum density-weighted query selection score s_t is stored and queried at the end of the period. By using this strategy, the model can select the most beneficial sample.

Explorer. 'Explorer' assumes the most stringent situation where the learning agent cannot store any input sample. In this situation, the learning agent selects B samples online for each query period W, and cannot change the sample once selected. Therefore, the chance of selecting an beneficial sample decreases as the exploration gets longer because of the fixed query selection period. In this strategy, the uncertainty distribution of the data explored so far is continuously estimated to solve the exploration-exploitation dilemma (Berger-Tal et al., 2014). The random variable of the uncertainty distribution $\mathcal{S}_n$ is assumed to follow the normal distribution $\mathcal{S}_n \sim N(\mu_n, \sigma_n^2)$, where n is the number of accumulated samples since the previous query. The parameters μ_n and σ_n^2 are calculated as the sample mean and variance of s_t as in Equation 10.

$$\begin{aligned} \mu_n &= (1 - n^{-1})\mu_{n-1} + n^{-1} s_t \\ \sigma_n^2 &= (1 - n^{-1})\sigma_{n-1}^2 + n^{-1}(\mu_n - s_t)^2 \end{aligned} \tag{10}$$

An input sample is considered beneficial if its density-based query selection score is likely to be greater than that of most unseen samples. To be more concrete, we select the t-th sample if Equation 11 is satisfied. Here, $Binom$ denotes a probability mass function of a binomial distribution, $F_{\mathcal{S}_n}(\cdot)$ is the cumulative density function of $\mathcal{S}_n$, t_p is the time from the start of the current period, and b is the number of remaining query budget in the current period. If $\sigma_n = 0$, we set $F_{\mathcal{S}_n}(s_t)$ to 0.5.

$$\sum_{m=0}^{b-1} Binom\left(m; W - t_p, 1 - F_{\mathcal{S}_n}(s_t)\right) > 0.5 \tag{11}$$

When a query is selected, b is reduced by 1, and both μ_0 and σ_0 are reset to 0. If b is 0, no query is performed until the end of the period, and b is reset to B when a new period starts. With the 'Explorer' strategy, the model can efficiently and effectively choose the useful samples by predicting the beneficialness of the unseen ones.

5.3. Feature Extraction & Dimension Reduction

We projected the input data into 4-dimensional embedding space using the pre-trained Parametric UMAP (Sainburg et al., 2020) for each dataset. The embedding space was mapped to $[0, 1]^4$ using min-max normalization on training data. Prior to projection, we extracted 2048-dimensional feature vectors from CIFAR-10 using the BYOL (Grill et al., 2020) pre-trained with the ImageNet dataset (Deng et al., 2009). Parametric UMAP is suitable for use in semi-supervised learning by capturing structure in unlabeled data to provide the embeddings for unseen data online which can be clustered properly. We trained Parametric UMAP using 30% of the training data for each dataset, and the rest was used for MPART training. Note that no labeled data was used until this step.

6. Results and Discussion

6.1. Comparison with Competitive Models

The performance of the competitive models LPART (Kim et al., 2020) and A-SOINN (Shen et al., 2011) was compared to that of MPART with various query selection strategy, shown in Table 1 and Figure 4. The fully supervised settings including multi-layer perceptron (MLP) model were used as references, which was trained using all labeled data for each dataset. The MLP is consisted of 3 layers with 128 neurons per layer and we reported the highest test accuracy while training up to 200 epochs. The same ART-related parameter values were used in MPART and LPART, while A-SOINN parameters were adjusted to achieve its best performance

Table 1. Comparison of classification accuracy (mean ± std) between our model (MPART) and the competitive models according to the query selection strategy. We also report classification accuracy of MPART depending on whether density-weighted query selection score (DS) is applied or not. The number of layers $L = 3$ for message passing of MPART was used. (unit : %)

Query Selection Frequency	Model	Mouse retina transcriptomes	Fashion MNIST	EMNIST Letters	CIFAR-10
Fully Supervised	MLP	94.0±1.7	71.9±1.6	50.9±2.0	**75.1**±1.7
	LPART	**97.2**±0.1	73.2±0.3	60.2±0.4	74.2±0.3
	A-SOINN	94.2±1.5	66.5±1.5	47.6±1.6	65.5±1.7
	MPART	**97.2**±0.1	**73.3**±0.2	**60.7**±0.3	74.3±0.3
1 / 1000	LPART	82.5±7.5	41.1±6.4	14.7±2.6	23.8±5.3
	A-SOINN	88.1±13.0	45.2±7.0	16.9±4.1	37.7±6.8
	MPART-Random	81.0±7.0	42.8±8.5	17.1±3.1	40.9±8.1
	MPART-Memory	**93.3**±3.3	**53.6**±4.8	**26.1**±2.8	**55.5**±4.6
	MPART-Explorer	91.8±4.0	53.5±5.4	25.4±2.4	55.4±3.7
	MPART-Memory (w/o DS)	90.5±5.6	51.4±5.6	19.8±3.5	44.9±7.0
	MPART-Explorer (w/o DS)	87.2±8.1	49.9±6.9	18.2±3.4	44.1±6.7
1 / 500	LPART	89.0±6.3	54.6±5.9	21.0±3.4	34.0±4.7
	A-SOINN	91.0±5.4	50.3±6.0	25.6±4.8	44.2±7.0
	MPART-Random	88.3±4.4	54.3±7.6	23.4±3.8	50.9±5.2
	MPART-Memory	94.5±1.1	**62.0**±4.4	35.6±2.5	**59.7**±3.4
	MPART-Explorer	**94.6**±0.9	59.4±5.2	**36.0**±2.9	59.2±3.3
	MPART-Memory (w/o DS)	91.9±3.4	51.7±5.8	28.7±3.3	52.2±5.4
	MPART-Explorer (w/o DS)	92.1±3.4	53.9±4.8	29.2±3.8	51.6±6.6
1 / 100	LPART	94.6±1.5	64.4±2.0	38.0±2.3	62.0±2.9
	A-SOINN	91.5±7.0	55.7±6.8	29.9±4.9	50.9±7.9
	MPART-Random	94.8±1.2	66.8±2.0	43.6±3.2	63.4±2.6
	MPART-Memory	**95.9**±0.5	**67.7**±1.6	**47.9**±1.6	**67.4**±1.5
	MPART-Explorer	**95.9**±0.9	**67.7**±1.3	47.5±1.5	66.8±1.4
	MPART-Memory (w/o DS)	95.0±1.6	63.2±2.9	39.7±2.5	60.6±2.7
	MPART-Explorer (w/o DS)	94.4±1.5	65.7±2.1	41.9±2.0	62.3±2.4

for fair comparison. A-SOINN needs to query the prototype of the most dense node, which is a weighted sum of encoded representations and does not correspond to any input samples. Therefore, in A-SOINN, the prototype of the node was not directly queried, but the most recent input sample that activates the node was queried. In all experimental settings except with MLP, the MPART showed the highest accuracy.

Comparison with LPART. MPART with 'Random' strategy showed higher performance than LPART, which performs online semi-supervised learning. This indicates that although both models are based on the same backbone model, Fuzzy ART, MPART's message passing method is more effective than LPART's label propagation method. MPART's massage passing differs from LPART's label propagation in several aspects. The critical difference is that LPART does not sufficiently learn and use topology information of input data. LPART permanently updates the label density q when label propagation is performed, whereas MPART updates q^0 using only true labels and infers the class through message passing. The MPART's message passing method with multi-layers prevents information propagation of one node from permanently affecting other nodes, and when a new label is acquired, the information can be immediately reflected in the inference of all nodes. In addition, MPART queries representative samples by estimating the distribution density of input data using the topology learned online, whereas in the case of LPART, it is difficult to select a representative sample because it cannot effectively estimate the distribution of the input data.

Comparison with A-SOINN. A-SOINN employs an explicit removal of edges and nodes to reduce the noise and maintain a reasonable model size. The removal procedure heavily depends on the deletion period λ, which should ideally be determined according to the input statistics. Since such information is not known *a priori*, a wrong choice of λ severely degrades the model performance. Also, each node in A-SOINN holds only one label which is fixed since the first assignment either with actual label or by propagation from other nodes. Therefore, the order of labeling highly influences the inference results, especially at deci-

Table 2. Classification accuracy (mean ± std) according to MPART's query frequency and number of layers L for message passing. The 'Explorer' strategy was used for query selection of MPART. (unit : %)

Query Selection Frequency	Number of Layers	Mouse retina transcriptomes	Fashion MNIST	EMNIST Letters	CIFAR-10
2 / 2000	$L = 5$	90.0±7.1	52.6±5.7	25.6±2.8	52.9±4.6
	$L = 3$	**92.3**±4.0	**56.5**±3.9	**26.5**±2.3	**53.9**±4.4
	$L = 1$	66.9±4.7	44.0±5.2	14.3±1.6	27.3±2.2
	w/o MP	31.1±4.9	15.2±0.9	4.9±0.3	11.3±0.3
4 / 2000	$L = 5$	**94.1**±2.2	**60.1**±4.6	33.4±3.2	58.1±4.0
	$L = 3$	93.1±3.5	60.0±4.4	**36.0**±2.8	**59.5**±3.8
	$L = 1$	86.8±2.6	59.2±3.9	21.4±1.2	38.3±2.0
	w/o MP	31.0±2.3	19.6±1.6	6.0±0.2	12.5±0.5
20 / 2000	$L = 5$	**96.1**±0.5	**67.6**±1.5	47.6±1.6	**67.4**±1.6
	$L = 3$	95.8±0.7	67.3±1.5	**48.0**±1.7	67.2±1.3
	$L = 1$	95.0±1.1	66.6±1.4	40.3±1.5	59.1±1.1
	w/o MP	57.8±1.5	43.6±2.2	12.2±0.6	20.5±0.6

sion boundaries. This might explain why A-SOINN even performs worse than LPART in fully-supervised setting.

Computational Cost. MPART requires more computation than LPART and A-SOINN, but the computational cost required to update the model is very low. We measured the run-time of our Python implementation on a 3.8 GHz CPU machine. The time taken to train and infer 10,000 of CIFAR-10 data samples was 11.2 seconds on average, which took about 1.12 ms per sample. For the CIFAR-10 dataset, the average numbers of nodes are 1260, 1922, and 2442 when trained with 15k, 30k and 45k samples respectively. Please refer to the Appendix for detailed statistical analysis and additional experimental results.

6.2. Ablation Study

Table 1 summarizes the results of performance evaluation of our model on four datasets according to query selection frequencies and strategies. In the w/o DS setting, the query selection score u_t of Equation 8 was used instead of s_t of Equation 9. In all experimental settings, the model that applies density-weighted query selection score showed significantly higher performance than the one without (w/o DS). When comparing the performance with respect to the query selection strategy, the accuracy of the 'Random' strategy was generally low, and the 'Memory' strategy and the 'Explorer' strategy showed almost similar performance. This is because the 'Explorer' strategy properly estimates the uncertainty distribution of input data and efficiently selects the representative samples based on the remaining query opportunities. In addition, the lower the query frequency, the greater the difference in performance depending on the query selection strategy, indicating that the strategy can have a significant impact on performance in situations where labeled data is extremely scarce. Table 2 shows the performance evaluation of MPART according to the number of layers L for message passing. As the number of layers increased, the classification performance generally increased, but the results of using 3 and 5 layers were almost the same. On the other hand, when message passing is not applied (w/o MP), there is a significant performance penalty.

7. Conclusions

We propose Message Passing Adaptive Resonance Theory (MPART) for online active semi-supervised learning. MPART learns the distribution and the topology of the input data online, infers the class of unlabeled data, and selects the informative and representative samples through message passing between nodes on the topological graph. By evaluating our method on datasets including EMNIST Letters and CIFAR-10, we show that it outperforms the competitive models. In an online learning environment where data storage is limited and data labeling is expensive, waste of useful samples is inevitable. The proposed model fully utilizes the underlying structure of input data so that it can minimize the waste. This approach also reduces the need to create large datasets in advance in order to apply machine learning to various industries. We believe MPART offers new opportunities for machine learning techniques to be widely used in real-world applications.

Acknowledgements

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (2019-0-01371-BabyMind/35%, 2015-0-00310-SW.StarLab/5%, 2017-0-01772-VTT/5%, 2018-0-00622-RMI/5%) grant funded by the Korean government.

References

Anagnostopoulos, G. C. and Georgiopulos, M. Hypersphere art and artmap for unsupervised and supervised, incremental learning. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 6, pp. 59–64. IEEE, 2000.

Attenberg, J. and Provost, F. Online active inference and learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 186–194, 2011.

Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9368–9377, 2018.

Berger-Tal, O., Nathan, J., Meron, E., and Saltz, D. The exploration-exploitation dilemma: a multidisciplinary framework. *PloS one*, 9(4):e95693, 2014.

Carpenter, G. A., Grossberg, S., and Rosen, D. B. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural networks*, 4(6): 759–771, 1991.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Douze, M., Szlam, A., Hariharan, B., and Jégou, H. Low-shot learning with large-scale diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3349–3358, 2018.

Fritzke, B. A growing neural gas network learns topologies. In *Advances in neural information processing systems*, pp. 625–632, 1995.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. *arXiv preprint arXiv:1703.02910*, 2017.

Goldberg, A., Zhu, X., Furger, A., and Xu, J.-M. OASIS: Online active semi-supervised learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 362–367, 2011.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1): 23–63, 1987.

Hao, S., Lu, J., Zhao, P., Zhang, C., Hoi, S. C., and Miao, C. Second-order online active learning and its applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1338–1351, 2017.

Huang, S.-J., Jin, R., and Zhou, Z.-H. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949, 2014.

Isawa, H., Tomita, M., Matsushita, H., and Nishio, Y. Fuzzy adaptive resonance theory with group learning and its applications. In *Proc. of International Symposium on Nonlinear Theory and its Applications*, pp. 292–295, 2007.

Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.

Kim, J., Kim, T., Kim, S., and Yoo, C. D. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2019.

Kim, T., Hwang, I., Kang, G.-C., Choi, W.-S., Kim, H., and Zhang, B.-T. Label propagation adaptive resonance theory for semi-supervised continuous learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4012–4016. IEEE, 2020.

Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Lee, S.-W., Kim, J.-H., Jun, J., Ha, J.-W., and Zhang, B.-T. Overcoming catastrophic forgetting by incremental moment matching. *arXiv preprint arXiv:1703.08475*, 2017.

Loy, C. C., Hospedales, T. M., Xiang, T., and Gong, S. Stream-based joint exploration-exploitation active learning. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1560–1567. IEEE, 2012.

Lughofer, E. On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences*, 415:356–376, 2017.

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

Poličar, P. G., Stražar, M., and Zupan, B. opentsne: a modular python library for t-sne dimensionality reduction and embedding. *BioRxiv*, pp. 731877, 2019.

Sainburg, T., McInnes, L., and Gentner, T. Q. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning. *arXiv preprint arXiv:2009.12981*, 2020.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

Serrao, E. and Spiliopoulou, M. Active stream learning with an oracle of unknown availability for sentiment prediction. In *IAL@ PKDD/ECML*, pp. 36–47, 2018.

Settles, B. Active learning literature survey. 2009.

Settles, B. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pp. 1–18, 2011.

Shen, F. and Hasegawa, O. An incremental network for on-line unsupervised classification and topology learning. *Neural networks*, 19(1):90–106, 2006.

Shen, F., Ogura, T., and Hasegawa, O. An enhanced self-organizing incremental neural network for online unsupervised learning. *Neural Networks*, 20(8):893–903, 2007.

Shen, F., Yu, H., Sakurai, K., and Hasegawa, O. An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. *Neural Computing and Applications*, 20(7):1061–1074, 2011.

Stretcu, O., Viswanathan, K., Movshovitz-Attias, D., Platanios, E., Ravi, S., and Tomkins, A. Graph agreement models for semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 8710–8720, 2019.

Tscherepanow, M. TopoART: A topology learning hierarchical art network. In *International Conference on Artificial Neural Networks*, pp. 157–167. Springer, 2010.

Weigl, E., Heidl, W., Lughofer, E., Radauer, T., and Eitzinger, C. On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications*, 27 (1):103–127, 2016.

Williamson, J. R. Gaussian artmap: A neural network for fast incremental learning of noisy multidimensional maps. *Neural networks*, 9(5):881–897, 1996.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhang, Y., Zhao, P., Cao, J., Ma, W., Huang, J., Wu, Q., and Tan, M. Online adaptive asymmetric active learning for budgeted imbalanced data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2768–2777, 2018.