

MALADY: Multiclass Active Learning with Auction Dynamics on Graphs

Gokul Bhusal, Kevin Miller, and Ekaterina Merkurjev

Abstract—Active learning enhances the performance of machine learning methods, particularly in low-label rate scenarios, by judiciously selecting a limited number of unlabeled data points for labeling, with the goal of improving the performance of an underlying classifier. In this work, we introduce the Multiclass Active Learning with Auction Dynamics on Graphs (MALADY) algorithm, which leverages an auction dynamics technique on similarity graphs for efficient active learning. In particular, the proposed algorithm incorporates an active learning loop using as its underlying semi-supervised procedure an efficient and effective similarity graph-based auction method consisting of upper and lower bound auctions that integrate class size constraints. In addition, we introduce a novel active learning acquisition function that incorporates the dual variable of the auction algorithm to measure the uncertainty in the classifier to prioritize queries near the decision boundaries between different classes. Overall, the proposed method can efficiently obtain accurate results using extremely small labeled sets containing just a few elements per class; this is crucial since labeled data is scarce for many applications. Moreover, the proposed technique can incorporate class size information, which improves accuracy even further. Lastly, using experiments on classification tasks and various data sets, we evaluate the performance of our proposed method and show that it exceeds that of comparison algorithms.

Impact Statement—Sufficient amounts of labeled data are an essential element for good performance in machine learning tasks. However, in many practical applications, obtaining sufficient amounts of labeled data can be very expensive and time-consuming. Graph-based semi-supervised methods have shown great promise in low-label rate machine learning. However, one can further enhance the performance of semi-supervised learning by integrating an active learning framework. In this work, we propose an active learning setting that incorporates an auction dynamics technique for the semi-supervised learning problem in a similarity graph-based framework. The experimental results demonstrate a significant improvement in the semi-supervised algorithm performance when integrating our active learning method. A wide range of applications, including healthcare, natural language processing, and remote sensing, can greatly benefit from this work in data-efficient machine learning.

Index Terms—Active Learning, Auction Dynamics for Semi-Supervised Learning, Uncertainty Sampling

Received 15 October 2024; revised 8 April 2025; accepted 26 May 2025. This work was supported in part by the NSF DMS-2052983 grant. This article was recommended for publication by Editor-in-Chief Prof. Jun Wang. (Corresponding author: Ekaterina Merkurjev.)

Gokul Bhusal and Ekaterina Merkurjev is with the Department of Mathematics at Michigan State University, MI, 48824, USA (e-mail: gbhusal@msu.edu; merkurje@msu.edu). Ekaterina Merkurjev is also with the Department of CMSE at Michigan State University.

Kevin Miller is with the Department of Mathematics at Brigham Young University, Provo, UT, USA, 84602, USA. (e-mail: ksmiller@mathematics.byu.edu).

I. INTRODUCTION

THE choice of training points can have a significant impact on the performance of a machine learning (ML) model, particularly in semi-supervised learning (SSL) scenarios where the labeled set is small. Active learning (AL) is a sub-field of ML that improves the performance of underlying ML methods by carefully selecting unlabeled points to be labeled via the use of a human in the loop or a domain expert. Most AL methods alternate between (1) training a model using current labeled information and (2) selecting query points from an unlabeled set using an acquisition function that quantifies the utility of each point in the unlabeled set. This iterative process of training a classifier and labeling chosen query points is referred to as the AL process, whose flowchart is shown in Figure 1. Given a dataset $\mathcal{X} = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d$, we define the *labeled set* $\mathcal{L} \subset \mathcal{X}$ as the set of labeled points with the corresponding labels $y_i = y(x_i) \in \{1, 2, \dots, C\}$; then, by *labeled data*, we refer to both the labeled set \mathcal{L} of inputs along with their corresponding labels $\{y_i\}_{x_i \in \mathcal{L}}$. The *unlabeled set* then is the set of unlabeled inputs, $\mathcal{U} = \mathcal{X} \setminus \mathcal{L}$. A *query set* $\mathcal{Q} \subset \mathcal{U}$ is the set of unlabeled points that have been chosen to be labeled and added to the labeled set for the next iteration of the AL process. While the sets \mathcal{L} and \mathcal{U} change throughout the active learning process as query points are labeled, we simplify our notation of these sets to not explicitly denote the iteration; rather, \mathcal{L} , \mathcal{U} , and \mathcal{Q} will respectively denote the labeled, unlabeled, and query sets at the *current* iteration of the AL process.

Given the current \mathcal{L} and \mathcal{U} , the main challenge in AL is to design an *acquisition function* $\mathcal{A} : \mathcal{U} \rightarrow \mathbb{R}$ that quantifies the benefit of obtaining the label for each currently unlabeled point. The acquisition function values $\mathcal{A}(x)$ for each $x \in \mathcal{U}$ then allow one to prioritize which currently unlabeled data points are chosen to be in the query set \mathcal{Q} to in turn be labeled by a human in the loop or a domain expert. When $|\mathcal{Q}| = 1$, the process is referred to as *sequential* AL while $|\mathcal{Q}| > 1$ corresponds to *batch* AL. In this work, we focus on sequential AL, and Figure 1 reflects this specialization. While there are various ways to select $\mathcal{Q} \subset \mathcal{U}$ from the set of acquisition function values $\{\mathcal{A}(x)\}_{x \in \mathcal{U}}$, it is most common in sequential AL to use $\mathcal{Q} = \{x^*\}$, where $x^* = \arg \max_{x \in \mathcal{U}} \mathcal{A}(x)$ is the maximizer of the acquisition function.

Uncertainty sampling-based acquisition functions [1–3] are popular for practical active learning and are fre-

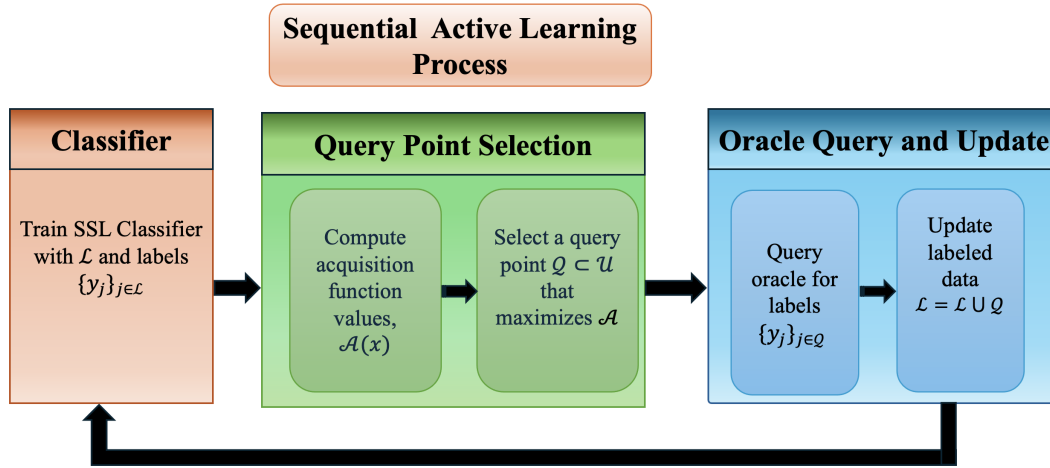


Fig. 1: The flowchart of the sequential active learning process. The process starts by training the underlying classifier using the initial labeled set \mathcal{L} . A query point $Q \in \mathcal{U}$ is then selected based on the current acquisition function values. This query point is also labeled according to an oracle (human in the loop or a domain expert), and subsequently, the point is added to the current labeled data \mathcal{L} . This process continues until the desired number of points is reached in the labeled set \mathcal{L} .

quently efficient to compute in applications. In particular, uncertainty-based acquisition functions favor query points that are near the current classifier's decision boundary whose inferred classifications are interpreted as most "uncertain". These methods are emblematic of *exploitative* AL since they explicitly use measures of distances to decision boundaries to select query points [4]. Moreover, some of the uncertainty-based acquisition functions designed for graph-based classifiers include the uncertainty norm for Poisson-reweighted Laplace Learning [4], variance minimization (VOpt) [5], model-change (MC) [6, 7], model-change variance optimality (MCVOpt) [6, 7], Σ -optimality [8], hierarchical sampling for active learning [9], cautious active clustering [10], and shortest-shortest path S^2 [11]. In this paper, we introduce a novel uncertainty-based acquisition function that leverages the computed optimal price and incentive values from the underlying auction dynamics graph-based classifier.

Acquisition functions for the AL process have also been integrated into deep learning frameworks to boost the performance of such parametric models. In particular, some of the recent works include batch AL by diverse gradient embeddings (BADGE) [12], AL framework in Bayesian deep learning [13], diverse batch acquisition for deep Bayesian AL [14], diffusion-based deep AL [15], stochastic batch acquisition functions [16], and AL for convolutional neural networks [17].

On the other hand, an important component of the AL process is the underlying method used for training a classifier based on the currently labeled data \mathcal{L} . To this end, (similarity) graph-based methods have proven to be effective models for semi-supervised and AL settings, particularly in the low label rate regime [18–21]. These methods leverage both the labeled and unlabeled sets to construct a similarity graph on which the observed

labels on \mathcal{L} are used to make inferences of the labels on the unlabeled data according to the graph topology; thus, the geometric structure of the dataset is modeled with the graph and similar data points receive similar inferred outputs. Such techniques are beneficial since they utilize the structure of and the information from the usually abundant unlabeled data to build a more effective model.

In contrast, variants of auction algorithms, originally developed by Bertsekas [22–25] for solving the classic assignment problem, have a simple intuitive structure, are easy to code, and have excellent performance in both accuracy and timing. In fact, auction methods substantially outperform their main competitors for important problems, both in theory and in practice, and are also naturally well-suited for parallel computation during coding. Therefore, in this paper, we integrate (similarity) graph-based techniques and auction-based procedures to derive the underlying algorithm used in the active learning process, as well as the acquisition function.

In [26], the authors have shown how auction algorithms can be utilized for various applications, such as semi-supervised graph-based learning and classification in the presence of equality or inequality class size constraints on the individual classes. Modifications of this technique for the classification of 3D sensory data are discussed in [27]. Moreover, auction methods can be used in Merriman, Bence, and Osher (MBO)-based threshold dynamics schemes, which involve threshold dynamics to tackle different tasks; this was shown in [26]. For example, the original MBO scheme [28] was developed in 1992 to simulate and approximate motion by mean curvature; the procedure consists of alternating between diffusion and thresholding. Such an MBO technique was then modified and adapted to a similarity graph setting for the purpose of binary classification and image inpainting in [29], with a multiclass extension developed in [30] and

[31]. Applications of the graph-based technique to heat kernel PageRank and hyperspectral imagery and video are detailed in [32] and [33], respectively. A summary of recent graph-based optimization approaches for machine learning, including MBO-based ones, is presented in [34].

Moreover, graph neural networks (GNN) [35] constitute an important area of modern, graph-based learning methods that have found application in node classification, graph classification, and link prediction. These methods leverage a graph's structure of nodes and edges to define compositions of transformations of node features in a fashion that is similar to deep neural networks. Notable contributions in this area include [36–38], with advances in graph classification highlighted in [39, 40].

With regards to active learning for graph-based semi-supervised classifiers, various recent pipelines have been developed for image processing applications, e.g., image segmentation [41], surface water and sediment detection [42], classification of synthetic aperture radar (SAR) data [43, 44], unsupervised clustering of hyperspectral images using nonlinear diffusion [45], hyperspectral unmixing [42], medical image analysis [46], hyperspectral image classification [47], and subgraph matching [48].

Overall, in this paper, we integrate active learning components to develop MALADY, an accurate active learning algorithm incorporating auction dynamics on similarity graphs for semi-supervised learning tasks.

The contributions of the paper are as follows:

- We propose a new active learning technique based on auction dynamics. For the underlying semi-supervised similarity graph-based classifier in our active learning process, we incorporate elements of the auction dynamics algorithm for semi-supervised learning in [26] and use a more general optimization energy functional. The general formulation can be reformulated as a series of modified assignment problems, each of which can be tackled using auction methods incorporating class size constraints.
- We propose a novel acquisition function for the active learning process that uses the dual variables of the dual formulations of the modified assignment problems incorporating upper and lower bound class size constraints. The proposed acquisition function not only measures the uncertainty of the volume bound auction algorithm for graph-based semi-supervised problems but also captures salient geometric classifier information that can be leveraged for exploitation of decision boundaries.
- The proposed method is able to perform accurately even when the labeled set is extremely small, as small as just a few labeled elements per class. This is important since labeled data is scarce for many applications. In addition, the algorithm integrates class size information, improving accuracy further.

- We conduct experiments on data classification using various datasets, and the results demonstrate that our proposed framework performs more accurately compared to other state-of-the-art methods.

The remainder of the paper is organized as follows: in Section II, we present background information on the graph-based framework and the auction algorithm with volume constraints. In Section III, we derive our semi-supervised classifier and the proposed acquisition function. The results of the experiments on benchmark data sets and the discussion of the results are presented in Section IV. Section V provides concluding remarks.

II. BACKGROUND

In this background section, we first discuss the (similarity) graph construction technique, which is fundamental for graph-based methods. Then, we discuss auction dynamics with volume constraints for semi-supervised learning [26] as the technique that is fundamental for the construction of our proposed semi-supervised method. The last subsection reviews some semi-supervised learning algorithms using auction dynamics procedures. The notation used in this paper is described in Table I.

Variable	Description
N	number of nodes
\mathcal{X}	dataset, vertex set
\mathcal{L}	labeled set
\mathcal{U}	unlabeled set
\mathcal{Q}	query set
K	number of similarity classes
U_i	upper bound for class i
B_i	lower bound for class i
S_i	similarity class for class i
V_i	cardinality of similarity class S_i

TABLE I: Notation Table

A. Graph Construction

In this section, we review the graph-based framework we use in this paper. Consider a dataset $\mathcal{X} = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^d$, where each data element is represented by a d -dimensional feature vector. We generate a undirected graph $G(\mathcal{X}, W)$ of vertices and edges between vertices, where \mathcal{X} is set of vertices representing the data elements, and $W \in \mathbb{R}^{N \times N}$ is the weight matrix, consisting of weights between pairs of vertices. The weight matrix W is computed using a weight function $w : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, where $W_{i,j} = w(x_i, x_j)$ denotes the weight on the edge between the vertices x_i and x_j , where $i \neq j$. Overall, the weight matrix quantifies the similarities between the feature vectors, i.e., data elements, whether labeled or unlabeled; thus, the graph-based framework can provide crucial information about the data and leverage information from the abundant unlabeled data. Some popular weight functions include:

- Gaussian weight function

$$w(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{\sigma^2}\right), \quad (1)$$

where $d(x_i, x_j)$ represents a distance (computed using a chosen measure, such as Euclidean distance) between vertices x_i and x_j , associated with the i^{th} and j^{th} data elements, and $\sigma > 0$ is a parameter which controls the scaling in the weight function.

- cosine similarity weight function [49]

$$w(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{\sigma(x)\sigma(y)}\right), \quad (2)$$

where $d(x_i, x_j) = \arccos\left(\frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}\right)$ and $\sqrt{\sigma(x_i)} = d(x_i, x_M)$ is a local parameter for each x_i , and x_M is the M^{th} closest vector to x_i .

With the assumption that the high-dimensional data is concentrated near a low-dimensional manifold and the manifold is locally Euclidean, we only compute the k nearest neighbors (kNN) of each point using an approximate nearest neighbor search algorithm. This ensures that the weight matrix is sparse and enhances efficiency. For an approximate nearest neighbor search, we use the GraphLearning package [19], which uses the Annoy library [50]. To preserve the symmetric property of the weight matrix, we calculate $W^{final}_{ij} = \{W_{ij} + W_{ji}\}/2$.

B. Auction Algorithm with Exact Volume Constraints

Since our proposed method involves auction techniques originally designed to solve the assignment problem, we review the procedures here. Given two disjoint sets \mathcal{X} and \mathcal{Y} of same cardinality r and a benefit function $c : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, the assignment problem aims to identify a one-to-one correspondence $M = \{(x_1, y_1), \dots, (x_r, y_r)\}$ of sets X and Y , so that the total benefit of the matching

$$\sum_{(x,y) \in M} c(x, y) \quad (3)$$

is maximized. The assignment problem (3) can be restated as the following optimization problem by representing the matching as a binary vector z , where $z_y(x) = 1$ if (x, y) are matched and 0 otherwise:

$$\begin{aligned} \max_{z: \mathcal{X} \times \mathcal{Y} \rightarrow \{0,1\}} & \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} c(x, y) z_y(x) \\ \text{s.t.} & \sum_{x \in \mathcal{X}} z_y(x) = 1 \quad \forall y, \quad \sum_{y \in \mathcal{Y}} z_y(x) = 1 \quad \forall x. \end{aligned} \quad (4)$$

Moreover, the above optimization problem can be written as a classical linear programming problem (LP) if we relax the binary constraint on z :

$$\begin{aligned} \max_{z \geq 0} & \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} c(x, y) z_y(x) \\ \text{s.t.} & \sum_{x \in \mathcal{X}} z_y(x) = 1, \quad \sum_{y \in \mathcal{Y}} z_y(x) = 1. \quad (LP) \end{aligned} \quad (5)$$

This relaxation turns out to be exact since the solution to a bounded and feasible linear programming problem always includes a vertex of the feasible polytope

\mathcal{P} . The relaxed constraint set in (5) is the polytope $\mathcal{P} = \left\{z \geq 0 : \sum_{x \in \mathcal{X}} z_y(x) = 1, \sum_{y \in \mathcal{Y}} z_y(x) = 1\right\}$, whose vertices are precisely vectors z whose entries are binary. Furthermore, we can split \mathcal{Y} into K similarity classes $\{S_i\}_{i=1}^K$ each of size V_i , and let $c(x, y) = a_i(x)$ for each $y \in S_i$. With these choices, we can reformulate (5):

$$\begin{aligned} \max_{z \geq 0} & \sum_{x \in \mathcal{X}} \sum_{i=1}^K a_i(x) \sum_{y \in S_i} z_y(x) \\ \text{s.t.} & \sum_{x \in \mathcal{X}} z_y(x) = 1, \quad \sum_{y \in \mathcal{Y}} z_y(x) = 1 \end{aligned} \quad (6)$$

Using $u_i(x) = \sum_{y \in S_i} z_y(x)$, we obtain equivalently:

$$\begin{aligned} \max_{u \geq 0} & \sum_{x \in \mathcal{X}} \sum_{i=1}^K a_i(x) u_i(x) \\ \text{s.t.} & \sum_{x \in \mathcal{X}} u_i(x) = V_i, \quad \sum_{i=1}^K u_i(x) = 1. \end{aligned} \quad (7)$$

We will now focus on solving (7), which is a special case of (5). First, it is helpful to practically interpret (7). In particular, one can view each class in the data classification problem as an institution, such as a gym, that offers a certain number of memberships and the data elements as people trying to obtain a deal on buying access to only one of the institutions. Moreover, the coefficients $a_i(x)$ can represent person x 's desire to buy the membership in institution i . In this view, the solution to (7) maximizes the total satisfaction of the population. Notice that the class size constraints (i.e. number of memberships available) make the assignment of institution memberships to people nontrivial.

One approach to assigning memberships involves the market mechanism where an institution j is equipped with a price p_j . Then, person x will want to buy the membership from an institution that offers the best value:

$$j^*(x) = \arg \max_{1 \leq j \leq K} a_j(x) - p_j \quad (8)$$

However, the resulting matching will probably not satisfy the class size constraints. Thus, the challenge is to compute an optimal price vector, called an equilibrium price vector, that results in an institution-person matching that satisfies the constraints on the number of available memberships. The answer lies in the dual formulation of the assignment problem, which can be shown to be the following problem:

$$\begin{aligned} \min_{p \in \mathbb{R}^K, \pi \in \mathbb{R}^N} & \sum_{i=1}^K p_i V_i + \sum_x \pi(x) \\ \text{subject to} & \pi(x) + p_i \geq a_i(x). \end{aligned} \quad (9)$$

In fact, it turns out that the equilibrium price vector p_* of the primal problem is in fact the optimal solution of the dual problem (9). Moreover, the optimal value of π

is completely determined by \mathbf{p} ; given a price vector \mathbf{p} , the above dual problem is minimized when $\pi(x)$ equals the maximum value of $a_i(x) - p_i$ over i .

According to the complementary slackness (CS) condition, a complete assignment M and a price vector \mathbf{p} are primal and dual optimal if and only if each person is assigned to an institution offering the best deal to them. Moreover, the CS condition can be relaxed to allow a person to be assigned to institutions that are within ϵ of achieving the best deal for them in the definition (8). In particular, we say that an assignment M and a price vector \mathbf{p} satisfies ϵ -complementary slackness (ϵ -CS) if

$$\pi(x) - \epsilon \leq a_i(x) - p_i \text{ for all } i. \quad (10)$$

This modification is very useful in the case of a price war. In a price war, multiple people compete for the same memberships without ever raising the prices of the institutions, trapping the algorithm in an infinite loop.

In [22], Bertsekas detailed an auction technique for solving the assignment problem by finding the equilibrium (optimal) price vector. Since the seminal paper, Bertsekas and others have extended the technique to more general problems and have improved the computational aspects. For example, [24] develops a technique that efficiently handles assignment problems with multiple identical objects. Overall, an exhaustive reference on auction algorithms is included in [23], which contains information on linear network optimization. Continuous and discrete models for network optimization are described in [25].

Bertsekas's auction technique contains the following steps: at the beginning of an iteration, one starts with an assignment and price vector satisfying ϵ -CS with $\epsilon > 0$. There are two phases in each iteration: the bidding phase and the assignment phase (the assignment M and price vector \mathbf{p} is updated while maintaining ϵ -CS):

- Bidding phase: For each x , under the assignment M :
 - Compute the current value $v_i = a_i(x) - p_i$ for each class i , and choose $i^* \in \arg \max v_i$.
 - Find best value offered by a class other than i^* :

$$u_i = \arg \max_{i \neq i^*} v_i. \quad (11)$$

- Compute the bid $b(x)$ of element x for class i^* :

$$b(x) = p_{i^*} + \epsilon + v_{i^*} - u_i \quad (12)$$

- Assignment phase:
 - If class i^* has already given out V_{i^*} memberships, remove y currently assigned to the class with the lowest bid and add x to class i^* , and set p_{i^*} to be the minimum bid value over all currently assigned to class i^* .
 - If class i^* has not yet given out all V_{i^*} memberships, add x to class i^* . If now all memberships of class i^* are bought, set p_{i^*} to be the minimum

bid value over all currently assigned to class i^* . Otherwise, the price remains the same.

The Membership Auction algorithm described above is given in Appendix as Algorithm 5.

C. Auction Algorithm with Upper and Lower Bound Volume Constraints

In many scenarios, the exact number of elements in each class is unknown. Thus, in [26], the authors present an upper and lower bound auction algorithm for semi-supervised learning that allows the size of each class to fluctuate between upper and lower bounds. In particular, let K be the number of classes, and suppose that class j must have at least B_j members and at most U_j members, where $B_j \leq U_j$, and $\sum_{j=1}^K B_j \leq N \leq \sum_{j=1}^K U_j$, where N is the number of data elements in the data set.

With these additional constraints, the modified version of the assignment problem becomes:

$$\begin{aligned} & \max_{u \geq 0} \sum_x \sum_{i=1}^K a_i(x) u_i(x) \\ & \text{s.t.} \sum_{i=1}^K u_i(x) = 1, \quad B_i \leq \sum_x u_i(x) \leq U_i \quad \forall i. \end{aligned} \quad (13)$$

The addition of these bounds in the optimization problem introduces some complexities in the problem. For example, each data element x always aims to obtain the most desirable class j , which may result in a deficiency of members in other classes. To solve this problem, the authors of [26] introduce an idea of incentives \mathbf{t} in the market mechanics. In particular, class i must sell B_i memberships, and if it is having trouble attracting a certain number of members, it should offer incentives to attract elements to join the class. This results in competition among the classes, and the classes that are deficient in members are forced to offer competitive incentives to attract the necessary number of data elements. To satisfy the lower bounds, one needs to apply an adaptation of the reverse auction algorithm of Bertsekas and coauthors [51] and have the classes bid on the data elements.

Following the work of [26], the modified dual problem with price and incentives is given as

$$\begin{aligned} & \min_{\mathbf{p} \geq 0, \mathbf{t} \geq 0, \boldsymbol{\pi} \in \mathbb{R}^N} \sum_{i=1}^K (p_i U_i - t_i B_i) + \sum_x \pi(x) \\ & \text{subject to} \quad \pi(x) + p_i - t_i \geq a_i(x) \quad \forall i. \end{aligned} \quad (14)$$

The complementary slackness condition for (13) and (14) states that \mathbf{u} and $(\mathbf{p}, \mathbf{t}, \boldsymbol{\pi})$ are optimal if and only if

$$\begin{aligned} & \sum_{i=1}^K \sum_x u_i(x) (a_i(x) - p_i + t_i - \pi(x)) \\ & + \sum_{i=1}^K p_i (U_i - \sum_x u_i(x)) + \sum_{i=1}^K t_i (\sum_x u_i(x) - B_i) = 0. \end{aligned} \quad (15)$$

With the addition of incentives, the ϵ -CS condition for every matched pair (x, i) satisfies

$$a_i(x) - p_i + t_i + \epsilon \geq \max_{1 \leq j \leq N} a_j(x_i) - p_j + t_j. \quad (16)$$

Unfortunately, the last two terms in (15) do not have useful ϵ relaxations. Thus, [26] proposes a two-stage method to solve (13). First, one should run the forward auction algorithm, Algorithm 4 in the Appendix, to satisfy the upper bound constraints with a complete ϵ -CS matching. The output of this stage is then fed into a lower bound auction, Algorithm 3 in the Appendix; the goal of the lower bound auction is to obtain a matching that satisfies the lower bound class size constraints as well. Thus, at the end, one will obtain a ϵ -CS matching satisfying *both* the upper and lower bound class constraints. We refer the reader to [26] for a detailed description.

III. PROPOSED METHOD

A. Proposed Semi-Supervised Learning Framework

1) *Notation*: We use a graph-based framework to derive our underlying semi-supervised classifier. Let K denote the number of classes, and let the data set \mathcal{X} consist of the training elements \mathcal{L} with label information and \mathcal{U} , the unlabeled training data elements. We embed our data set into a weighted similarity graph $G(\mathcal{X}, W)$ using the similarity functions mentioned in Section II-A. For class i , let B_i and U_i denote the lower bounds and upper bounds of the class, respectively. If exact class sizes are known, we can set $B_i = U_i$, and when the class information is not available, we simply set $B_i = 0$ and $U_i = |\mathcal{X}|$. Let $l_x \in \{1, \dots, K\}$ denote the label of $x \in \mathcal{L}$, e_i denote a vector with 1 in the i^{th} place and 0 elsewhere.

2) *Derivation*: The proposed method is derived using constrained optimization of an energy of the form

$$E(\mathcal{X}) = R(\mathcal{X}) + F(\mathcal{X}), \quad (17)$$

where $R(\mathcal{X})$ is a regularizing term which ensures the smoothness in the partition $\mathcal{X} = (\mathcal{X}_1, \dots, \mathcal{X}_K)$ into K classes, and $F(\mathcal{X})$ is a fidelity term containing information from the labeled training data. We can reformulate (17) by imposing constraints that incorporate the labeled training data and class size information:

$$B_i \leq |\mathcal{X}_i| \leq U_i, \quad \mathcal{L}_i \subset \mathcal{X}_i \quad \text{for all } 1 \leq i \leq K, \quad (18)$$

where U_i and B_i are upper and lower bounds on the class sizes, respectively, and \mathcal{L}_i is the set of labeled training points with label i . Then, to guarantee a notion of smoothness for partitions of the data set \mathcal{X} , one can use a weighted graph cut as a regularizing term. The weighted graph cut is defined as the following term:

$$\text{Cut}(\mathcal{X}) = \frac{1}{2} \sum_{i=1}^K \sum_{x \in \mathcal{X}_i} \sum_{y \notin \mathcal{X}_i} w(x, y), \quad (19)$$

where the entries $w(x, y)$ describe how strongly the points x and y are connected. Unfortunately, the graph

cut problem is NP-hard. Instead, similarly to [26, 52], one can consider the graph heat content (GHC) as a convex relaxation of the graph cut. It is defined as:

$$\text{GHC}(\mathbf{u}, W) = \sum_{i=1}^K \sum_{x, y \in \mathcal{X}} w(x, y) u_i(x) (1 - u_i(y)), \quad (20)$$

where $\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K$ is an element of the convex relaxation of the space of K -phase partitions of \mathcal{X} . For each $x \in \mathcal{X}$, $\mathbf{u}(x) = (u_1(x), \dots, u_K(x)) \in \mathbb{R}^K$, where $u_i(x)$ is the i^{th} entry of vector $\mathbf{u}(x)$. As long as the matrix W , for which $W_{ij} = w(x_i, x_j)$, is a positive semi-definite matrix, the above GHC term is a concave term.

With this in mind, we propose the following problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \text{GHC}(\mathbf{u}, W) + J_{\text{conc}}(\mathbf{u}) \\ \text{subject to} \quad & \sum_{i=1}^K u_i(x) = 1 \quad \forall x, \\ & u(x) = e_{l_x} \quad \forall x \in \mathcal{L}, \\ & B_i \leq \sum_{x \in \mathcal{U}} u_i(x) \leq U_i \quad \forall i \end{aligned} \quad (21)$$

over $\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K$, $\mathbf{u} \geq 0$. In this formulation, $\text{GHC}(\mathbf{u}, W)$ is the graph heat content energy in (20), which depends on the graph weights, and $J_{\text{conc}}(\mathbf{u})$ is any concave function of \mathbf{u} , including any linear function in \mathbf{u} .

The $J_{\text{conc}}(\mathbf{u})$ term is a term that allows one to consider a wide range of optimization problems within our framework. Without this term, the problem would only involve minimizing the graph heat content subject to certain constraints. The $J_{\text{conc}}(\mathbf{u})$ term and some examples of it one can use are described at the end of Section IIIA.

The motivation for considering the model (21) is:

- The first term of (21) is a convexified graph cut, and its minimizer groups data elements so that those of different classes are as dissimilar as possible.
- One can integrate information about class sizes, which can often improve algorithm performance.
- It enables one to incorporate a combination of weighted edge-based, class-based, and label-based terms that contain important information about the data set elements. This can improve accuracy.
- For a positive semi-definite matrix $W_{ij} = w(x_i, x_j)$, the energy is a concave function of \mathbf{u} . Therefore, one may obtain a minimization procedure for (21) by considering a "gradient flow" scheme involving linearizations of the energy under constraints.

Overall, each step of the scheme to solve (21) is equivalent to finding the $(k+1)^{\text{th}}$ iterate \mathbf{u}^{k+1} as a solution of the following modified assignment problem:

$$\arg \max_{\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K, \mathbf{u} \geq 0} \sum_{i=1}^K \sum_{x \in \mathcal{U}} (1 - (\nabla J_{\text{conc}})_i(x) - \sum_{y \notin \mathcal{X}_i^k} w(x, y)) u_i(x) \quad (22)$$

$$\begin{aligned} \text{s.t. } & \sum_{i=1}^K u_i(x) = 1, u(x) = e_{l_x} \quad \forall x \in \mathcal{L}, \\ & L_i \leq \sum_{x \in \mathcal{U}} u_i(x) \leq U_i \quad \forall i, \end{aligned} \quad (23)$$

At each iteration, a partition of the data can be formed via $\mathcal{X}_i^{k+1} = \{x \text{ s.t. } \arg \max_j u_j^{k+1}(x) = i\}$. Moreover, the elements $\{a_i(x)\}$, where

$$a_i(x) = (1 - (\nabla J_{\text{conc}})_i(x) - \sum_{y \notin \mathcal{X}_i^k} w(x, y)), \quad (24)$$

are referred to as coefficients of the modified assignment problem. Note that the above scheme is equivalent to the upper and lower bound assignment problem (13). Our goal is to solve each step of the scheme in (22)-(23) with an auction-type algorithm that incorporates lower and upper bound class size constraints and also integrates a novel active learning component which will be described in future sections of the paper.

One can solve each step of the scheme in (22)-(23) using an auction-type technique similar to the one used in [26] and described in Section IIC. In particular, one alternates between an upper bound auction procedure and a lower bound auction procedure while decreasing the ϵ parameter after each lower bound auction. This ϵ -scaling procedure is used to reduce the complexity of the technique and is described in more detail in [26]. Our semi-supervised learning framework to be used in the active learning process is described as Algorithm 1.

In general, one can consider many choices for the function J_{conc} . For example, in this paper, we consider a Poisson term, such as the one defined in [19], as our concave function J_{conc} . In particular, we consider $J_{\text{conc}} = -\gamma \sum_{j=1}^k (l_j - \bar{l}) \cdot u(x_j)$, where k is the number of labeled elements, $\{x_1, \dots, x_k\}$ are the labeled data elements with labels $\{l_1, \dots, l_k\} \in \{e_1, \dots, e_k\}$, which are indicator vectors in \mathbb{R}^k , $\gamma > 0$, and $\bar{l} = \frac{1}{k} \sum_{j=1}^k l_j$. An optimization problem consisting of this term, coupled with a graph-based regularizer (20), will result in a method that is provably advantageous for low label rates; please refer to [19] for a theoretical justification of Poisson-like terms.

Another example of J_{conc} one can use is the modularity term in [53]: $-\gamma \sum_i \sum_{x \in \mathcal{X}} k_x |u_i(x) - \frac{1}{2r} \sum_{y \in V} k_y u_i(y)|^2$, where a total variation optimization problem with this term is shown to be equivalent to modularity optimization, which tries to detect communities in the graph based on their modularity. Here, $k_x = \sum_y w(x, y)$ is the degree of vertex x , $\gamma > 0$ is a resolution parameter, and $2r = \sum_x k_x$. Using this term will open new avenues to approach network science and detection of communities.

Lastly, it is clear that any linear term in u , such as $\sum_{i=1}^m \sum_{x \in \mathcal{X}} R_i(x) u_i(x)$, can be used for J_{conc} . In particular, one can assign $R_i(x)$ to be a class homogeneity term, where $R_i(x)$ is the cost of assigning x to class i . In the

latter case, one should formulate $R_i(x)$ so that it is small if x is likely to belong to class i and large otherwise. The terms may be defined using the eigenvectors of the graph Laplacian or using a fit to an expected value of a variable. Please refer to [54] for a more detailed description of class homogeneity terms.

Algorithm 1: Auction Dynamics Semi-supervised learning (SSL) framework

Input: Vertex set \mathcal{X} , Weight matrix W , initial partition \mathcal{X} , lower bound B , upper bound U , time step δt , number of steps s , auction error tolerance ϵ_{\min} , epsilon scaling factor α , initial epsilon ϵ_0 .

Result: Final configuration \mathcal{X}^s , price vector p , incentives vector t

Initialization: Set $\mathcal{X}^0 := \mathcal{X}$, set $\bar{\epsilon} = \epsilon_{\min}/N$;

for k **from** 0 **to** $s - 1$ **do**

 Calculate assignment problem coefficients:

$$a_i^{k+1}(x) = (1 - (\nabla J_{\text{conc}})_i(x) - \sum_{y \notin V_i^k} w(x, y));$$

 Initialize prices $p = \mathbf{0}$, incentives $t = \mathbf{0}$, $\epsilon = \epsilon_0$;

while $\epsilon \geq \bar{\epsilon}$ **do**

 Run Algorithm 4 (Upper Bound Auction):

$$(\mathcal{X}_{\text{out1}}, p_{\text{out1}}, t_{\text{out1}}) = \text{UBA}(\epsilon, B, U, a^{k+1}, p, t, \mathcal{X});$$

 Run Algorithm 3 (Lower Bound Auction):

$$(\mathcal{X}_{\text{out2}}, p_{\text{out2}}, t_{\text{out2}}) = \text{LBA}(\epsilon, B, U, a^{k+1}, p_{\text{out1}}, t_{\text{out1}}, \mathcal{X}_{\text{out1}});$$

 Set $(p, t) = (p_{\text{out2}}, t_{\text{out2}})$;

 Divide ϵ by α ;

if $\epsilon < \bar{\epsilon}$ **then**

 Set $\mathcal{X}^{k+1} = \mathcal{X}_{\text{out2}}$;

end

end

end

return (\mathcal{X}^s, p, t)

B. Proposed active learning acquisition function

In the active learning process shown in Figure 1, a chosen acquisition function \mathcal{A} is used to select informative query points for which labels will be requested. Before defining our proposed acquisition function, several variables need to be introduced. First, for each unlabeled point x , we calculate:

$$\mathcal{M}(x) = v(x) - w(x), \quad (25)$$

where $v(x) = \max_{1 \leq i \leq K} (a_i(x) - p_i + t_i + \epsilon)$ and $w(x) = \max_{1 \leq i \leq K, i \neq i^*} (a_i(x) - p_i + t_i + \epsilon)$. Here, $a_i(x)$ is the coefficient of the modified assignment problem as defined in (24), p_i and t_i represent the price and incentives, respectively, for class i , and $i^* = \arg \max_{1 \leq i \leq K} (a_i(x) - p_i + t_i + \epsilon)$. As discussed in section II-C, the term $v(x)$ and $w(x)$ represent the best and second best deal values, respectively, in the market mechanism, offered to x by the end of auction method, and it is important to note that when calculating \mathcal{M} , one should use the final price

and incentive vector returned from Algorithm 1. Then, the general intuition to motivate our proposed acquisition function is that there is some level of uncertainty as to the optimal class of x when the value of $\mathcal{M}(x)$ is relatively small, and there is little uncertainty as to the optimal class when $\mathcal{M}(x)$ is big.

Overall, to align with a unified framework for maximizing acquisition functions for query point selection, we define our proposed acquisition function as follows:

The acquisition function (26) is a novel uncertainty sampling acquisition function [1] where the optimal price vector \mathbf{p} , optimal incentives vector \mathbf{t} , auction coefficient α and epsilon ϵ are used in each active iteration to determine the query points. Overall, one can think of this acquisition function as prioritizing *minimum margin sampling*, where the margin is defined as the difference between the best and second-best deal values offered to x at the end of the auction algorithm. The strategy is to query a point that has the smallest margin, as a smaller margin indicates greater uncertainty in the decision.

To gain better intuition, we run an experiment on a toy dataset in two dimensions to visualize which points are emphasized by the proposed acquisition function. The dataset consists of six Gaussian clusters, where each cluster contains 400 points and has a standard deviation of 0.25, and where the centers lie evenly spaced along the unit circle. The classification of the clusters is then assigned in an alternating pattern, as illustrated in Figure 2(a). With six initially labeled points, we sequentially query 100 points for labeling via the proposed acquisition function (26). Heatmaps of the values for our proposed acquisition function 26 and uncertainty sampling [1] with a Laplace learning [55] classifier are shown in panels (b) and (c), respectively. Note that brighter regions in these plots represent larger acquisition function values, helping us to visualize where the largest acquisition function values are concentrated. In our proposed active learning framework, the query point that is selected at each corresponding iteration would be the maximizer of the proposed acquisition function, representing the “brightest” point in the corresponding heatmap.

The main observation from this experiment is that *the proposed acquisition function accurately prioritizes data points along all boundaries between oppositely labeled clusters*. Specifically, points at boundaries with more overlap between clusters have the highest acquisition function values. This contrasts with a simple uncertainty sampling [1] acquisition function with the Laplace learning classifier [56], which in Figure 2 (c) demonstrates that only a subset of the important boundaries between oppositely labeled clusters are emphasized in sampling. Overall, successful active learning acquisition functions need to efficiently prioritize these decision boundary regions (referred to as “exploitation” in active learning [4]). The results of this experiment suggest that the proposed

acquisition function can efficiently prioritize decision boundaries modeled by regions between clusters in the corresponding similarity graph structure.

Using the proposed acquisition function (26), we turn to define our multiclass active learning method, MALADY.

Algorithm 2: MALADY: Multiclass Active Learning with Auction Dynamics on Graphs

Input: Vertex set \mathcal{X} , weight matrix W , initial partition \mathcal{X} , lower bound B , upper bound U , time step δt , number of steps s , auction error tolerance ϵ_{min} , epsilon scaling factor α , initial epsilon ϵ_0 , initial labeled set \mathcal{L}_0 , acquisition function $\mathcal{A}(x)$, and budget T .

Result: Accuracy on unlabeled set.

Initialization: $\mathcal{L} = \mathcal{L}_0$;

while $|\mathcal{L}| \leq T$ **do**

 Run Algorithm 1 (Semi-supervised method):

$(\mathcal{X}^s, \mathbf{p}, \mathbf{t}) = \text{SSL}(W, \mathcal{X}, \mathcal{X}, B, U, \delta t, s, \epsilon_{min})$;

 Pick $x^* \in \mathcal{U} = \mathcal{X} \setminus \mathcal{L}$ which has highest acquisition value $\mathcal{A}(x)$;

$\mathcal{L} \leftarrow \mathcal{L} \cup \{x^*\}$;

end

Run Algorithm 1 with labeled set \mathcal{L}

(Semi-supervised classification method):

$(\mathcal{X}^s, \mathbf{p}, \mathbf{t}) = \text{SSL}(W, \mathcal{X}, \mathcal{X}, B, U, \delta t, s, \epsilon_{min})$;

Evaluate the performance of the classifier;

return *Evaluation of algorithm*

C. Multiclass Active Learning Procedure

We now describe our proposed Multiclass Active Learning with Auction Dynamics on graphs (MALADY) procedure. Given an initially labeled set \mathcal{L}_0 with corresponding labels $\{y(x_i)\}_{x_i \in \mathcal{L}}$, we sequentially select query points that maximize our proposed acquisition function (26), $\mathcal{A}(x) = 1 - \mathcal{M}(x)$. We summarize our method as follows: Given input data and an initially labeled subset \mathcal{L}_0 , our proposed method consists of:

- 1) Construct a similarity graph from the data using a chosen similarity function such as (1).
- 2) With a current labeled set \mathcal{L} , run the underlying semi-supervised classifier (Algorithm 1) to get the optimal price and incentive vectors. For the first iteration, $\mathcal{L} = \mathcal{L}_0$.
- 3) Using acquisition function (26), select the (next) query point as: $x^* = \arg \max_{x \in \mathcal{U}} \mathcal{A}(x)$.
- 4) Query for the label y^* of the query point x^* and update the labeled set $\mathcal{L} \leftarrow \mathcal{L} \cup \{x^*\}$.
- 5) Repeat steps 2 through 4 until a stopping criterion, such as a budget limit $|\mathcal{L}| = T$, is met.
- 6) With a current labeled set \mathcal{L} , run the underlying semi-supervised classifier (Algorithm 1) to get the partition of the vertex set, i.e., the data.
- 7) Evaluate the classifier’s performance by calculating the chosen evaluation metric on the unlabeled set.

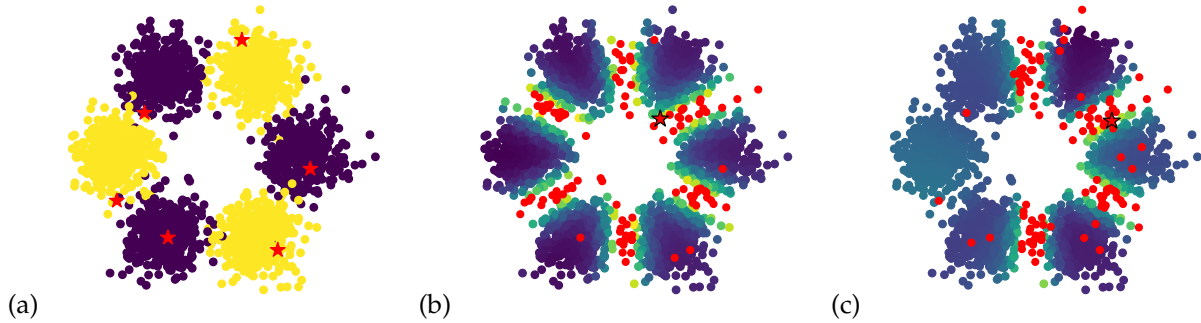


Fig. 2: (a) Ground truth with six initial points, (b) Acquisition function value for our proposed acquisition function (26) at Iteration 100, (c) Acquisition function value of uncertainty sampling [1] with Laplace learning [55] at Iteration 100. Brighter regions of the plot indicate larger acquisition function values, and labeled points are marked in red circles; the query point for the current iteration is marked as a red star with a black outline. Notice that our acquisition function (b) focuses on *all* decision boundaries between oppositely labeled clusters, whereas standard uncertainty sampling (c) only focuses on a subset of these boundaries.

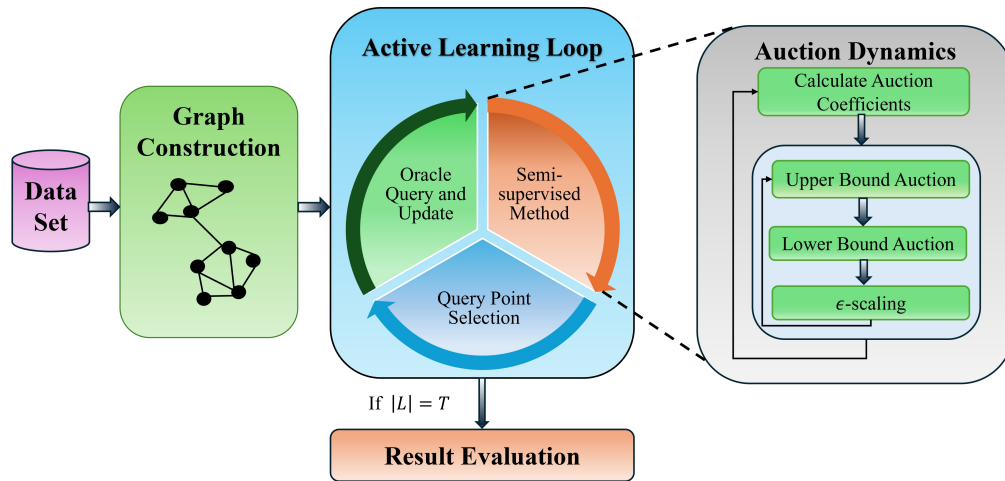


Fig. 3: The flowchart of our MALADY. Green box: Similarity graph construction using a chosen similarity function (section II-A); Blue box: Active learning process using proposed acquisition function (section III-B); Grey box: Auction dynamics method for semi-supervised inference (section III-A). The result is the partition of the vertex set; Red box: When $|L| = T$, the accuracy of the proposed method is computed.

The flowchart of MALADY is shown in Figure 3, and the algorithm is also described in detail as Algorithm 2.

IV. RESULTS AND DISCUSSION

A. Data sets

For the computational experiments in this paper, we use eight benchmark datasets, including two hyperspectral datasets: Landsat [57], USPS [58], Jasper Ridge [59], Urban [60], COIL-20 [61], CIFAR-10 [62], Opt-Digits [63], and Fashion-MNIST [64]. The details of these data sets are provided in the supporting documentation.

B. Comparison to other methods

In this section, we compare our proposed technique to those using other active learning acquisition functions in various graph-based SSL classifiers. The results of the experiments are shown in Figure 2. In particular, for all data sets, we compare our proposed method, MALADY, to the following methods: Random sampling

(Random), Variance minimization criterion (VOpt) [5], Minimum norm uncertainty sampling (Unc.-Norm)[4], Model change active learning (MCVOpt) [65], Uncertainty sampling (Unc.-Laplace) [1], and S^2 [66].

For a fair comparison, we used Algorithm 1 as the underlying SSL classifier for the Random, VOpt, MCVOpt, and S^2 methods since these acquisition functions are not tied to a particular, underlying graph-based semi-supervised classifier. Minimum norm uncertainty sampling (Unc.-Norm) and Unc.-Laplace, however, are acquisition functions that at each iteration depend upon the underlying classifier's predictions to quantify uncertainty, so we report the accuracy in their originally proposed graph-based SSL classifiers; respectively, Poisson-reweighted Laplace learning [4, 67] and Laplace learning [55].

Now, we provide a brief overview of our comparison methods. In [5] (VOpt method), the authors analyze the probability distribution of unlabeled vertices conditioned on the label information, which follows a mul-

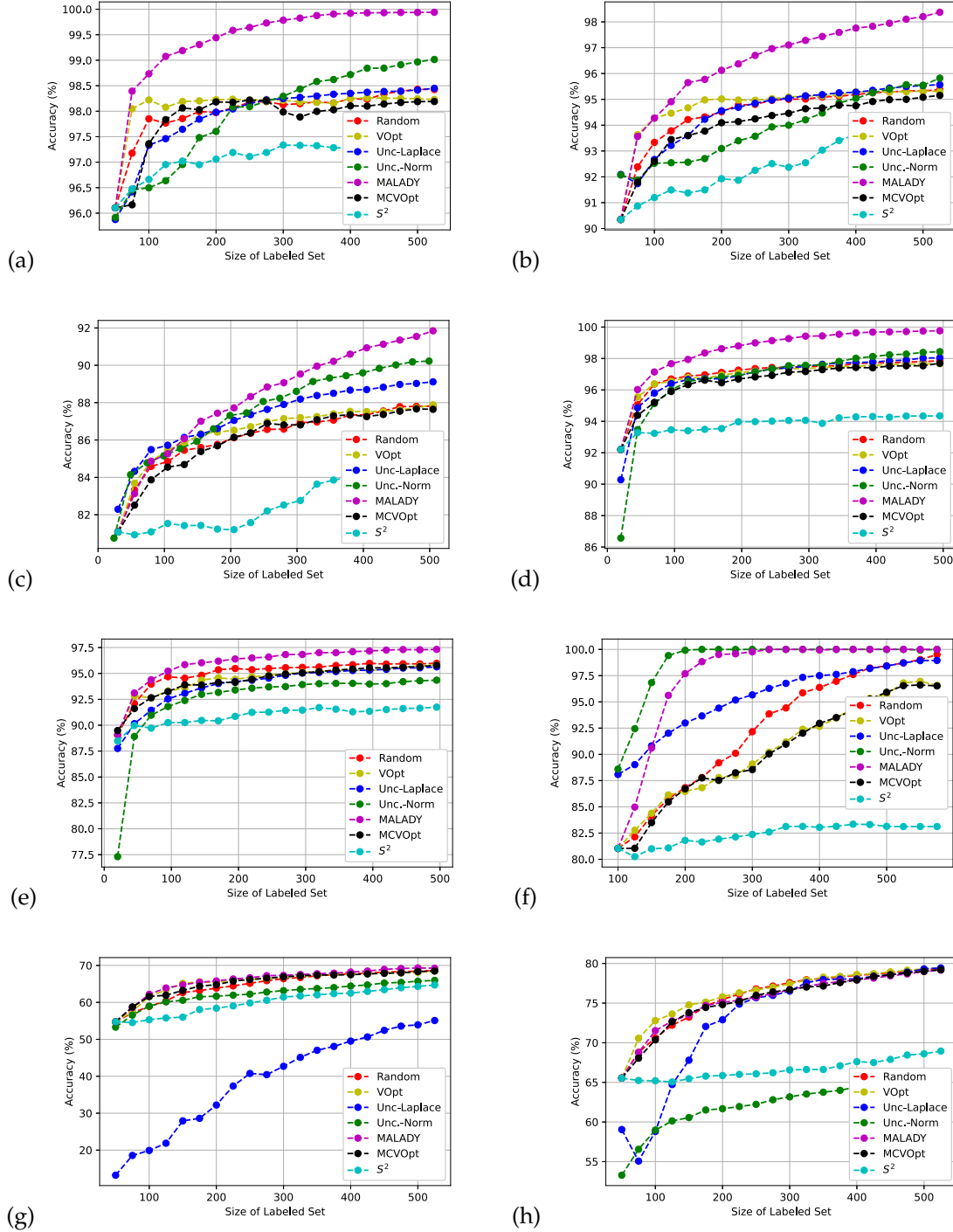


Fig. 4: Accuracy v.s. # of labeled points for each dataset: (a) Opt-Digits (b) USPS (c) Satellite (d) Jasper Ridge (e) Urban (f) COIL-20 (g) CIFAR-10 (h) FASHIONMNIST. To allow for a more fair comparison of each acquisition function's selections for labeled sets, we report accuracy results for Random, VOpt, MCVOpt, and S^2 using Algorithm 1 since these acquisition functions are not tied to a particular graph-based semi-supervised classifier. All other methods display accuracy results from their corresponding graph-based SSL classifiers.

tivariate normal distribution with mean corresponding to the harmonic solution across the field. The nodes are selected for querying such that the total variance of the distribution on the unlabeled data, as well as the expected prediction, is minimized. The MCVOpt (*Model Change plus VOpt*) acquisition function [68] is a heuristic combination of Model Change [6, 7] and VOpt [5] acqui-

sition functions. The Model Change [6] acquisition function computes the change in a soft-constraint Laplace learning classifier that a currently unlabeled point could induce as a result of labeling it; this is approximated with a spectral truncation (low-rank approximation) of the graph Laplacian to improve efficiency.

In minimum norm uncertainty sampling [4], the authors design an acquisition function that measures uncertainty in the Poisson reweighted Laplace learning algorithm (PWLL). In particular, the authors control the exploration versus exploitation tradeoff in the active learning process by introducing a diagonal perturbation in PWLL which produces exponential localization of solutions. The uncertainty sampling-based method [1] queries points near decision boundaries that are most uncertain. Common uncertainty measures include least confidence, smallest margin, and entropy. For comparison, we use the smallest margin acquisition function with Laplace learning as an underlying classifier, which prioritizes points near the decision boundaries. It should be noted that our proposed acquisition function for MALADY can be thought of as a form of uncertainty sampling where the quantification of uncertainty is derived from the auction dynamics graph-based classifier (Algorithm 1). Finally, the S^2 [66] algorithm queries the label of the vertex that bisects the shortest-shortest path between any pair of oppositely labeled vertices.

For the implementation of MALADY, VOpt, MCVOpt, Random, and Unc.-Laplace, we utilized the GraphLearning Python package [19] which contains both graph learning and active learning frameworks. For the implementation of Unc.-Norm [4] and S^2 , we utilize the code obtained from the author's repository.

C. Hyperparameters Selection

In this section, we outline the selected hyperparameters for our experiments. In particular, the number of nearest neighbors, N_n , for the weight matrix is a tunable parameter that can be adjusted to control the sparsity of the graph. For the computational experiments, we set the number of steps $s = 100$, auction error tolerance $\epsilon_{min} = 1e^{-6}$, epsilon scaling factor $\alpha = 4$, and the initial epsilon $\epsilon_0 = 1e^{-7}$. We also incorporate class size constraints for the classification, which are controlled by the upper bound U and lower bound B . In our computational experiments, we use exact class size constraints. See the Supplementary Information file for a comprehensive outline of all parameters used in the experiments, along with a discussion of their respective roles.

D. Ablation Studies

In order to evaluate the effectiveness and robustness of our method, we perform ablation studies on the role of the class size constraint, the initial labeled set, and the proposed acquisition function. Firstly, to examine the performance of our method with various class size constraints, we run the experiments in two datasets using various bounds. In particular, we test USPS and Opt-Digits with various class size constraints and show the result in Figure (5). We observe that our method performs better with exact constraints, especially in a low labeled rate. Second, to assess the effectiveness of

the auction dynamics algorithm, we ablate on the semi-supervised learning (SSL) classifier for output prediction accuracy on unlabeled sets. Specifically, we use our MALADY framework to sample the labeled set and tested different SSL classifiers to evaluate model accuracy. The results, presented in Figure 6, compare the performance of Multiclass MBO [30], Poisson learning [19], Laplace learning [55], and Poisson-reweighted Laplace learning (PWLL) [67]. The results in Figure 6 show that our proposed acquisition function in MALADY achieves optimal performance when paired with the auction dynamics algorithm (Algorithm 1) as the underlying SSL classifier. We also conduct additional ablation studies to understand how our framework performs with varying numbers of initially labeled samples and different distributions. Due to page limitations, however, we provide this ablation study's results in the Supporting Information file.

E. Performance and Discussion

Information about the data sets used for our computational experiments is detailed in Supporting Information file. For each data set, we use five initial labels per class; thus, the initial labeled set has a total of $5K$ labeled points, where K is the number of classes. We sequentially query 500 additional data elements to label. Moreover, for the hyperspectral data, we use the cosine similarity function (2) for graph weight computation. For others, we use the Gaussian weight function (1).

In Figure 4, we show the accuracy performance of each acquisition function averaged over 10 trials; for all data sets, accuracy was the main evaluation metric. From the results, one can see that our MALADY method has significantly outperformed the comparison methods in all data sets except COIL-20. For the COIL-20 data set, our MALADY algorithm performs similarly to the best-performing method Unc.-Norm [4], which uses a different underlying graph-based classifier than our model.

The significant difference in performance of our proposed MALADY method for active learning over other uncertainty-based acquisition functions (e.g., Unc.-Norm and Unc.-Laplace) warrants some further discussion. The success of uncertainty sampling active learning methods most often depends on the capability of the underlying classifier to use small amounts of labeled data to produce reasonable decision boundaries that can be highlighted by the corresponding acquisition function. Recall that our proposed acquisition function utilizes the auction values (i.e., coefficients, prices, and incentives) computed by the auction dynamics classifier (Algorithm 1) to identify the boundaries between clusters containing oppositely labeled nodes. Both the extensive empirical results (Figure 4) and the intuitive toy example visualized (Figure 2) suggest that these auction values are quite useful for accurately modeling the geometry of the clusters, especially these corresponding decision

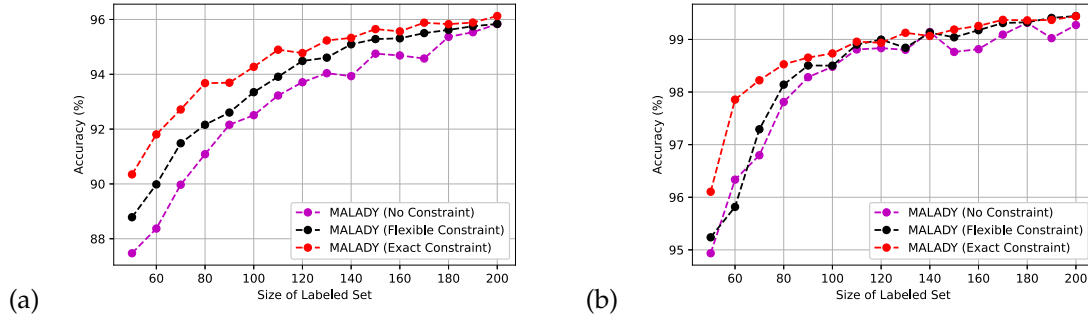


Fig. 5: MALADY under various class size constraints. The red curve represents the exact class size constraint, the black curve represents the flexible constraint, and the pink curve represents no class size constraint. (a): USPS data set (b): Opt-Digits data set.

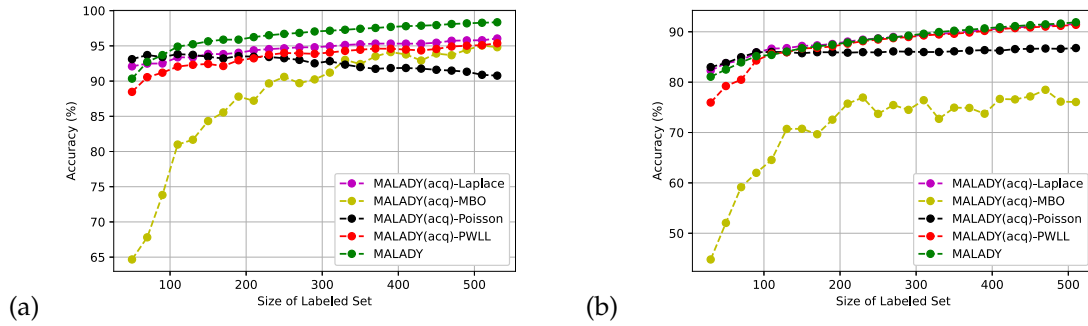


Fig. 6: Performance of different SSL classifiers on labeled set sampled using our MALADY framework. (a): USPS data set (b): Landsat data set.

boundaries even with low amounts of labeled data. While a rigorous theoretical analysis of the mechanism of Algorithm 1 that influences such accurate modeling of clustering geometry is outside the scope of the current work, we humbly suggest that this graph-based auction dynamics procedure is a key distinguishing component of MALADY that allows for superior uncertainty sampling performance.

V. CONCLUSION

In this paper, we introduce the Multiclass Active Learning with Auction Dynamics on Graphs (MALADY) method which integrates active learning with auction dynamics techniques for semi-supervised learning and classification and ensures exploitation in the active learning process. The proposed method allows one to obtain accurate results even in the case of very small labeled sets, a common scenario for many applications since it can be costly in time and money to obtain labeled data in many applications. The proposed method leverages a novel acquisition function derived from the dual variables in an auction algorithm. Lastly, the proposed algorithm also allows one to incorporate class size constraints for the classification task, which improves accuracy even further. Overall, the proposed procedure is a powerful approach for an important machine learning task.

ACKNOWLEDGEMENTS

This work is supported by NSF grant DMS-2052983.

REFERENCES

- [1] B. Settles, *Active Learning*. Morgan & Claypool Publishers LLC, 2012, vol. 6, no. 1.
- [2] A. L. Bertozzi, X. Luo, A. M. Stuart, and K. C. Zygalakis, "Uncertainty quantification in graph-based classification of high dimensional data," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 6, no. 2, pp. 568–595, 2018.
- [3] Y.-L. Qiao, C. X. Shi, C. Wang, H. Li, M. Haberland, X. Luo, A. M. Stuart, and A. L. Bertozzi, "Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos," *Image Processing: Algorithms and Systems*, 2019.
- [4] K. Miller and J. Calder, "Poisson reweighted Laplacian uncertainty sampling for graph-based active learning," *SIAM Journal on Mathematics of Data Science*, vol. 5, no. 4, pp. 1160–1190, 2023.
- [5] M. Ji and J. Han, in *Artificial Intelligence and Statistics*, 2012, pp. 556–564.
- [6] K. S. Miller and A. L. Bertozzi, "Model change active learning in graph-based semi-supervised learning," *Communications on Applied Mathematics and Computation*, pp. 1–29, 2024.
- [7] K. Miller, "Active learning and uncertainty in graph-based semi-supervised learning," Ph.D. dissertation, UCLA, 2022.
- [8] Y. Ma, R. Garnett, and J. Schneider, " Σ -optimality

- for active learning on Gaussian random fields," in *Advances in Neural Information Processing Systems* 26. Curran Associates, Inc., 2013, pp. 2751–2759.
- [9] S. Dasgupta and D. Hsu, "Hierarchical sampling for active learning," in *Proceedings of the 25th International Conference on Machine Learning*. Association for Computing Machinery, 2008, pp. 208–215.
- [10] A. Cloninger and H. N. Mhaskar, "Cautious active clustering," *Applied and Computational Harmonic Analysis*, vol. 54, pp. 44–74, 2021.
- [11] G. Dasarathy, R. Nowak, and X. Zhu, "S2: An efficient graph based active learning algorithm with application to nonparametric classification," in *Conference on Learning Theory*, vol. 40, 2015, pp. 503–522.
- [12] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," *International Conference on Learning Representations*, 2020.
- [13] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proceedings of the International Conference on Machine Learning*, 2017, pp. 1183–1192.
- [14] A. Kirsch, J. van Amersfoort, and Y. Gal, "Batch-BALD: Efficient and diverse batch acquisition for deep bayesian active learning," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [15] D. Kushnir and L. Venturi, "Diffusion-based sampling for deep active learning," *International Conference on Sampling Theory and Applications*, pp. 1–9, 2023.
- [16] A. Kirsch, S. Farquhar, P. Atighehchian, A. Jesson, F. Branchaud-Charron, and Y. Gal, "Stochastic batch acquisition: A simple baseline for deep active learning," *Transactions on Machine Learning Research*, 2023.
- [17] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *International Conference on Learning Representations*, 2018.
- [18] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-based semi-supervised learning: A comprehensive review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8174–8194, 2022.
- [19] J. Calder, B. Cook, M. Thorpe, and D. Slepcev, "Poisson learning: Graph based semi-supervised learning at very low label rates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1306–1316.
- [20] J. Calder, D. Slepcev, and M. Thorpe, "Rates of convergence for Laplacian semi-supervised learning with low labeling rates," *Research in the Mathematical Sciences*, vol. 10, no. 1, p. 10, 2023.
- [21] G. Bhusal, E. Merkurjev, and G.-W. Wei, "Persistent Laplacian-enhanced algorithm for scarcely labeled data classification," *Machine Learning*, vol. 113, no. 10, pp. 7267–7292, 2024.
- [22] D. P. Bertsekas, "A distributed algorithm for the assignment problem," *Laboratory for Information and Decision Systems Working Paper*, MIT, 1979.
- [23] D. Bertsekas, "Linear network optimization," 1991.
- [24] D. P. Bertsekas and D. A. Castanon, "The auction algorithm for the transportation problem," *Annals of Operations Research*, vol. 20, no. 1, pp. 67–96, 1989.
- [25] D. Bertsekas, *Network optimization: continuous and discrete models*. Athena Scientific, 1998, vol. 8.
- [26] M. Jacobs, E. Merkurjev, and S. Esedoğlu, "Auction dynamics: A volume constrained MBO scheme," *Journal of Computational Physics*, vol. 354, pp. 288–310, 2018.
- [27] E. Merkurjev, "A fast graph-based data classification method with applications to 3D sensory data in the form of point clouds," *Pattern Recognition Letters*, vol. 136, pp. 154–160, 2020.
- [28] B. Merriman, J. K. Bence, and S. Osher, "Diffusion generated motion by mean curvature," in *Proceedings of the Computational Crystal Growers Workshop*, 1992, pp. 73–83.
- [29] E. Merkurjev, T. Kostic, and A. L. Bertozzi, "An MBO scheme on graphs for classification and image processing," *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 1903–1930, 2013.
- [30] C. Garcia-Cardona, E. Merkurjev, A. Bertozzi, A. Flenner, and A. Percus, "Multiclass data segmentation using diffuse interface methods on graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1600–1613, 2014.
- [31] E. Merkurjev, C. Garcia-Cardona, A. Bertozzi, A. Flenner, and A. Percus, "Diffuse interface methods for multiclass segmentation of high-dimensional data," *Applied Mathematics Letters*, vol. 33, pp. 29–34, 2014.
- [32] E. Merkurjev, A. L. Bertozzi, and F. Chung, "A semi-supervised heat kernel pagerank MBO algorithm for data classification," *Communications in Mathematical Sciences*, vol. 16, no. 5, pp. 1241–1265, 2018.
- [33] E. Merkurjev, J. Sunu, and A. Bertozzi, "Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video," in *IEEE International Conference on Image Processing*, 2014, pp. 689–693.
- [34] A. Bertozzi and E. Merkurjev, "Graph-based optimization approaches for machine learning, uncertainty quantification and networks," in *Handbook of Numerical Analysis*, 2019, vol. 20, pp. 503–531.
- [35] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.
- [36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [38] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, "Permutation equivariant graph framelets for heterophilous graph learning," *IEEE Transactions on*

- Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11 634–11 648, 2024.
- [39] L. Bai, L. Cui, Y. Wang, M. Li, J. Li, S. Y. Philip, and E. R. Hancock, "HAQJSK: Hierarchical-aligned quantum jensen-shannon kernels for graph classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6370–6384, 2024.
- [40] X. Han, Z. Jiang, N. Liu, and X. Hu, "G-mixup: Graph data augmentation for graph classification," in *International Conference on Machine Learning*. PMLR, 2022, pp. 8230–8248.
- [41] B. Chen, K. Miller, A. L. Bertozzi, and J. Schwenk, "Batch active learning for multispectral and hyperspectral image segmentation using similarity graphs," *Communications on Applied Mathematics and Computation*, vol. 6, no. 2, pp. 1013–1033, 2024.
- [42] B. Chen, Y. Lou, A. L. Bertozzi, and J. Chanussot, "Graph-based active learning for nearly blind hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.
- [43] K. Miller, J. Mauro, J. Setiadi, X. Baca, Z. Shi, J. Calder, and A. L. Bertozzi, "Graph-based active learning for semi-supervised classification of sar data," in *Algorithms for Synthetic Aperture Radar Imagery XXIX*, vol. 12095. SPIE, 2022, pp. 126–139.
- [44] J. Chapman, B. Chen, Z. Tan, J. Calder, K. Miller, and A. Bertozzi, "Novel batch active learning approach and its application on the synthetic aperture radar datasets," in *Proceedings of Society of Photo-Optical Instrumentation Engineers Conference on Defense + Commercial Sensing*. SPIE, 2023.
- [45] J. M. Murphy and M. Maggioni, "Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1829–1845, 2019.
- [46] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Medical Image Analysis*, vol. 71, p. 102062, 2021.
- [47] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral image classification with convolutional neural network and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.
- [48] Y. Ge, D. Yang, and A. L. Bertozzi, "Iterative active learning strategies for subgraph matching," *Pattern Recognition*, vol. 158, p. 110797, 2025.
- [49] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Database Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.
- [50] "ANNOY library," <https://github.com/spotify/annoy>, 2024.
- [51] D. P. Bertsekas, D. A. Castanon, and H. Tsaknakis, "Reverse auction and the solution of inequality constrained assignment problems," *SIAM Journal on Optimization*, vol. 3, no. 2, pp. 268–297, 1993.
- [52] S. Esedoglu and M. Jacobs, "Convolution kernels and stability of threshold dynamics methods," *SIAM Journal on Numerical Analysis*, vol. 55, no. 5, pp. 2123–2150, 2017.
- [53] H. Hu, T. Laurent, M. A. Porter, and A. L. Bertozzi, "A method based on total variation for network modularity optimization using the MBO scheme," *SIAM Journal on Applied Mathematics*, vol. 73, no. 6, pp. 2224–2246, 2013.
- [54] E. Bae and E. Merkurjev, "Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds," *Journal of Mathematical Imaging and Vision*, vol. 58, pp. 468–493, 2017.
- [55] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the International Conference on Machine Learning*, 2003, pp. 912–919.
- [56] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *International Conference on Machine Learning workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.
- [57] "Landsat data set," <https://archive.ics.uci.edu/dataset/146/statlog+landsat+satellite>, 2024.
- [58] "USPS data," <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>, 2024.
- [59] "Jasper data set," <https://lesun.weebly.com/hyperspectral-data-set.html>, 2024.
- [60] "Urban data set," <https://erdclibrary.erdclibrary.dren.mil/jspui/handle/11681/2925>, 2024.
- [61] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.
- [62] COIL-20, *Technical report CUCS-005-96*, 1996.
- [63] OptDigits, "Optical Recognition of Handwritten Digits," UCI Machine Learning Repository, 1998, DOI: <https://doi.org/10.24432/C50P49>.
- [64] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.
- [65] K. S. Miller and A. L. Bertozzi, "Model change active learning in graph-based semi-supervised learning," *Communications on Applied Mathematics and Computation*, vol. 6, no. 2, pp. 1270–1298, 2024.
- [66] G. Dasarathy, R. Nowak, and X. Zhu, "S2: An efficient graph based active learning algorithm with application to nonparametric classification," in *Conference on Learning Theory*, 2015, pp. 503–522.
- [67] J. Calder and D. Slepčev, "Properly-weighted graph Laplacian for semi-supervised learning," *Applied Mathematics & Optimization*, vol. 82, pp. 1111–1159, 2020.
- [68] K. Miller, J. Mauro, J. Setiadi, X. Baca, Z. Shi, J. Calder, and A. Bertozzi, "Graph-based active learning for semi-supervised classification of sar data," in *2022 SPIE Conference on Defense + Commercial Sensing*, 2022.

APPENDIX

In this section, we provide the algorithm for upper and lower bound auctions (Algorithms 3 and 4, respectively) and the membership auction (Algorithm 5). The upper and lower bound auction algorithms are important for integrating the class size information. In algorithm 1, once we calculate the assignment problem coefficients, we pass them onto the upper bound auction algorithm (3) to satisfy the upper bound constraint, and the result is fed into the lower bound algorithm (4) to satisfy the lower bound constraint.

Algorithm 3: Lower Bound Auction (LBA)[26]

Input: $\epsilon > 0$, bounds B, U , coefficients a , initial prices p^0 , initial admissible incentives t^0 , complete (but possibly lower infeasible) ϵ -CS matching \mathcal{X}^0
Initialization: Set $d = p^0 - t^0$, set $\mathcal{X} = \mathcal{X}^0$;
Result: complete and feasible ϵ -CS matching and admissible prices and admissible incentives (\mathcal{X}, p, t) .

while there exists some i with $(|\mathcal{X}_i| < U_i$ and $d_i > 0)$ or $(|\mathcal{X}_i| < B_i)$ **do**
 for each i^* with $(|\mathcal{X}_{i^*}| < U_{i^*}$ and $d_{i^*} > 0)$ or $(|\mathcal{X}_{i^*}| < B_{i^*})$ **do**
 for each $x \notin \mathcal{X}_{i^*}$ **do**
 Let j be x 's current phase;
 Calculate $\Delta(x) = (a_j(x) - d_j) - (a_{i^*}(x) - d_{i^*})$;
 end
 while $(|\mathcal{X}_{i^*}| < U_{i^*}$ and $d_{i^*} > 0)$ or $(|\mathcal{X}_{i^*}| < B_{i^*})$ **do**
 Find $x \in \arg \min_{y \notin \mathcal{X}_{i^*}} \Delta(y)$;
 if $|\mathcal{X}_{i^*}| < B_{i^*}$ **then**
 Remove x from its current phase and add x to \mathcal{X}_{i^*} ;
 if $|\mathcal{X}_{i^*}| = B_{i^*}$ and $\Delta(x) \geq 0$ **then**
 Subtract $\Delta(x) + \epsilon$ from d_{i^*} ;
 end
 else
 if $\Delta(x) + \epsilon \geq d_{i^*}$ **then**
 Set $d_{i^*} = 0$;
 else
 Remove x from its current phase and add x to \mathcal{X}_{i^*} ;
 if $|\mathcal{X}_{i^*}| = U_{i^*}$ and $\Delta(x) \geq 0$ **then**
 Subtract $\Delta(x) + \epsilon$ from d_{i^*} ;
 end
 end
 end
 end
end
Set $p = \max(d, 0)$, set $t = \max(-d, 0)$;
return (\mathcal{X}, p, t)

Algorithm 4: Upper Bound Auction (UBA) [26]

Input: $\epsilon > 0$, bounds B, U , coefficients a , initial prices p^0 and incentives t^0 , data $x \in \mathcal{X}$
Result: Prices p , incentives t , and complete ϵ -CS matching \mathcal{X} satisfying upper bounds.
Initialization: All x unassigned, $d = p^0 - t^0$, $\mathcal{X} = \emptyset$
while some x is marked as unassigned **do**
 for each unassigned $x \in \mathcal{X}$ **do**
 Calculate $i_{cs}(x, p)$; choose $i^* \in i_{cs}(x, d)$;
 $b(x) = d_{i^*} + \epsilon + (a_{i^*}(x) - d_{i^*}) - (a_{i_{next}}(x) - d_{i_{next}})$;
 if $|\mathcal{X}_{i^*}| = U_{i^*}$ **then**
 Find $y = \arg \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 Remove y from \mathcal{X}_{i^*} and add x to \mathcal{X}_{i^*} ;
 Mark y as unassigned, x as assigned;
 Set $d_{i^*} = \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 else if $|\mathcal{X}_i| = B_i$ and $d_i < 0$ **then**
 Find $y = \arg \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 Remove y from \mathcal{X}_{i^*} and add x to \mathcal{X}_{i^*} ;
 Mark y as unassigned, x as assigned;
 Set $d_{i^*} = \min(\min_{z \in \mathcal{X}_{i^*}} b(z), 0)$;
 else
 Mark x as assigned and add x to \mathcal{X}_{i^*} ;
 end
 end
end
Set $p = \max(d, 0)$, set $t = \max(-d, 0)$;
return (\mathcal{X}, p, t)

Algorithm 5: Membership Auction [24, 26]

Input: $\epsilon > 0$, volumes V , coefficients a , initial prices p^0 , data $x \in \mathcal{X}$
Result: Final p , complete ϵ -CS matching (\mathcal{X}, p) .
Initialization: For every $i \in \{1, \dots, N\}$ mark all x as unassigned, set $p = p^0$, set $\mathcal{X} = \emptyset$;
while some x is marked as unassigned **do**
 for each unassigned $x \in \mathcal{X}$ **do**
 Calculate $i_{cs}(x, p)$; choose $i^* \in i_{cs}(x, p)$;
 $b(x) = p_{i^*} + \epsilon + (a_{i^*}(x) - p_{i^*}) - (a_{i_{next}}(x) - p_{i_{next}})$;
 if $|\mathcal{X}_{i^*}| = V_{i^*}$ **then**
 Find $y = \arg \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 Remove y from \mathcal{X}_{i^*} and add x to \mathcal{X}_{i^*} ;
 Mark y as unassigned, x as assigned;
 Set $p_{i^*} = \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 else
 Mark x as assigned and add x to \mathcal{X}_{i^*} ;
 if $|\mathcal{X}_i| = V_i$ **then**
 Set $p_{i^*} = \min_{z \in \mathcal{X}_{i^*}} b(z)$;
 end
 end
 end
end
return (\mathcal{X}, p)
