



Batch Active Learning for Multispectral and Hyperspectral Image Segmentation Using Similarity Graphs

Bohan Chen¹ · Kevin Miller² · Andrea L. Bertozzi¹ · Jon Schwenk³

Received: 27 November 2022 / Revised: 11 April 2023 / Accepted: 4 May 2023 /

Published online: 20 July 2023

© The Author(s) 2023

Abstract

Graph learning, when used as a semi-supervised learning (SSL) method, performs well for classification tasks with a low label rate. We provide a graph-based batch active learning pipeline for pixel/patch neighborhood multi- or hyperspectral image segmentation. Our batch active learning approach selects a collection of unlabeled pixels that satisfy a graph local maximum constraint for the active learning acquisition function that determines the relative importance of each pixel to the classification. This work builds on recent advances in the design of novel active learning acquisition functions (e.g., the Model Change approach in arXiv:2110.07739) while adding important further developments including patch-neighborhood image analysis and batch active learning methods to further increase the accuracy and greatly increase the computational efficiency of these methods. In addition to improvements in the accuracy, our approach can greatly reduce the number of labeled pixels needed to achieve the same level of the accuracy based on randomly selected labeled pixels.

Keywords Image segmentation · Graph learning · Batch active learning · Hyperspectral image

Mathematics Subject Classification 68T05 · 68T45 · 94A08

✉ Bohan Chen
bhchenyz@g.ucla.edu

Kevin Miller
ksmiller@utexas.edu

Andrea L. Bertozzi
bertozzi@math.ucla.edu

Jon Schwenk
jschwenk@lanl.gov

¹ Department of Mathematics, University of California, Los Angeles, 520 Portola Plaza, Los Angeles 90095, CA, USA

² Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, 201 E 24th St, Austin 78712, TX, USA

³ Los Alamos National Laboratory, Los Alamos, NM 87545, USA

1 Introduction

Image segmentation is a basic problem in the field of machine learning and computer vision. One older approach involves partial differential equation (PDE)-based methods, which segment an image by solving a PDE on the image numerically, based on minimization of an energy functional [9, 12, 23, 34]. More recently, graph-based methods have also been developed for both semi-supervised and unsupervised learning on image processing [3, 7, 8, 14, 17, 18, 20, 27, 29, 30]. Another common choice is the neural network methods, including convolutional neural networks (CNN) [35] and graph convolutional networks (GCN) [41, 43], with trainable convolutional filters optimized by minimizing the difference between predicted and ground-truth labels.

We employ a graph-based learning method where the feature vectors of each pixel in an image are used to construct a graph whose edge weights are determined by feature vector similarity [6]. This approach has been proven successful in noisy image recovery [29], studies using remotely sensed images to combine LIDAR and optical images [21], and blind hyperspectral unmixing [37]. Graph learning is an approach that trains a classifier by minimizing a graph-based energy function to directly identify a function on the nodes of the graph. This is different from graph neural networks that train a convolutional kernel and evolve the corresponding convolutional operator on the graph.

Active learning is a branch of machine learning that judiciously selects a limited number of unlabeled data to query for labels, with the aim of maximally improving the underlying classifier's performance [39]. An acquisition function is used to quantify which data would be useful to label from the set of available unlabeled data. Active learning can significantly improve classifier performance at very low label rates and minimize the cost of labeling data by domain experts [15, 31, 33, 39].

Traditional active learning selects labeled data sequentially, i.e., in each step, only the global maximum of the acquisition function is selected. Batch active learning selects a query set of multiple points in each step of the active learning process. Batch active learning provides new challenges compared to sequential active learning. Selecting data with similar information is redundant and does not fully utilize the acquisition function. Some prior methods for batch active learning imitate sequential active learning by selecting the batch through a greedy sequential process [11, 22, 25] or segment the candidate set into several small subsets and select the batch samples as the collection of the maximum points of each small subset [16, 24]. We introduce a novel batch active learning approach called LocalMax, to select a collection of unlabeled data that satisfy the graph local maximum condition in each step of the active learning process. The original version of LocalMax was developed for the synthetic aperture radar (SAR) image classification tasks [13]. Compared with other batch active learning approaches, LocalMax is more efficient while having almost identical performance as sequential active learning.

The novelty of this paper lies in the batch graph-based active learning pipeline for image segmentation tasks with very low label rates. This pipeline is specifically designed for the hyperspectral pixels classification task. With fewer than 0.5% labeled pixels selected by the active learning process, the graph learning classifier achieves similar accuracy as that with 10% randomly sampled labeled pixels.

Active learning has recently been shown to produce excellent results for hyperspectral image segmentation, especially when combined with similarity graphs and a novel Model-Change acquisition function [31]. However, such methods require sequential updates that involve the recalculation of the segmentation problem with each active learning step. While

these steps are computationally efficient, there is a significant gain in efficiency to be made by developing a batch process for active learning. In the model-change method, a strategic choice of data point to be labeled is made at each sequential step, and that choice does not naturally extend to a batch process. In particular, if one uses a sequential acquisition function for a batch process, the points to be labeled will not be optimal (e.g., they could be chosen from the same area). We propose a novel efficient batch active learning method based on a LocalMax condition. This was initially developed for SAR image classification [13], and here we further develop it for hyperspectral pixel classification. In this paper, while expanding the application of LocalMax on the image segmentation task, we accelerate the LocalMax algorithm and provide a more detailed analysis of its computational complexity.

2 Background for Graph-Based Active Learning Model

In this section, we review basic graph learning and some active learning techniques applied to graph learning classifiers. We construct a similarity graph via a K -nearest neighbors approach [1]. We apply graph Laplace learning [45] with some labeled nodes to classify unlabeled nodes. The labeled nodes are selected through the active learning process.

2.1 Graph Construction

We generate a graph based on the dataset $X = \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^d$ of d -dimensional feature vectors. X is indexed by the index set $Z = \{1, 2, \dots, N\}$. Consider the graph $G(X, W)$ with the vertex (node) set X and the edge weight matrix $W \in \mathbb{R}^{N \times N}$, where W_{ij} denotes the edge weight between vertices $i \neq j$. The weight W_{ij} is chosen to be proportional to the similarity between corresponding feature vectors x_i and x_j . In our model, we choose

$$W_{ij} = \exp\left(-\frac{\angle(x_i, x_j)^2}{\sqrt{\tau_i \tau_j}}\right), \quad (1)$$

where $\angle(x_i, x_j) = \arccos\left(\frac{x_i^\top x_j}{\|x_i\| \|x_j\|}\right)$ is the angle between feature vectors x_i and x_j . The normalization constant τ_i is chosen according to the similarity to the K th nearest neighbor of i (i.e., $\tau_i = \angle(x_i, x_{i_K})$, where x_{i_K} is the K th nearest neighbor of x_i).

To improve the computational efficiency, we require the $N \times N$ weight matrix W to be sparse. For each vertex x_i , we only consider edges between x_i and its K -nearest neighbors (KNN) according to the angle similarity stated above. This can be done by an approximate nearest neighbor search algorithm [1]. Let x_{i_k} , $k = 1, 2, \dots, K$ be the K -nearest neighbors of x_i (including x_i itself) according to the angle similarity. Define a sparse weight matrix by

$$\bar{W}_{ij} = \begin{cases} W_{ij}, & j = i_1, i_2, \dots, i_K, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

For practical purposes, K is chosen as small as possible while ensuring the connectivity of the corresponding graph G . The connectivity property is required when calculating the acquisition functions (Sect. 2.3). We symmetrize the sparse weight matrix to obtain our final weight matrix by redefining $W_{ij} := (\bar{W}_{ij} + \bar{W}_{ji})/2$. Note that W is sparse, symmetric, and non-negative (i.e., $W_{ij} \geq 0$).

2.2 Graph Learning

With a graph $G(X, W)$ constructed as described in the previous section, we now describe a graph-based approach for semi-supervised learning (SSL) and present previous work in this field. Assume we have some observations of the ground-truth labels on a subset of vertices $Z_0 \subset Z$. Let $y^\dagger: Z_0 \rightarrow \{1, 2, \dots, n_c\}$ be the ground-truth labeling function that maps each index $j \in Z_0$ to exactly one class label $y_j^\dagger = y^\dagger(j) \in \{1, 2, \dots, n_c\}$. The corresponding one-hot encoding mapping is $\mathbf{y}^\dagger: Z_0 \rightarrow \{e_1, e_2, \dots, e_{n_c}\}$ defined by $\mathbf{y}^\dagger(j) = e_{y^\dagger(j)}$, where e_k is the k th standard basis vector with all zeros except a 1 at the i th entry. The goal for the semi-supervised learning task is to predict the labels of the *unlabeled* vertices $x_i \in X, i \in Z - Z_0$.

Important geometric information about the dataset X is encoded in graph Laplacian matrices [2, 42] defined on G . Define $d_j = \sum_{k \in Z} W_{jk}$ to be the degree of node j and let D be the diagonal matrix with diagonal entries d_1, d_2, \dots, d_N . While there are various graph Laplacians one could define [42], we use the unnormalized graph Laplacian matrix $L_u = D - W$ in this paper.

The inferred classification of unlabeled vertices comes from thresholding a continuous-valued node function $\mathbf{u}: Z \rightarrow \mathbb{R}^{n_c}$. In particular, the predicted label of $x_i \in X$ is $y_i = \arg \max \{u_1(i), u_2(i), \dots, u_{n_c}(i)\}$, where $u_k(i)$ is the k th entry of $\mathbf{u}(i)$. Consider an $N \times n_c$ matrix U , whose i th row is $\mathbf{u}(i)$; that is, each node function \mathbf{u} can be identified by a matrix U whose i th represents the output of \mathbf{u} at node i . The graph-based SSL model that we consider obtains an optimal \hat{U} (i.e., optimal node function $\hat{\mathbf{u}}$) by solving an optimization problem of the form

$$\begin{aligned} \hat{U} &= \arg \min_u J_\ell(U, \mathbf{y}^\dagger) \\ &= \arg \min_{U \in \mathbb{R}^{N \times n_c}} \frac{1}{2} \langle U, LU \rangle_F + \sum_{j \in Z_0} \ell(\mathbf{u}(j), \mathbf{y}^\dagger(j)), \end{aligned} \quad (3)$$

where $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product for matrices. The loss function $\ell: \mathbb{R}^{n_c} \times \mathbb{R}^{n_c} \rightarrow \mathbb{R}$ measures the difference between the prediction $\mathbf{u}(i)$ and the ground-truth $\mathbf{y}^\dagger(i)$ for i in the observation set Z_0 . While there are several choices for the loss function, we simply apply a hard-constraint penalty

$$\ell_h(x, y) = \begin{cases} +\infty, & \text{if } x \neq y, \\ 0, & \text{if } x = y. \end{cases} \quad (4)$$

This hard-constraint penalty function ℓ_h forces the minimizer \hat{U} to be exactly the same as the ground-truth \mathbf{y}^\dagger on the observation set Z_0 . This SSL scheme was introduced in [45] and we refer to it as Laplace learning. We can reorder the vertices to be able to write $U = \begin{bmatrix} U_l \\ U_u \end{bmatrix}$, where U_l corresponds to the submatrix of U whose rows are in the labeled (observed) index set Z_0 and U_u similarly corresponds to the submatrix of U whose rows are in $Z - Z_0$ (i.e., the unlabeled index set). Likewise, we can split the weight matrix W and degree matrix D into labeled and unlabeled submatrices as

$$W = \begin{bmatrix} W_{ll} & W_{lu} \\ W_{ul} & W_{uu} \end{bmatrix}, \quad D = \begin{bmatrix} D_{ll} & D_{lu} \\ D_{ul} & D_{uu} \end{bmatrix}. \quad (5)$$

As a result of the hard-constraint labeling of Laplace learning, \hat{U}_l is fixed as the one-hot encodings of the observations on the labeled set Z_0 . According to [45], the optimizer \hat{U}_u of Laplace learning can be calculated explicitly as

$$\hat{U}_u = (D_{uu} - W_{uu})^{-1} W_{ul} \hat{U}_l. \quad (6)$$

The Laplace learning gives a harmonic solution $\hat{\mathbf{u}}$ on the graph G . It infers the sum-to-one property of the graph Laplace learning output node function $\hat{\mathbf{u}}$. If the ground truth labels are given in one-hot forms, for any node $i \in Z$, we have $\hat{u}_k(i) \geq 0$, $k = 1, 2, \dots, n_c$ and $\sum_{k=1}^{n_c} \hat{u}_k(i) = 1$, where $\hat{\mathbf{u}}(i) = (\hat{u}_1(i), \hat{u}_2(i), \dots, \hat{u}_{n_c}(i))$. With this property, at node $i \in Z$, $u_k(i)$ can be treated as the predicted probability that node i belongs to the class k .

There are various other graph SSL schemes based on the optimization problem (3). The main difference between them and the Laplace learning scheme is the choice of penalty function ℓ . In this paper, we use the multiclass Gaussian regression (MGR) model [4, 26] which applies an L_2 -norm penalty function $\ell_\gamma(x, y) = \frac{1}{2\gamma^2} \|x - y\|_2^2$. The MGR model is an approximation of the graph Laplace learning model in the sense that $\gamma \rightarrow \infty$.

Denote by $\mathcal{G}(N)$ the computation cost of a Laplace learning process on graph $G = (X, W)$ with the labeled set Z_0 . Assume the graph is constructed with the KNN sparse similarity matrix W (Sect. 2.1) and that the size of labeled set is much smaller than the number of nodes, i.e., $|Z_0| \ll |Z|$. Recall the computational complexity of the conjugate gradient method to solve the linear equation $Ax = b$ (Chapter 10 of [40]) is $O(m\sqrt{\kappa})$, where m and κ are the number of non-zero entries and the condition number of matrix A , respectively. If we solve Eq. (6) by the conjugate gradient method, we have $\mathcal{G}(N) = O(KN\sqrt{\kappa_L})$, where κ_L is the condition number of the graph Laplacian L .

2.3 Active Learning: Review of Acquisition Functions

Active learning improves the performance of the underlying SSL methods by carefully selecting unlabeled points to hand label via the use of an oracle or human in the loop. The aim of active learning is to identify which unlabeled inputs ($x_i \in X$ with $i \in Z - Z_0$) for which it would be the “most helpful” to have a human in the loop observe and obtain labels. The core of active learning is the *acquisition function* $\mathcal{A}: Z - Z_0 \rightarrow \mathbb{R}$, which evaluates the benefit of obtaining the label of each unlabeled datapoint. The *query set* $\mathcal{Q} \subset Z - Z_0$ of unlabeled points that are to be labeled is chosen via the optimization of an acquisition function. Note that in this work, we use Laplace learning [45] as the underlying semi-supervised classifier. Figure 1 is the flowchart of our active learning process based on the graph Laplace learning classifier. The acquisition functions we introduce are designed for the graph learning classifier, including the Uncertainty (UC) [5, 32, 36], Model-Change (MC) [31, 32], Variance Minimization (VOpt) [22], and Model-Change Variance Optimal (MCVOpt) acquisition functions [33].

The UC acquisition function \mathcal{A}_{UC} quantifies the uncertainty of the classifier \mathbf{u} on each unlabeled node [5, 32, 36] by the current classifier’s output value for that unlabeled node, (i) . Uncertainty sampling thus prioritizes querying points that are close to the current classifier’s decision boundaries. Various methods can be applied to quantify the uncertainty based on $\hat{\mathbf{u}}$. Here we consider the smallest-margin uncertainty acquisition function:

$$\mathcal{A}_{\text{UC}}(i) = 1 - \left(u_{k_0}(i) - \max_{k=1,2,\dots,n_c, k \neq k_0} u_k(i) \right), \quad (7)$$

where $i \in Z$, $k_0 = \arg \max_{j=1,2,\dots,n_c} u_j(x)$.

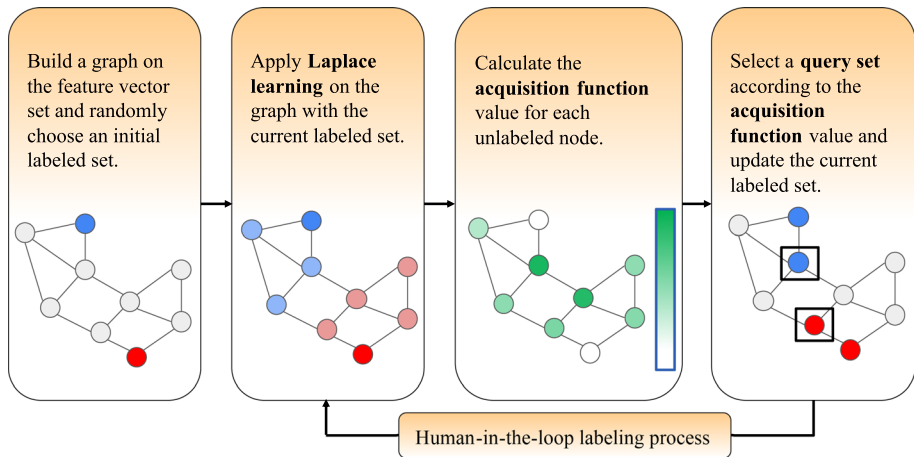


Fig. 1 Flowchart of the active learning process. The active learning loop is based on a fixed graph. In each step, we apply Laplace learning on the graph and update the labeled set with a query set selected based on the current acquisition function values. It should be noticed that it might need the human-in-the-loop process to obtain the label of the selected query set in each step of the active learning process

We need some prior knowledge before introducing the formula for the VOpt, MC, and MCVOpt acquisition functions. $L = L_u$ is a semi-positive definite matrix. By adjusting the number of nearest neighbors K considered at each vertex, we can guarantee that the graph G is connected. Further, the corresponding Laplacian matrix L has exactly one zero eigenvalue. We may order the eigenvalues of L as $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N$, and then consider the truncated decomposition of L with the smallest $M < N$ eigenvalues as $\hat{L} = V\Lambda V^T$, where $\Lambda \in \mathbb{R}^{M \times M}$ is a diagonal matrix with diagonal entries $\lambda_1, \lambda_2, \dots, \lambda_M$ and $V = [v^1, v^2, \dots, v^M] \in \mathbb{R}^{N \times M}$ is the matrix of corresponding eigenvectors. Furthermore, we define v_k as the k th column of V^T , $A = V^T U \in \mathbb{R}^{M \times n_c}$ and $\hat{A} = V^T \hat{U} \in \mathbb{R}^{M \times n_c}$, where $\hat{U} = \arg \min_{U \in \mathbb{R}^{N \times n_c}} \tilde{J}_\ell(A, \mathbf{y}^\dagger)$.

In addition, in the interest of the numerical stability and similar to [33], we replace the hard-constraint penalty function ℓ_h with the MGR penalty function $\ell_\gamma(x, y) = \frac{1}{2\gamma^2} \|x - y\|_2^2$ for MC acquisition function calculations (but not for the underlying SSL model). The MGR penalty ℓ_γ is a numerically stable perturbation of the hard-constraint penalty function ℓ_h . When $\gamma \rightarrow 0^+$, $\ell_\gamma \rightarrow \ell_h$. Then we define the MGR correlation matrix by

$$C_{\text{MGR}} = \left(\Lambda + V^T \left(\frac{1}{\gamma^2} P^T P \right) V \right)^{-1}, \quad (8)$$

where $P \in \mathbb{R}^{|Z_0| \times N}$ is a projection matrix onto the indices corresponding to the labeled set Z_0 . When the graph G is connected, the matrix C_{MGR} is guaranteed to exist, i.e., the matrix $(\Lambda + V^T (\frac{1}{\gamma^2} P^T P) V)$ is invertible.

The VOpt acquisition function $\mathcal{A}_{\text{VOpt}}$ is developed to minimize the expected error of the prediction results [22]. If we acquire the label of the unlabeled node $i \in Z \setminus Z_0$ and use labels of $Z \cup \{i\}$ to process the graph learning, then the expected prediction error on the set $Z \setminus (Z_0 \cup \{i\})$ can be computed as follows:

$$\mathbb{E} \left(\sum_{i \in Z \setminus (Z_0 \cup \{k\})} \|\mathbf{u}(i) - \mathbf{y}_i^\dagger\|^2 \right) = \text{Tr}(L_k^{-1}), \quad (9)$$

where L_k^{-1} is the submatrix of the graph Laplacian L with both row and column indices $Z \setminus (Z_0 \cup \{k\})$. Approximating the matrix L_k^{-1} by the truncated decomposition, we have the VOpt acquisition function

$$\mathcal{A}_{\text{VOpt}}(k) = \frac{1}{\gamma^2 + \mathbf{v}_k^\top C_{\text{MGR}} \mathbf{v}_k} \|C_{\text{MGR}} \mathbf{v}_k\|_2^2. \quad (10)$$

The MC and MCVOpt acquisition functions [31–33] were developed based on the look-ahead model with the objective energy

$$J_\ell^{k, \hat{\mathbf{y}}_k}(U, \mathbf{y}^\dagger; \hat{\mathbf{y}}_k) = \frac{1}{2} \langle U, LU \rangle_F + \sum_{i \in Z_0} \ell(\mathbf{u}(i), \mathbf{y}^\dagger(i)) + \ell(\mathbf{u}(k), \hat{\mathbf{y}}_k), \quad (11)$$

where $\hat{\mathbf{y}}_k$ is the one-hot pseudo-label for the *currently unlabeled* node $k \in Z - Z_0$. Practically, $\hat{\mathbf{y}}_k$ is the one-hot thresholding vector of $\hat{\mathbf{u}}(k)$ (the k th column of \hat{U}). Let $\hat{U}^{k, \hat{\mathbf{y}}_k} = \arg \min J_\ell^{k, \hat{\mathbf{y}}_k}(U, \mathbf{y}^\dagger; \hat{\mathbf{y}}_k)$. With the spectral truncation, the MC acquisition function is given by

$$\begin{aligned} \mathcal{A}_{\text{MC}}(k) &= \|\hat{U}^{k, \hat{\mathbf{y}}_k} - \hat{U}\|_F \\ &= \frac{1}{\gamma^2 + \mathbf{v}_k^\top C_{\text{MGR}} \mathbf{v}_k} \|C_{\text{MGR}} \mathbf{v}_k\|_2 \|\hat{A}^\top \mathbf{v}_k - \hat{\mathbf{y}}_k\|_2. \end{aligned} \quad (12)$$

Similarly, the MCVOpt acquisition function can be written as

$$\mathcal{A}_{\text{MCVOpt}}(k) = \frac{1}{\gamma^2 + \mathbf{v}_k^\top C_{\text{MGR}} \mathbf{v}_k} \|C_{\text{MGR}} \mathbf{v}_k\|_2^2 \|\hat{A}^\top \mathbf{v}_k - \hat{\mathbf{y}}_k\|_2. \quad (13)$$

3 Batch Active Learning Pipeline for Image Segmentation

This section introduces our pipeline with the graph Laplace learning classifier and batch active learning. Given an image for the segmentation task, we extract a feature vector for each pixel and construct a similarity graph based on the cosine similarity between these feature vectors according to Sect. 2.1. Then we apply the graph Laplace learning (Sect. 2.2) with labeled pixels selected by our batch active learning approach, LocalMax. The node classification on the graph gives a segmentation of the given image.

3.1 Image Segmentation Pipeline

We develop an image segmentation pipeline with graph learning and batch active learning approaches mentioned in Sects. 2 and 3.2. The first step for pixel classification is to associate each pixel with a feature vector. One can consider simply using the pixel values of all channels as the corresponding feature vector. In this case, the dimension of a feature vector is the same

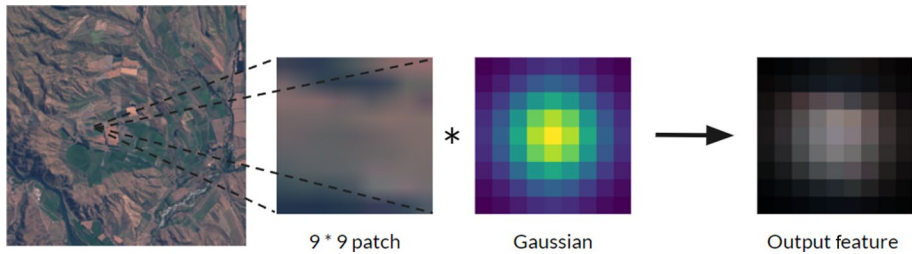


Fig. 2 The feature extraction process for a single pixel. The feature vector is a Gaussian-weighted patch centered on the pixel

as the number of channels of the image. While this simple construction is straightforward, it is useful to include neighborhood information of each pixel for feature extraction.

For pixel i , consider a $(2k + 1) \times (2k + 1)$ neighborhood patch P_i centered at pixel i . If pixel i is near the boundary of the image, apply reflection padding to expand the image before taking the neighborhood patch. Inspired by the non-local means method [10], we consider a $(2k + 1) \times (2k + 1)$ discrete Gaussian kernel G with $\sigma = k/2$. Specifically,

$$G(i, j) = \frac{\alpha}{2\pi\sigma^2} \exp\left(-\frac{(i - k - 1)^2 + (j - k - 1)^2}{2\sigma^2}\right),$$

where α is a constant such that $\sum_{i,j=1}^{2k+1} G(i, j) = 1$. The weighted patch is then defined by

$$P_i^w(i, j) = P_i(i, j)G(i, j)$$

for each pair of pixels $i, j = 1, 2, \dots, 2k + 1$. This feature process is illustrated by Fig. 2.

We apply this non-local means weighting process to each of the C channels in the image. Flattening these weighted patches and concatenating them together gives the non-local means feature vector for pixel i . The dimension of the resulting non-local means feature vector for a given pixel is $d = C(2k + 1)^2$.

For a given image, we extract each pixel's non-local means feature vector to get the feature vector set X ($|X| = N$). Then we build a similarity graph $G(X, W)$ based on the feature vector set S and with the sparse similarity weight matrix W via the KNN method based on these according to Sect. 2.1. For efficiency, we want to select a relatively small KNN parameter K while keeping the connectivity of the generated graph G . Throughout all of the experiments, we find that $K = 50$ is a reasonable value to ensure the connectivity and relative sparsity of the resulting similarity graph. In practice, we suggest that one can use a binary search to identify relatively small values of K that result in graph connectivity.

On the graph G , we randomly initialize a labeled node set and apply the LocalMax batch active learning to select a labeled set of nodes Z_0 according to Sects. 2.3 and 3.2. Finally, we predict the labels of unlabeled nodes $Z - Z_0$ on the graph G with the graph Laplace learning classifier based on the selected labeled node set Z_0 . This node classification on G gives a segmentation on the given image. The flowchart of our pipeline is Fig. 3.

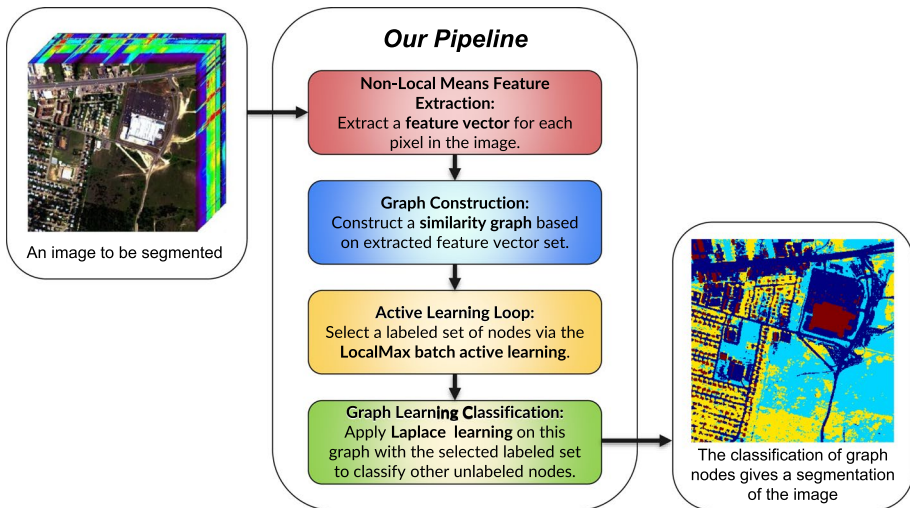


Fig. 3 Our graph-based active learning pipeline for the image segmentation task. Red box: feature extraction (Sect. 3.1); Blue box: Graph Construction (Sect. 2.1); Yellow box: Batch Active Learning (Sects. 2.3 and 3.2); Green box: Graph Learning (Sect. 2.2)

3.2 Batch Method

In the active learning process illustrated by Fig. 1, we select a query set according to a prescribed acquisition function \mathcal{A} . In the sequential active learning process, the query set $Q \subset Z - Z_0$ is selected by

$$Q = \{k\}, k = \arg \max_{k \in Z - Z_0} \mathcal{A}(k). \quad (14)$$

For batch active learning with the batch size B , simply selecting the top- B maximizers of the acquisition function likely includes nodes that are connected in the graph. As an inductive bias, the graph Laplace learning would have similar outputs with neighboring labeled nodes. Therefore, it is redundant to sample neighbors in the graph.

We propose a batch active learning method named LocalMax. This method was originally developed for the classification task of SAR datasets [13]. We define the local maximum of a certain node function $\mathcal{A}: Z \rightarrow \mathbb{R}$ on a KNN-generated graph G according to Definition 1.

Definition 1 (Local max of a graph node function) Consider a KNN-generated graph $G = (X, W)$, where X is the set of nodes indexed by Z and W is the edge weight matrix. For a graph node function $\mathcal{A}: Z \rightarrow \mathbb{R}$, $k \in Z$ is a *local maximum node* if and only if for any j , $\mathcal{A}(k) \geq \mathcal{A}(j)$, if there is an edge between x_j and x_k . Equivalently, $k \in Z$ is a local maximum if and only if

$$\mathcal{A}(k) \geq \mathcal{A}(j), \forall j \text{ s.t. } W_{jk} > 0. \quad (15)$$

The LocalMax batch active learning method selects the batch query set to be the top- B local maximums of the acquisition function \mathcal{A} in the graph G . Algorithm 1 shows the

process of the LocalMax batch active learning method. It should be noted that the batch size B can not be extremely large as there might not be enough local maximums of the discrete set of acquisition function values at a given iteration. Algorithm 1 has the maximal computational complexity $O(KN)$, where K is the KNN parameter (Sect. 2.1) and N is the number of nodes in the graph G . Usually $K \ll N$, the computational complexity is $O(N)$.

Algorithm 1 LocalMax Batch Active Learning

```

1: Input: A KNN graph  $G = (X, W)$ ,  $X$  is indexed by  $Z$ . A labeled index
   set  $Z_0$ . An acquisition function  $\mathcal{A}: Z - Z_0 \rightarrow \mathbb{R}^+$ . Batch size  $B$ .
2: Output: The query set  $\mathcal{Q}$ .
3: Extend the domain of  $\mathcal{A}$  to  $Z$  by defining  $\mathcal{A}(j) = 0, \forall j \in Z_0$ 
4:  $S \leftarrow Z - Z_0; \mathcal{Q} = \emptyset$ 
5: while  $S \neq \emptyset$  and  $|\mathcal{Q}| < B$  do
6:    $k \leftarrow \arg \max_{k \in S} \mathcal{A}(k)$ 
7:    $N(k) = \{i \in Z: W_{jk} > 0\}$ 
8:   if  $\mathcal{A}(k) \geq \mathcal{A}(i), \forall i \in N(k)$  then
9:      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{k\}$ 
10:  end if
11:   $S \leftarrow S - N(k)$ 
12: end while
  
```

This method has some important advantages. Practically, regions of high acquisition value will only have a small number of local maxima, so the LocalMax method obtains a batch of nodes from multiple regions of high acquisition. Due to the complicated structure of data, there are often many regions with high acquisition, so batches can be relatively large. This method also selects what the model predicts to be the most important point from the high-acquisition region.

According to Sect. 2.2, the computational cost of the graph Laplacian learning is $\mathcal{G}(N) = O(KN\sqrt{\kappa_L})$. If we want to sample a total number of M query nodes from the whole active learning process, the computational complexity of the sequential active learning process is $M\mathcal{G}(N)$ while the LocalMax batch active learning with the batch size B has the computational complexity $M/B[O(N) + \mathcal{G}(N)]$. This implies that the LocalMax batch active learning process is much more efficient than the sequential active learning, proportionally to the batch size B . This result is verified by experiments in Sect. 4.1.

4 Experiments and Results

This section shows the experiments and results of our graph-based active learning pipeline on the image segmentation tasks. We run two types of experiments. The first one compares LocalMax, the sequential active learning process, and the other two straightforward batch sampling approaches for active learning. The second one is the application of our pipeline on image segmentation tasks and the comparison with a similar semi-supervised approach proposed in [27]. Finally, we provide some comments about our experiment results.

We consider three datasets, Landsat-7, Urban, and Kennedy Space Center (KSC) in the following experiments. Here are brief introductions for these datasets.

- i. The Urban dataset was recorded in October 1995 by the Hyperspectral Digital Imagery Collection Experiment (HYDICE) over the urban area in Copperas Cove, TX, U.S. Zhu et al. [44] provided the ground truth labels with four classes: asphalt, grass, tree, and roof. This dataset contains a hyperspectral image with 307×307 pixels, each corresponding to a 2 square meters area. The raw image includes 210 channels, but we use the clean version with 162 channels after removing some channels due to dense water vapor and atmospheric effects. Figure 4 shows the raw image and ground truth labels of the Urban dataset.
- ii. The Landsat-7 dataset in this paper is a multispectral image of the Colville River (Alaska, USA) from the RiverPIXELS dataset [43], which provides paired Landsat and water-and-sediment labeled patches of size $256 \times 256 \times 6$, where the 6 multispectral channels correspond to Blue, Green, Red, Near IR, Shortwave IR 1, and Shortwave IR 2. Each pixel in the image covers roughly 900 m^2 . The RiverPIXELS provides ground-truth labels of this image with three classes, land, water, and bare sediment.
- iii. The Kennedy Space Center Dataset is a hyperspectral image at the Kennedy Space Center (KSC) in Florida, acquired by the NASA AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) instrument. This hyperspectral image has a size 512×614 . The raw image includes 224 channels, while we are using the clean version with 176 channels after removing water absorption and low SNR channels. There are 314×368 pixels in this dataset, while only 5211 (around 1.66%) of them have ground-truth labels. The ground-truth labels include 13 classes of different land coverings in this region.

We consider the overall accuracy (OA) to evaluate the performance of different methods. The definition of the OA is

$$\text{OA} = \frac{\text{Number of Correctly Classified Pixels}}{\text{Total Number of Pixels}}. \quad (16)$$

It should be noticed that, according to the KNN parameter selection process mentioned in Sect. 3.1, the number of neighborhoods selected to construct the graph is $K = 50$.

4.1 Comparison between LocalMax and Sequential Active Learning: Accuracy and Efficiency

In this section, we only conduct our experiments on the Urban dataset. We apply the hyperspectral pixel values as the feature vector, i.e., for each pixel, the feature vector corresponding to it is the vector of 162 channels. With four acquisition functions, we sample up to 134 pixels (0.15% of all pixels) with different active learning sampling methods. We initialize the labeled set with 10 random pixels in each class (in total 40 pixels) and sample extra 94 pixels according to the active learning methods. For a certain acquisition function \mathcal{A} , we consider four sampling methods: Sequential, Random, Top-Max, and LocalMax, the last three of which are batch active learning with a batch size B .

- **Sequential sampling** selects the global maximum node of \mathcal{A} to update the current labeled set. The query set $\mathcal{Q} = \{k^*\}$ and $k^* = \arg \max_{k \in \mathcal{Z} - \mathcal{Z}_0} \mathcal{A}(k)$.

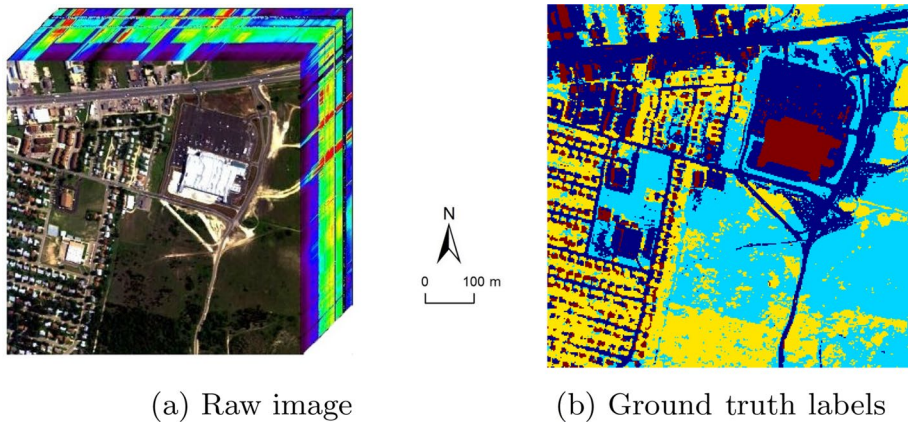


Fig. 4 Urban Dataset. **a** shows the raw hyperspectral image we used for experiments. **b** shows the ground-truth labels. Label information: asphalt (navy blue), grass (light blue), trees (yellow), roof (red)

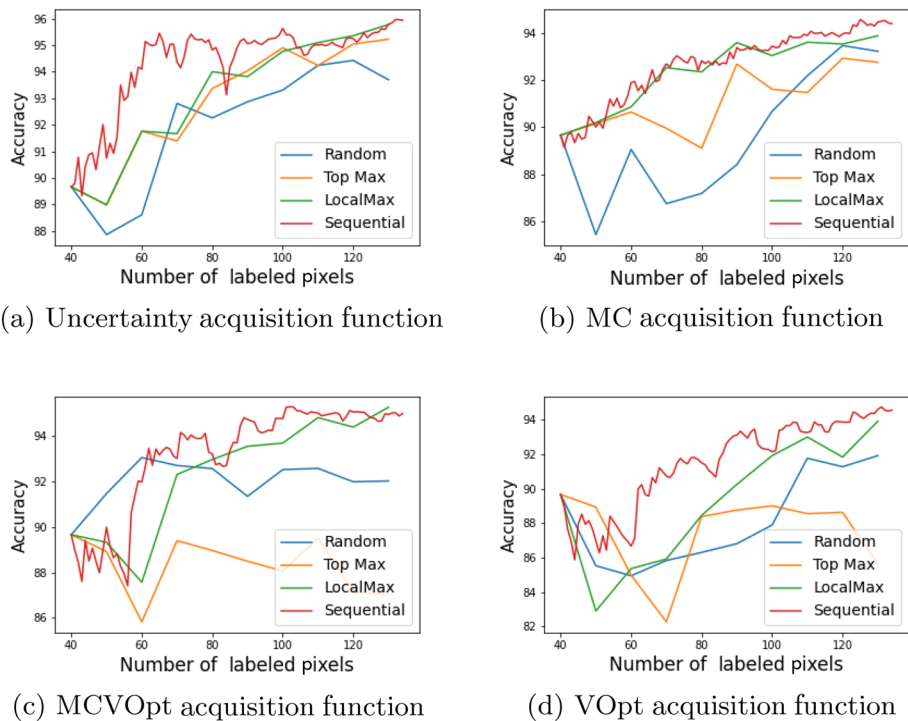


Fig. 5 Comparison between batch active learning methods and sequential active learning for four acquisition functions. Each panel includes four curves, of which the X-axis is the number of labeled pixels and the Y-axis is the accuracy. The blue, yellow, green, and red curves correspond to the Random, Top-Max, LocalMax, and Sequential sampling method, respectively, for the active learning process. More details on accuracy values and time consumption are shown in Table 1. Descriptions of each sampling method are in Sect. 4.1

Table 1 Comparison between different active learning sampling methods

| Label percentage | | | 0.1% | | 0.15% | |
|------------------|------------|-----|----------|--------------|----------|--------------|
| Acq | Sampling | B | Time(s) | Acc(%) | Time(s) | Acc(%) |
| UC | Sequential | 1 | 1 290.13 | 94.90 | 2 576.86 | 95.93 |
| | Random | 10 | 180.42 | 92.86 | 358.84 | 93.69 |
| | Top-Max | 10 | 169.59 | 94.03 | 343.24 | 95.22 |
| | LocalMax | 10 | 165.30 | 93.81 | 326.56 | 95.78 |
| MC | Sequential | 1 | 1 250.22 | 92.70 | 2 505.60 | 94.40 |
| | Random | 10 | 166.59 | 88.41 | 327.95 | 93.22 |
| | Top-Max | 10 | 165.07 | 92.68 | 327.35 | 92.71 |
| | LocalMax | 10 | 165.28 | 93.59 | 334.10 | 93.88 |
| MCVOpt | Sequential | 1 | 1 131.92 | 93.70 | 2 257.06 | 94.99 |
| | Random | 10 | 161.83 | 91.34 | 322.44 | 92.02 |
| | Top-Max | 10 | 160.61 | 88.49 | 323.99 | 87.02 |
| | LocalMax | 10 | 164.79 | 93.55 | 321.92 | 95.27 |
| MC | Sequential | 1 | 1 289.92 | 92.60 | 2 576.86 | 94.54 |
| | Random | 10 | 175.99 | 86.80 | 346.61 | 91.91 |
| | Top-Max | 10 | 169.91 | 88.75 | 339.96 | 85.62 |
| | LocalMax | 10 | 175.61 | 90.24 | 357.60 | 93.91 |

This table shows the efficiency and accuracy performance of active learning sampling methods with different acquisition functions. The first row “Acq” refers to the acquisition function. The second row “Sampling” refers to the choice of active learning sampling methods, including Sequential, Random, Top-Max, and LocalMax—the last three of which are batch active learning. The third row B is the batch size. We sample up to 0.1% and 0.15% labeled pixels in the Urban dataset and show the timings and accuracies. The top two accuracy values are bolded for each acquisition function. Descriptions of each sampling method are in Sect. 4.1

- **Random sampling** selects a batch of B unlabeled nodes according to the uniform distribution on the unlabeled node set $Z - Z_0$.
- **Top-Max sampling** selects a batch of B unlabeled nodes as the top- B maximum of \mathcal{A} , i.e., the query set $\mathcal{Q} = \{i_1, i_2, \dots, i_B\} \subset Z - Z_0$ where $i_1 = \arg \max_{i \in Z - Z_0} \mathcal{A}(i)$ and $i_b = \arg \max_{i \in Z - Z_0 - \{i_1, \dots, i_{b-1}\}} \mathcal{A}(i)$ for $b = 2, 3, \dots, B$.
- **LocalMax sampling** is the method we proposed in Sect. 3.2.

Figure 5 shows the curves between the accuracy and the number of labeled pixels for the four acquisitions, and Table 1 shows the time consumption and accuracy values for label rates 0.1%, 0.15%. From these experiments, we conclude the following.

- **Accuracy Performance** Sequential active learning has the best accuracy performance according to Fig. 5, which shows its higher accuracy for almost all numbers of labeled pixels. Our batch active learning method LocalMax has the second-best accuracy values and performs almost identically as the sequential one, especially for larger numbers of labeled pixels. This is also verified by Table 1 which shows in bold the top-2 accuracy values. LocalMax consistently shows accuracies in the top 2 and sometimes performs better than Sequential.

- ii. **Efficiency Performance** According to the timings in Table 1, LocalMax takes approximately the same time as the Random and Top-Max sampling methods while the Sequential active learning takes around eight times longer. The time multiplier 8 is close to the theoretical multiplier 10 (same as the batch size B) according to Sect. 3.2.

In summary, LocalMax batch active learning is much more efficient than Sequential active learning without significantly sacrificing accuracy.

4.2 Semi-supervised Image Segmentation with Low Label Rates

Here we perform segmentation experiments on all three datasets, Landsat-7, Urban, and KSC. The performances are evaluated on the OA as a function of the amount of labeled data used in training. Our method is compared with the graph-based semi-supervised method, abbr. **GL-SSL**, proposed in [27]. There are some differences between our graph learning method and the GL-SSL proposed in [27], the foremost being that our method uses the KNN approach to build a sparse similarity graph while the GL-SSL is based on a fully connected graph and uses the Nyström extension method to approximate the graph Laplacian matrix. In addition, our approach solves the optimal node function by minimizing the energy function of graph Laplace learning while GL-SSL is to minimize a regularized Ginzburg-Landau functional [17, 28, 29]. We have reimplemented the GL-SSL code¹ in Python and the corresponding results are based on our code. The parameters selected for GL-SSL are based on their recommendations with some fine-tuning. To avoid the bloated content, we only provide results of GL-SSL with the randomly selected training dataset. We also test the performance of GL-SSL with the training set selected by our active learning approach. Since the active learning process introduced in this paper is designed for graph Laplace learning, the OA performance of GL-SSL with an active learning training set is almost the same as that of a random training set. The OA result of our method or GL-SSL with the randomly sampled training dataset is the average OA value of 15 times random samples. Each time the random sampling process starts with 1 random pixel in each class, then randomly chooses the rest of the labeled pixels with equal probability.

Results of the Landsat-7 dataset are shown in Table 2 and Fig. 6. We use each pixel's non-local means feature vector to build the similarity graph. The neighborhood patch size is 7×7 , which leads to a 294 dimensional feature vector for each pixel. We sample up to 200 pixels (0.3% of all pixels) based on the random initialization of one pixel in each class (three pixels in total for the initialization) via the LocalMax batch active learning approach of the batch size 20. In addition, we sample up to 3 300 (around 5% of all pixels) labeled pixels based on the random initialization of 10 labeled pixels in each class (30 in total) and batch size 100. According to OA values shown in Table 2, the UC and MCVOpt acquisition functions reach a better accuracy with 0.3% labeled pixels than randomly selecting 5% labeled pixels. Figure 6 shows the segmentation result of our graph-based batch active learning method with UC acquisition functions.

Results of the **Urban dataset** are shown in Table 3 and Fig. 7. For these experiments, we use the hyperspectral pixel values as the feature vector for each pixel. We sample up to 286 pixels (0.3% of all pixels) based on the random initialization of one pixel in each class

¹ <http://www.ipol.im/pub/art/2017/204/>

Table 2 The OA of a landsat-7 multispectral image (65 536 pixels)

| Label rate (batch size) | | 0.1%(20) | 0.2%(20) | 0.3%(20) | 5%(100) |
|-------------------------|-----------|---------------|---------------|---------------|---------------|
| Method | Sampling | | | | |
| Ours | LM UC | 25.17% | 95.92% | 96.23% | 98.65% |
| Ours | LM MC | 91.75% | 95.04% | 95.80% | 97.50% |
| Ours | LM MCVOpt | 94.81% | 95.23% | 96.25% | 97.74% |
| Ours | LM VOpt | 93.61% | 94.4% | 94.78% | 96.44% |
| Ours | Random | 88.43% | 91.38% | 92.55% | 96.12% |
| GL-SSL [27] | Random | 20.77% | 41.44% | 48.56% | 94.94% |

This table shows the OA of our methods on the Landsat-7 multispectral image for different label rates. “Sampling” shows the training pixel selection method. The first four rows are the results of our LocalMax batch active learning method with different acquisition functions. Each process is initiated with 10 labeled pixels per class (30 in total). The OA results with the randomly sampled training dataset are the average value of 15 times random samples

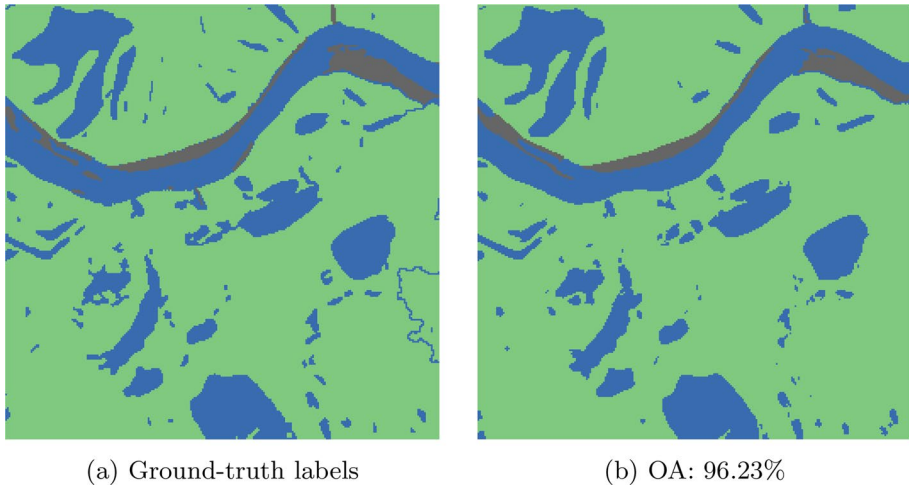


Fig. 6 The ground-truth and segmentation result of the Landsat-7 multispectral image from the RiverPIX-ELS dataset. **a** ground-truth labels; **b** segmentation result with 0.3% labeled pixels sampled according to LocalMax with a batch size of 20 and the UC acquisition function. The similarity graph is based on non-local means feature vectors with the neighborhood patch of size 7×7

(four pixels in total for the initialization) with LocalMax with the batch size 10. In addition, we sample 4 700 (around 5% of all pixels) with LocalMax with the batch size 100 based on the random initialization of 10 pixels in each class (40 in total). It can be seen that LocalMax batch active learning with the UC acquisition function attains an accuracy of 97.30% with only 0.3% labeled pixels, which is similar to the accuracy 97.76% with 10% randomly selected labeled pixels.

On the KSC dataset, we only consider and calculate our results on the 5 211 labeled pixels (around 1.66%) out of 314 368 total pixels. It is visualized in Fig. 8a. We segment this hyperspectral image into 13 classes according to ground-truth labels. We sample up to

Table 3 The OA of urban dataset (94 249 pixels)

| Label rate (batch size) | | 0.1%(10) | 0.2%(10) | 0.3%(10) | 5%(100) | 10% |
|-------------------------|-----------|---------------|---------------|---------------|---------------|---------------|
| Method | Sampling | | | | | |
| Ours | LM UC | 94.96% | 95.56% | 97.30% | 99.71% | N\A |
| Ours | LM MC | 90.71% | 93.31% | 94.84% | 98.60% | N\A |
| Ours | LM MCVOpt | 94.13% | 95.33% | 95.35% | 98.94% | N\A |
| Ours | LM VOpt | 91.32% | 94.27% | 94.52% | 97.44% | N\A |
| Ours | Random | 87.20% | 91.93% | 93.30% | 97.20% | 97.76% |
| GL-SSL [27] | Random | 53.06% | 55.08% | 58.79% | 90.71% | <u>93.48%</u> |

This table shows the OA of the Urban dataset among different methods and under different label rates. “Sampling” shows the training pixel selection method. The first four rows are the results of our LocalMax batch active learning method with different acquisition functions. Each process is initiated with 10 labeled pixels per class (30 in total). The OA results with the randomly sampled training dataset are the average value of 15 times random samples. The N\A in the column 10% means “Not Applicable”. The underlined value is copied from the paper [27]

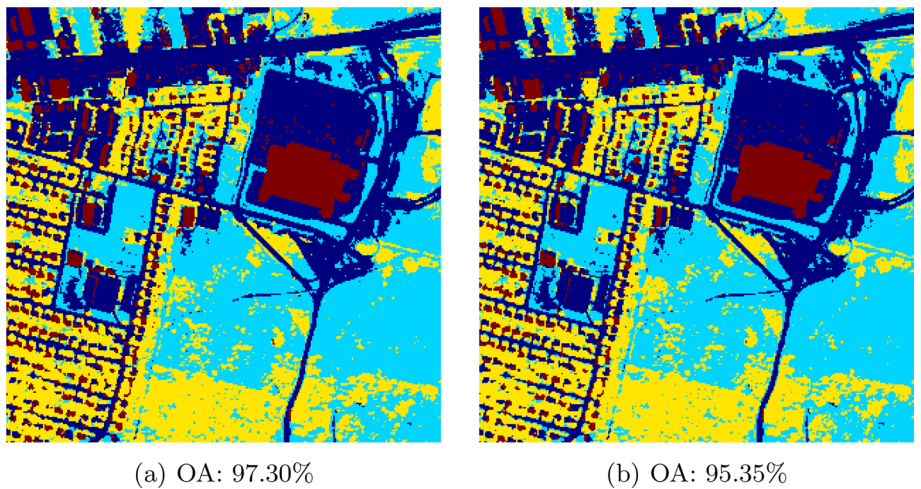


Fig. 7 The segmentation result of the Urban dataset with 0.3% labeled pixels sampled according to LocalMax batch active learning with the batch size 10. **a** UC acquisition function; **b** MCVOpt acquisition function. Label information: asphalt (navy blue), grass (light blue), trees (yellow), and roof (red). The ground-truth labels are in Fig. 4b

325 pixels (6% of all pixels with ground-truth labels) based on the random initialization of 1 pixel in each class (13 pixels in total for the initialization). Table 4 shows the OA of the KSC dataset. Our LocalMax batch active learning with the UC acquisition function performs best. Figure 8b shows the segmentation result of our graph-based batch active learning method with UC acquisition functions.

In summary, with a very low label rate (less than 0.5%), our method has a relatively good performance on the OA while GL-SSL does not perform well. Our graph-based active learning approach can significantly reduce the number of labeled pixels required

Table 4 The OA of KSC Dataset (5 122 pixels with ground-truth labels)

| Label rate (Batch size) | | 2.5%(5) | 4%(5) | 6%(5) |
|-------------------------|-----------------|---------------|---------------|---------------|
| Method | Sampling | | | |
| Ours | LocalMax UC | 85.47% | 88.66% | 89.83% |
| Ours | LocalMax MC | 82.04% | 85.86% | 87.85% |
| Ours | LocalMax MCVOpt | 85.12% | 85.86% | 87.13% |
| Ours | LocalMax VOpt | 83.22% | 84.68% | 86.66% |
| Ours | Random | 82.22% | 85.22% | 85.83% |
| GL-SSL [27] | Random | <u>80.37%</u> | 81.67% | 83.55% |

This table shows the OA of the KSC dataset among different methods and under different label rates. Accuracy values and the percentage of labeled pixels in this table are calculated based on 5 211 labeled pixels out of 314 368 total pixels. “Sampling” shows the training pixel selection method. The first four rows are the results of our LocalMax batch active learning method with different acquisition functions. All those sampling processes start with 1 labeled pixel in each class (13 in total). The OA results with the randomly sampled training dataset are the average value of 15 times random samples. The underlined value is copied from paper [27]

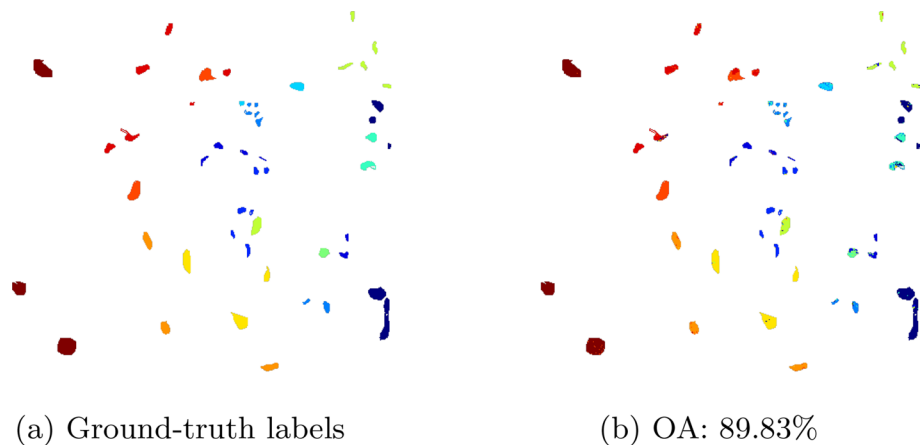


Fig. 8 The ground-truth and segmentation result of the KSC dataset. **a** ground-truth labels of 5 211 pixels, including 13 classes; **b** segmentation result with 6% labeled pixels sampled according to LocalMax batch active learning with the batch size 10 and the UC acquisition function

for the semi-supervised image segmentation task. According to Tables 2 and 3, the graph Laplace learning with 0.3% labeled pixels selected by active learning has a similar accuracy performance to that with 5% to 10% randomly selected labeled pixels.

4.3 Comments About Our Experiments

In this part, we comment on our experiments in Sects. 4.1 and 4.2. With a relatively small percentage of training data, the MCVOpt acquisition function performs similarly or even

better than the UC acquisition function. This is evident in Table 2. With 0.1% labeled data, the UC has only 25% accuracy, while other acquisition functions have more than 90%. When the percentage of training data increases, the UC acquisition function achieves superior OA. A reason for this phenomenon may lie in the difference between the tendencies of the different acquisition functions to exploration or exploitation in the active learning process. Roughly speaking, the UC acquisition function focuses on exploiting the classifier's current decision boundaries pursuant to the current labeled set by querying data points along said decision boundaries between different classes. The other three acquisition function, MC, VOpt, and MCVOpt, focus more on exploration than exploitation. They are developed to explore the geometric structure of the entire dataset. At the beginning of the active learning process, it is more important to explore the inherent geometric or clustering structure of the entire dataset. When the labeled percentage or the current accuracy is relatively high, exploiting the decision boundary can be more helpful for increasing the overall classification accuracy. As an empirical summary, we recommend using the UC acquisition function, but we caution that the UC acquisition function is not necessarily always the best choice due to its tendency to exploit during the active learning process.

According to Fig. 5, there is a gap between the accuracies of LocalMax and the sequential active learning for a small number of labeled nodes. The main reason for such an initial gap is that a newly labeled node can significantly change the acquisition function's value when the amount of labeled data is relatively small. In such a circumstance, LocalMax may not perform well since the next global maximum of the acquisition function might not be close to any local maximum of the current acquisition function. This gap does not necessarily remain throughout the active learning process, so the LocalMax batch sampling method can better approximate sequential sampling as the number of labeled data increases.

5 Conclusion

We propose a graph-based batch active learning pipeline for multi- and hyperspectral image segmentation. Our method showed excellent image segmentation skill using very low percentages of available training data. Compared with a similar graph-based image segmentation method proposed in [27], our method requires fewer labeled pixels to achieve better OA. This suggests that careful selection points to label through active learning is beneficial for this application of graph-based semi-supervised classification. In addition, we introduced a batch active learning approach, LocalMax, to select a batch of pixels in each step of the active learning process. According to our experiments, LocalMax batch active learning not only accelerates the process of sampling pixels but also retains similar accuracies as sequential active learning using the same acquisition function.

Acknowledgements Bohan Chen is supported by the UC-National Lab In-Residence Graduate Fellowship Grant L21GF3606. Kevin Miller was supported by a DOD National Defense Science and Engineering Graduate (NDSEG) Research Fellowship. Jon Schwenk is supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20170668PRD1 and 20210213ER. Andrea Bertozzi is supported by the NGA under Contract No. HM04762110003. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NGA. Approved for public release, NGA-U-2023-00757.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no relevant financial or non-financial interests, competing interests, affiliations, or involvements with any organization or entity related to the subject matter or materials discussed in this manuscript. They have transparently disclosed all funding sources and reported any affiliations or relationships that might appear to present a conflict.

Ethical Standard The authors diligently adhere to established ethical guidelines and principles throughout the research process. Their work complies with all applicable regulations and is conducted with integrity, professionalism, and a commitment to the responsible and ethical dissemination of knowledge.

Informed Consent The authors have designed their research methodology, data collection, and analysis procedures with utmost consideration for ethical concerns.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. ACM* **45**(6), 891–923 (1998). <https://doi.org/10.1145/293347.293348>
2. Belkin, M., Niyogi, P., Sindhvani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **7**, 2399–2434 (2006)
3. Bertozzi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Model. Simul.* **10**(3), 1090–1118 (2012)
4. Bertozzi, A.L., Hosseini, B., Li, H., Miller, K., Stuart, A.M.: Posterior consistency of semi-supervised regression on graphs. *Inverse Problems* **37**(10), 105011 (2021)
5. Bertozzi, A.L., Luo, X., Stuart, A.M., Zygalakis, K.C.: Uncertainty quantification in graph-based classification of high dimensional data. *SIAM/ASA J. Uncertain. Quantif.* **6**(2), 568–595 (2018)
6. Bertozzi, A.L., Merkurjev, E.: Graph-based optimization approaches for machine learning, uncertainty quantification and networks. In: *Processing, Analyzing and Learning of Images, Shapes, and Forms. Part 2*, 503–531, *Handb. Numer. Anal.*, 20, Elsevier/North-Holland, Amsterdam (2019)
7. Boyd, Z.M., Bae, E., Tai, X.-C., Bertozzi, A.L.: Simplified energy landscape for modularity using total variation. *SIAM J. Appl. Math.* **78**(5), 2439–2464 (2018)
8. Boyd, Z.M., Porter, M.A., Bertozzi, A.L.: Stochastic block models are a discrete surface tension. *J. Nonlinear Sci.* **30**(5), 2429–2462 (2020)
9. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imaging Vision* **28**(2), 151–167 (2007)
10. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 60–65. IEEE (2005)
11. Cai, W., Zhang, Y., Zhou, J.: Maximizing expected model change for active learning in regression. In: *2013 IEEE 13th International Conference on Data Mining*, pp. 51–60. IEEE (2013)
12. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
13. Chapman, J., Chen, B., Tan, Z., Calder, J., Miller, K., Bertozzi, A.L.: Novel batch active learning approach and its application on the synthetic aperture radar datasets. In: *SPIE Defense and Commercial Sensing: Algorithms for Synthetic Aperture Radar Imagery XXX* (2023)
14. Ciurte, A., Bresson, X., Cuisenaire, O., Houhou, N., Nedevski, S., Thiran, J.-P., Cuadra, M.B.: Semi-supervised segmentation of ultrasound images based on patch representation and continuous min cut. *PLoS ONE* **9**(7), e100972 (2014)
15. Dasgupta, S.: Two faces of active learning. *Theoret. Comput. Sci.* **412**(19), 1767–1781 (2011). <https://doi.org/10.1016/j.tcs.2010.12.054>

16. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: International Conference on Machine Learning, pp. 1183–1192. PMLR (2017)
17. Garcia-Cardona, C., Merkurjev, E., Bertozzi, A.L., Flenner, A., Percus, A.G.: Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1600–1613 (2014)
18. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. *Multiscale Model. Simul.* **7**(3), 1005–1028 (2009). <https://doi.org/10.1137/070698592>
19. Hu, H., Sunu, J., Bertozzi, A.L.: Multi-class graph Mumford-Shah model for plume detection using the MBO scheme. In: Proceedings of the EMMCVPR Conference in Hong Kong. **8932**, 209–222. Tai, X.-C. et al. (Eds), Springer Lecture Notes in Computer Science (2015)
20. Hu, H., Laurent, T., Porter, M.A., Bertozzi, A.L.: A method based on total variation for network modularity optimization using the MBO scheme. *SIAM J. Appl. Math.* **73**(6), 2224–2246 (2013)
21. Iyer, G., Chanussot, J., Bertozzi, A.L.: A graph-based approach for data fusion and segmentation of multimodal images. *IEEE Trans. Geosci. Remote Sensing* **59**(5), 4419–4429 (2021). <https://doi.org/10.1109/TGRS.2020.2971395>
22. Ji, M., Han, J.: A variance minimization criterion to active learning on graphs. In: Artificial Intelligence and Statistics, pp. 556–564. PMLR (2012)
23. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comp. Vision* **1**, 321–331 (2004)
24. Kushnir, D., Venturi, L.: Diffusion-based deep active learning. [arXiv:2003.10339](https://arxiv.org/abs/2003.10339) (2020)
25. Ma, Y., Garnett, R., Schneider, J.G.: Sigma-optimality for active learning on Gaussian random fields. In: NIPS, pp. 2751–2759 (2013)
26. Ma, Y., Huang, T.-K., Schneider, J.G.: Active search and bandits on graphs using sigma-optimality. In: UAI, vol. 542, pp. 551 (2015)
27. Meng, Z., Merkurjev, E., Koniges, A., Bertozzi, A.L.: Hyperspectral image classification using graph clustering methods. *IPOL J. Image Process. Online* **7**, 218–245 (2017). <https://doi.org/10.5201/ipol.2017.204>
28. Merkurjev, E., Garcia-Cardona, C., Bertozzi, A.L., Flenner, A., Percus, A.G.: Diffuse interface methods for multiclass segmentation of high-dimensional data. *Appl. Math. Lett.* **33**, 29–34 (2014)
29. Merkurjev, E., Kostić, T., Bertozzi, A.L.: An MBO scheme on graphs for classification and image processing. *SIAM J. Imaging Sci.* **6**(4), 1903–1930 (2013)
30. Merkurjev, E., Sunu, J., Bertozzi, A.L.: Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 689–693. IEEE (2014)
31. Miller, K., Bertozzi, A.L.: Model-change active learning in graph-based semi-supervised learning. <https://doi.org/10.48550/arXiv.2110.07739> (2021)
32. Miller, K., Li, H., Bertozzi, A.L.: Efficient graph-based active learning with probit likelihood via Gaussian approximations. [arXiv:2007.11126](https://arxiv.org/abs/2007.11126) (2020).
33. Miller, K., Mauro, J., Setiadi, J., Baca, X., Shi, Z., Calder, J., Bertozzi, A.L.: Graph-based active learning for semi-supervised classification of SAR data. [arXiv:2204.00005](https://arxiv.org/abs/2204.00005) (2022)
34. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* **42**(5), 577–685 (1989). <https://doi.org/10.1002/cpa.3160420503>
35. O’Shea, K., Nash, R.: An introduction to convolutional neural networks. [arXiv:1511.08458](https://arxiv.org/abs/1511.08458) (2015)
36. Qiao, Y., Shi, C., Wang, C., Li, H., Haberland, M., Luo, X., Stuart, A.M., Bertozzi, A.L.: Uncertainty quantification for semi-supervised multi-class classification in image processing and ego-motion analysis of body-worn videos. *Electron. Imaging* **31**(11), 1–264 (2019)
37. Qin, J., Lee, H., Chi, J.T., Drumetz, L., Chanussot, J., Lou, Y., Bertozzi, A.L.: Blind hyperspectral unmixing based on graph total variation regularization. *IEEE Trans. Geosci. Remote Sensing* **59**(4), 3338–3351 (2021). <https://doi.org/10.1109/TGRS.2020.3020810>
38. Schwenk, J., Rowland, J.: RiverPIXELS: paired Landsat images and expert-labeled sediment and water pixels for a selection of rivers v1.0. United States. <https://data.ess-dive.lbl.gov/view/>, <https://doi.org/10.15485/1865732>
39. Settles, B.: Active Learning vol. 6, pp. 1–114. Morgan & Claypool Publishers LLC, Carnegie Mellon University, USA (2012). <https://doi.org/10.2200/s00429ed1v01y201207aim018>
40. Shewchuk, J.R.: An introduction to the conjugate gradient method without the agonizing pain. Carnegie-Mellon University, Pittsburgh, PA (1994)
41. Thorpe, M., Nguyen, T.M., Xia, H., Strohmer, T., Bertozzi, A., Osher, S., Wang, B.: Grand++: graph neural diffusion with a source term. In: International Conference on Learning Representations (2021)
42. Von Luxburg, U.: A tutorial on spectral clustering. *Stat. Comput.* **17**(4), 395–416 (2007)
43. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning, pp. 6861–6871. PMLR (2019)

44. Zhu, F., Wang, Y., Xiang, S., Fan, B., Pan, C.: Structured sparse method for hyperspectral unmixing. *ISPRS-J. Photogramm. Remote Sens.* **88**, 101–118 (2014)
45. Zhu, X., Ghahramani, Z., Lafferty, J.D.: Semi-supervised learning using Gaussian fields and harmonic functions. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 912–919 (2003)