FERMAT ACTIVE LAPLACE LEARNING FOR SEMI-SUPERVISED HYPERSPECTRAL IMAGE CLASSIFICATION

Vutichart Buranasiri, James M. Murphy

Tufts University, Department of Mathematics, Medford, MA 02155, USA

ABSTRACT

Two active learning algorithms for hyperspectral image (HSI) classification are proposed that combine density-aware Fermat distances with Poisson-reweighted harmonic label propagation. Our methods actively query points using an uncertainty-based acquisition function, extending *Poisson ReWeighted Laplace Learning* (PWLL). Our first algorithm, *Fermat Active Laplace Learning* (FALL), builds an affinity matrix using Fermat distances between all data points. Then, PWLL is run with a diagonal perturbation using the minimum-norm acquisition function. In contrast, *Approximate FALL* (A-FALL) computes Fermat distances between each data point and landmark pixels selected via farthest-point sampling and constructs the affinity matrix using landmark multidimensional scaling. After several query rounds, A-FALL selects the Fermat exponent p using a leave-one-out cross-validation variant. FALL and A-FALL leverage Fermat distances and subsequent harmonic label propagation to provide a density-aware estimation of the data manifold, improving labeling accuracy. Experiments on Salinas A and Pavia show the effectiveness of FALL and the scalability of A-FALL to large HSI scenes.

***Index Terms*—** Active learning, semi-supervised learning, hyperspectral image classification, Fermat distance

1. INTRODUCTION

Machine learning has led to major advances in remote sensing [1, 2, 3]; however, many modern approaches require a large number of labeled training samples. While recent supervised deep learning architectures can achieve superior labeling accuracy on benchmark hyperspectral images (HSI) [4], they are often impractical when access to training data is limited. This has motivated semi-supervised methods that require only a small subset of labeled data for HSI classification [5]. In particular, graph-based semi-supervised methods such as Laplace learning [6] and Poisson learning [7] have shown strong performance on benchmark high-dimensional datasets.

Recently, the *active learning* [8, 9, 10, 11, 12] regime has shown promise for HSI segmentation [13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. Starting from an unlabeled dataset, active learning iteratively queries points for labels to select the most informative pixels. There is a vast literature on query selection [26]. Broadly, such methods can be categorized into those that prioritize exploring unknown regions of the dataset and those that exploit the downstream classifier's decision boundaries [12].

The Poisson ReWeighted Laplace Learning (PWLL) algorithm combines an uncertainty sampling querying method with a variant of the properly weighted Laplace Learning classifier, showing strong results on certain datasets [27]. A key algorithmic choice in graph-based methods is the construction of edge weights. While the RBF kernel is commonly used, our approach considers a family of weighted shortest-path metrics known as Fermat distances [28, 29, 30, 31, 32, 33, 34], which use powers of Euclidean distances as weights. Fermat and other shortest-path distances have been used in unsupervised and semi-supervised machine learning problems [35, 36, 37, 38, 34, 39].

Motivated by these developments, we propose *Fermat Active Laplace Learning* (FALL) and *Approximate FALL* (A-FALL). FALL extends PWLL by using an affinity matrix built from Fermat distances over all data points. In contrast, A-FALL computes Fermat distances over m landmark pixels and uses landmark multi-dimensional scaling (LMDS) [40] to compute approximate weights for downstream PWLL classification. Additionally, a variant of leave-one-out cross-validation (LOO-CV) [41, 42] is used in A-FALL to score a fixed set of candidate p-values. Experiments on Salinas A and Pavia show the effectiveness of FALL and the computational efficiency of A-FALL in the low-label rate regime.

The rest of this paper is organized as follows. In Section 2, we review relevant background material. We outline Fermat exponent learning algorithms in Section 3, proposed algorithms in Section 4, and experimental results in Section 5. Code is available at: https://github.com/vburan01/FALL.

2. BACKGROUND

2.1. Fermat Distances

Let $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^D$ denote a HSI with D spectral bands, with pixels $\{x_i\}_{i=1}^N$ in $\mathbb{R}^D$. Given pixels x_i and x_j, the dis-

The authors gratefully acknowledge support from NSF DMS-2318894.

crete p-weighted Fermat distance between x_i and x_j is

$$\ell_p(x_i, x_j) = \min_{\pi=\{x_k\}_{k=1}^T} \left(\sum_{k=1}^{T-1} \|x_k - x_{k+1}\|^p \right)^{1/p}$$

where π is a path in $\{x_i\}_{i=1}^N$ such that $x_1 = x_i$ and $x_T = x_j$, and $\|\cdot\|$ is the Euclidean 2-norm. Intuitively, $\ell_p(x_i, x_j)$ is small when x_i and x_j are close in the geodesic sense, and the shortest path passes through high-density regions. As $p \geq 1$ increases, paths through high-density regions are favored more than paths through points close in geodesic distance.

Let $\mathcal{G}_X^p$ denote the complete graph with edge weights $\|x_i - x_j\|^p$, and let $\mathcal{G}_X^{p,k} \subset \mathcal{G}_X^p$ be the undirected, union-symmetrized kNN subgraph that retains an edge $\{x_i, x_j\}$ if x_i is one of the k nearest neighbors of x_j. It is shown in [33] that if $k \approx \log(N)$, the sample Fermat distances computed over $\mathcal{G}_X^p$ and $\mathcal{G}_X^{p,k}$ agree with high probability under standard sampling assumptions. Thus, Fermat distances can be computed using Dijkstra's algorithm in $\mathcal{O}(N^2 \log^2(N))$ time instead of $\mathcal{O}(N^3)$.

2.2. Poisson Reweighted Laplacian Learning

We summarize the PWLL-τ method introduced in [27]. Let $X = \mathcal{U} \cup \mathcal{L}$, where $\mathcal{L}$ is the set of labeled data points and $\mathcal{U}$ are unlabeled. Let C be the number of classes, and $e_i \in \mathbb{R}^C$ be the i^{th} canonical basis vector. The PWLL-τ algorithm solves the following problem for a function $u : X \to \mathbb{R}^C$:

$$\min_u \sum_{x_i, x_j \in X} \frac{1}{2} \gamma(x_i)\gamma(x_j) W_{ij} \|u(x_i) - u(x_j)\|^2 + \tau \sum_{x_i \in \mathcal{U}} \|u(x_i)\|^2$$

subject to $u(x) = e_{y(x)}$ for $x \in \mathcal{L}$, where $y(x)$ denotes the class label of x, W_{ij} is the $(i,j)^{th}$ entry of the chosen symmetric, nonnegative weight matrix W, and $\tau \geq 0$ controls the weighting of the diagonal perturbation. Larger τ encourages u to take smaller values for data points away from the labeled set $\mathcal{L}$. In practice, we apply an exponential decay schedule for τ over active learning rounds. The reweighting function $\gamma : X \to \mathbb{R}_+$ is found by solving the graph Poisson equation, $\sum_{x_j \in X} W_{ij}(\gamma(x_i) - \gamma(x_j)) = \sum_{x_k \in \mathcal{L}} (\delta_{ik} - \frac{1}{N})$ for all $x_i \in X$, where δ_{ik} is the Kronecker delta. We resolve the additive ambiguity of solutions by grounding one node and uniformly shifting the resulting potential so its minimum value is strictly positive. Thus, the reweighting γ is learned from the data and is motivated by the continuum version of the graph Poisson equation, whose fundamental solution produces the appropriate scaling of γ in regions near the labeled set $\mathcal{L}$ [27]. The inferred classification $\hat{y}(x_i)$ for a pixel x_i is given by $\hat{y}(x_i) = \arg\max_{k \in \{1,\ldots,C\}} u_k(x_i)$, where $u_k(x_i)$ is the k^{th} entry of $u(x_i) \in \mathbb{R}^C$.

After each PWLL-τ solve, the minimum-norm acquisition function $A : X \to \mathbb{R}_{\geq 0}$, given by $A(x_i) = \|u(x_i)\|$ for $x_i \in X$, is used to iteratively query the next labeled point. After K querying rounds, the $(K+1)^{th}$ queried point is given by $\arg\min_{x_j \in \mathcal{U}} A(x_j)$. Intuitively, A reflects the uncertainty of the classification produced by the PWLL-τ algorithm. With decaying τ-regularization, points selected by A are encouraged to spread over unexplored regions until sufficient labeling has occurred to cover the dataset in the low-label rate regime. Afterward, as $\tau \to 0$, A will take on smaller values in regions of high uncertainty. Thus, the acquisition function A effectively balances exploration and exploitation of data points for querying.

2.3. Landmark Multi-dimensional Scaling

Given data points $\{x_i\}_{i=1}^N$, let $\{z_i\}_{i=1}^m$ denote a set of landmark points, where $m < N$. We use *farthest point sampling* (FPS) [43] to select landmark points, which iteratively chooses the $(k+1)^{th}$ pixel furthest in spectral ℓ_2 norm from the previous k points. Let $D^{\text{dist}} \in \mathbb{R}^{N \times m}$ be the distance matrix defined by $D_{ij}^{\text{dist}} = \ell_p(x_i, z_j)$, let Δ_m be the squared landmark-to-landmark sub-matrix defined by $(\Delta_m)_{ij} = \ell_p(z_i, z_j)^2$. LMDS is run using distances from D^{dist} as follows. We form the mean-centered inner product matrix $B = -H\Delta_m H/2$, where H is the centering matrix given by $H_{ij} = \delta_{ij} - 1/m$. Let $\{\lambda_i\}_{i=1}^{n_p}$ denote the n_p strictly positive eigenvalues of B ordered such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n_p}$, and let v_i denote corresponding orthonormal eigenvectors associated with λ_i for each $i = 1, \ldots, n_p$. For a chosen embedding dimension $r \leq n_p$, the landmark embedding matrix L is given by $L = [\sqrt{\lambda_1} v_1, \sqrt{\lambda_2} v_2, \ldots, \sqrt{\lambda_r} v_r] \in \mathbb{R}^{m \times r}$. The i^{th} row of L gives the r-dimensional LMDS embedding coordinates of z_i, denoted $\{z_i^{MDS}\}_{i=1}^m$. To obtain the remaining embedding coordinates for non-landmark points, let $\Delta_{x_i} \in \mathbb{R}^m$ be defined entrywise by $(\Delta_{x_i})_j = \ell_p(x_i, z_j)^2$ for each pixel x_i. Define $\bar{\Delta}_m \in \mathbb{R}^m$ componentwise by $(\bar{\Delta}_m)_j = \left(\sum_{b=1}^m \ell_p(z_j, z_b)^2\right)/m$. The embedding coordinates of x_i are given by $x_i^{MDS} = L^\dagger(\bar{\Delta}_m - \Delta_{x_i})/2 \in \mathbb{R}^r$, where $L^\dagger$ is the pseudoinverse of L, given by $L^\dagger = [v_1/\sqrt{\lambda_1}, v_2/\sqrt{\lambda_2}, \ldots, v_r/\sqrt{\lambda_r}]^\top \in \mathbb{R}^{r \times m}$.

2.4. Kron Reduction

We now develop the application of Kron reduction to the PWLL harmonic label propagation problem, following the framework of [44]. We first note that the minimization in Section 2.2 is equivalent to minimizing the τ-regularized Dirichlet energy of the Poisson Reweighted Laplacian $L_\gamma = D_\gamma - W_\gamma$, where W_γ is the reweighted weight matrix given entrywise by $(W_\gamma)_{ij} = \gamma_i \gamma_j W_{ij}$, and D_γ is the reweighted diagonal degree matrix given by $(D_\gamma)_{ii} = \sum_j (W_\gamma)_{ij}$. Let $U \in \mathbb{R}^{N \times C}$ be the matrix obtained via stacking $u(x_i)^\top$ row-wise. We can block partition L_γ and U into labeled and unlabeled components: $L_\gamma = \begin{bmatrix} L_{\mathcal{L}\mathcal{L}} & L_{\mathcal{L}\mathcal{U}} \\ L_{\mathcal{U}\mathcal{L}} & L_{\mathcal{U}\mathcal{U}} \end{bmatrix}$ and $U = \begin{bmatrix} U_{\mathcal{L}} \\ U_{\mathcal{U}} \end{bmatrix}$. Note that since $\frac{1}{2} \sum_{x_i, x_j \in X} (W_\gamma)_{ij} \|u(x_i) - u(x_j)\|^2 = \text{tr}(U^T L_\gamma U)$, it follows that the τ-regularized harmonic problem is equiv-

alent to $\min_{U_\mathcal{U}} E(U_\mathcal{L}, U_\mathcal{U}) = \min_{U_\mathcal{U}} \frac{1}{2}\operatorname{tr}(U^\top L_\gamma U) + \frac{\tau}{2}\operatorname{tr}(U_\mathcal{U}^\top U_\mathcal{U})$ subject to the constraint $U_\mathcal{L} = Y_\mathcal{L}$, where $Y_\mathcal{L} \in \mathbb{R}^{|\mathcal{L}|\times C}$ is the label matrix whose i^{th} row is given by $e_{y(x_i)}^\top$ for each labeled point $x_i \in \mathcal{L}$. Given that $L_{\mathcal{U}\mathcal{U}} + \tau I$ is invertible, where I is the $|\mathcal{U}| \times |\mathcal{U}|$ identity, taking the derivative with respect to $U_\mathcal{U}$ yields the optimal $U_\mathcal{U}$, given by $U_\mathcal{U}^\star = -(L_{\mathcal{U}\mathcal{U}} + \tau I)^{-1} L_{\mathcal{U}\mathcal{L}} Y_\mathcal{L}$. We note that $L_{\mathcal{U}\mathcal{U}} + \tau I$ is invertible when $\tau > 0$, and when $\tau = 0$ and every connected component of the graph associated with W_γ contains a labeled node. Substituting $U_\mathcal{U}^\star$ and simplifying yields the τ-regularized reduced energy as a function of $Y_\mathcal{L}$:

$$E(Y_\mathcal{L}, U_\mathcal{U}^\star) = \frac{1}{2}\operatorname{tr}\left(Y_\mathcal{L}^\top \left[L_{\mathcal{L}\mathcal{L}} - L_{\mathcal{L}\mathcal{U}}(L_{\mathcal{U}\mathcal{U}} + \tau I)^{-1} L_{\mathcal{U}\mathcal{L}}\right] Y_\mathcal{L}\right)$$

where $S_\gamma = L_{\mathcal{L}\mathcal{L}} - L_{\mathcal{L}\mathcal{U}}(L_{\mathcal{U}\mathcal{U}} + \tau I)^{-1} L_{\mathcal{U}\mathcal{L}}$ is the Schur complement of the block matrix $\begin{bmatrix} L_{\mathcal{L}\mathcal{L}} & L_{\mathcal{L}\mathcal{U}} \\ L_{\mathcal{U}\mathcal{L}} & L_{\mathcal{U}\mathcal{U}} + \tau I \end{bmatrix}$ with respect to $L_{\mathcal{U}\mathcal{U}} + \tau I$. In particular, the quadratic form of S_γ represents the reduced τ-regularized Dirichlet energy over labeled nodes $\mathcal{L}$ after unlabeled nodes $\mathcal{U}$ have been eliminated through optimality conditions.

3. LEARNING THE FERMAT EXPONENT

The Fermat geometry is heavily dependent on the choice of p, which determines how density-aware the underlying data manifold is. We devise two methods, exact leave-one-out (ELOO) and approximate leave-one-out (ALOO), for learning p during the active learning process, motivated by [41, 42]. ELOO exactly evaluates the ELOO cross-validation criterion for each candidate exponent in a fixed set, but it is expensive and impractical for larger HSI scenes and budgets. Thus, we propose ALOO, which uses principles of Kron reduction to learn p efficiently.

3.1. Exact LOO p-learning

Suppose we have a set of candidate exponents $\mathcal{P}$ and a current labeled set $\mathcal{L}$. For each $a \in \mathcal{L}$ we consider leaving out a and forming the reduced set $\mathcal{R}_a = \mathcal{L} \setminus \{a\}$. For each $p \in \mathcal{P}$, we form the Poisson reweighted weight matrix $W^{(p)}_{\gamma,\mathcal{R}_a}$ using labels in $\mathcal{R}_a$, defined by $(W^{(p)}_{\gamma,\mathcal{R}_a})_{ij} = \gamma_i^{(p,\mathcal{R}_a)}\gamma_j^{(p,\mathcal{R}_a)} W^{(p)}_{ij}$. The weight matrix $W^{(p)}$ is formed using LMDS with landmark distances ℓ_p, and $\gamma^{(p,\mathcal{R}_a)}$ is the solution of the graph Poisson equation with weights $W^{(p)}$ using labeled nodes $\mathcal{R}_a$. We then solve the PWLL-τ minimization subject to $u(x) = e_{y(x)}$ for all $x \in \mathcal{R}_a$ to obtain $u^{(p)}_{\mathcal{R}_a} : X \to \mathbb{R}^C$. The ELOO prediction for the held-out label a is thus given by $u^{(p)}_{\mathcal{R}_a}(a) \in \mathbb{R}^C$. The ELOO algorithm solves the PWLL-τ minimization problem over $\mathcal{R}_a$ for each $a \in \mathcal{L}$ and for each $p \in \mathcal{P}$, leading to $|\mathcal{P}||\mathcal{L}|$ total solves.

We then define the ELOO score function $S_{\text{ELOO}}(p) = \frac{1}{|\mathcal{L}|}\sum_{a\in\mathcal{L}} \|\Pi(u^{(p)}_{\mathcal{R}_a}(a)) - e_{y(a)}\|^2$, where $\Pi : \mathbb{R}^C \to \mathbb{R}^C$ is a normalization function defined by $\Pi(x) = \frac{[x]_+}{\sum_{i=1}^C [x_i]_+}$ when $\sum_{i=1}^C [x_i]_+ > \epsilon_{\text{mach}}$ and $\Pi(x) = \frac{1}{C}\mathbf{1}$ otherwise, where $\mathbf{1}$ is the all-ones vector, ϵ_{mach} is machine precision, and $[x]_+ = (\max\{x_1, 0\}, \ldots, \max\{x_C, 0\})^\top$. Thus, S_{ELOO} calculates the average difference between normalized ELOO predictions and true labels. We introduce a confidence margin term, $M_{\text{ELOO}}(p) = \frac{1}{|\mathcal{L}|}\sum_{a\in|\mathcal{L}|} \lambda_a(p)$, where $\lambda_a(p) = \Pi(u^{(p)}_{\mathcal{R}_a}(a))_{y(a)} - \max_{k\neq y(a)} \Pi(u^{(p)}_{\mathcal{R}_a}(a))_k$. Hence, $M_{\text{ELOO}}(p)$ finds the average margin between the true class score and the largest competing class score. The final margin-regularized ELOO score is given by $\mathbf{S}_{\text{ELOO}}(p) = \nu_1 M_{\text{ELOO}}(p) - S_{\text{ELOO}}(p)$, where ν_1 is a weight parameter.

3.2. Approximate LOO p-learning

In ALOO, we first form the Poisson reweighted weight matrix $W^{(p)}_{\gamma,\mathcal{L}}$ using all labels in $\mathcal{L}$ for each $p \in \mathcal{P}$, with associated Poisson reweighted Laplacian $L^{(p)}_{\gamma,\mathcal{L}}$. We then solve the PWLL-τ minimization subject to $u(x) = e_{y(x)}$ for all $x \in \mathcal{L}$ to obtain inferred classifications $u^{(p)}_\star : X \to \mathbb{R}^C$. Then, let $U^\star_\mathcal{U} \in \mathbb{R}^{|\mathcal{U}|\times C}$ denote the matrix with i^{th} row given by $u^{(p)}_\star(x_i)$, where x_i is the i^{th} unlabeled data point in $\mathcal{U}$. Instead of exactly solving the PWLL-τ problem, we form the Schur complement $S_p = (L^{(p)}_{\gamma,\mathcal{L}})_{\mathcal{L}\mathcal{L}} - (L^{(p)}_{\gamma,\mathcal{L}})_{\mathcal{L}\mathcal{U}}[(L^{(p)}_{\gamma,\mathcal{L}})_{\mathcal{U}\mathcal{U}} + \tau I]^{-1}(L^{(p)}_{\gamma,\mathcal{L}})_{\mathcal{U}\mathcal{L}}$ (Section 2.4). Then, for each $a \in \mathcal{L}$, we can define block-partitioned matrices from sets $\{a\}$ and $\mathcal{R}_a$ as $F_\mathcal{L} = \begin{bmatrix} F_a \\ Y_{\mathcal{R}_a} \end{bmatrix}$ and $S_p = \begin{bmatrix} S_{p,aa} & S_{p,a\mathcal{R}_a} \\ S_{p,\mathcal{R}_a a} & S_{p,\mathcal{R}_a\mathcal{R}_a} \end{bmatrix}$. In particular, $F_a \in \mathbb{R}^{1\times C}$ is the unknown class prediction for a, and $Y_{\mathcal{R}_a} \in \mathbb{R}^{|\mathcal{R}_a|\times C}$ denote the matrix of fixed labels. When a is held out, the τ-regularized energy as a function of $Y_\mathcal{L}$ is given by $E_a(Y_\mathcal{L}, U^\star_\mathcal{U}) = \frac{1}{2}\operatorname{tr}\left(Y_\mathcal{L}^\top S_p Y_\mathcal{L}\right) + \frac{\tau}{2}\|F_a\|^2$. It follows that minimizing $E_a(Y_\mathcal{L}, U^\star_\mathcal{U})$ is equivalent to minimizing $E_a(F_a, Y_{\mathcal{R}_a}) = \frac{1}{2}(S_{p,aa} + \tau)\|F_a\|_2^2 + \operatorname{tr}(F_a^\top S_{p,a\mathcal{R}_a} Y_{\mathcal{R}_a}) + K_{p,a}$, where $K_{p,a} = \frac{1}{2}\operatorname{tr}(Y_{\mathcal{R}_a}^\top S_{p,\mathcal{R}_a\mathcal{R}_a} Y_{\mathcal{R}_a})$ is a constant with respect to F_a. Hence, taking the first-order condition yields $(S_{p,aa} + \tau)F_a + S_{p,a\mathcal{R}_a} Y_{\mathcal{R}_a} = 0$. Solving for F_a and finding ALOO predictions for a thus amounts to computing $F^{(\mathcal{R}_a)}_{p,a} = -\frac{S_{p,a\mathcal{R}_a} Y_{\mathcal{R}_a}}{S_{p,aa}+\tau}$.

ALOO thus computes predictions $F^{(\mathcal{R}_a)}_{p,a}$ for each $a \in \mathcal{L}$ and each $p \in \mathcal{P}$, without solving PWLL-τ for each set $\mathcal{R}_a$. Thus, the ALOO score function is $S_{\text{ALOO}}(p) = \frac{1}{|\mathcal{L}|}\sum_{a\in\mathcal{L}} \|\Pi(F^{(\mathcal{R}_a)}_{p,a}) - e_{y(a)}\|^2$ with the corresponding confidence margin term $M_{\text{ALOO}}(p) = \frac{1}{|\mathcal{L}|}\sum_{a\in\mathcal{L}} \lambda'_a(p)$, where $\lambda'_a(p) = \Pi(F^{(\mathcal{R}_a)}_{p,a})_{y(a)} - \max_{k\neq y(a)} \Pi(F^{(\mathcal{R}_a)}_{p,a})_k$. The final margin-regularized ALOO score value is hence given by $\mathbf{S}_{\text{ALOO}}(p) = \nu_2 M_{\text{ALOO}}(p) - S_{\text{ALOO}}(p)$. We use $\nu_1 = \nu_2 = 0.02$ in experiments. The ALOO formulation assumes that $W^{(p)}_{\gamma,\mathcal{L}} \approx W^{(p)}_{\gamma,\mathcal{R}_a}$ for each $a \in \mathcal{L}$ and each $p \in \mathcal{P}$. Numerical experiments suggest that this property holds.

4. THE PROPOSED ALGORITHMS

We propose two algorithms, FALL and A-FALL, which utilize Fermat distances in the underlying affinity matrix, leading to a density-aware estimation of the underlying data manifold. We further propose a method for learning the optimal exponent parameter p in A-FALL using ALOO that is scalable to large HSI scenes (Section 3). FALL and A-FALL show improved accuracy, with A-FALL being an efficient alternative.

A-FALL is outlined in Algorithm 2. The FPS algorithm [43] is first used to find m landmark points, ensuring that a spectrally diverse set of points is chosen efficiently. We fix $m = 300$ in experiments. Then, we compute Fermat distances over landmark points via shortest paths using Dijkstra's algorithm over a sparse kNN graph over X. The corresponding LMDS embeddings $\{x_i^{MDS}\}_{i=1}^N$ are computed according to section 2.3, where we set $r = 32$. LMDS approximates Fermat distances efficiently, making Fermat graph construction and p-learning methods computationally feasible.

In A-FALL, $W^{(p)}$ uses LMDS distances $d_p(i,j) = \|x_i^{\text{MDS}} - x_j^{\text{MDS}}\|$ for the corresponding exponent p and a self-tuned Gaussian kernel inspired by [45]. For each $x \in X$, we define the local bandwidth $\sigma_x = d_p(x, x_{k_\sigma})$ where x_{k_σ} is the k_σ^{th} nearest neighbor of x in the LMDS space. Given that $\mathcal{N}_{k_G}^{(p)}(i)$ is the set of k_G nearest neighbors of x_i excluding x_i itself, $W^{(p)}$ is given entrywise by $W_{ij}^{(p)} = \max(\widehat{W}_{ij}, \widehat{W}_{ji})$ and $W_{ii}^{(p)} = 0$, where $\widehat{W}_{ij} = \exp\left(-\frac{d_p(x_i,x_j)^2}{\eta^2 \sigma_i \sigma_j}\right)$ if $j \in \mathcal{N}_{k_G}^{(p)}(i)$ and $j \neq i$, and $\widehat{W}_{ij} = 0$ otherwise. We set $k_\sigma = k_G = 20$ and $\eta = 8$ in experiments. The self-tuned kernel improves numerical conditioning of the PWLL-τ solve.

FALL is outlined in algorithm 1. The weight matrix $W^{(p)}$ is defined entrywise in the same way as A-FALL, but with Fermat distances $d_p(i,j) = \ell_p(x_i, x_j)$. This leads to an exact representation of the Fermat geometry at the cost of higher computation time. In both algorithms, PWLL-τ is used as the downstream classifier as it mitigates the low-label regime degeneracy of Laplace learning through Poisson reweighting [27]. Additionally, τ-regularization increases numerical stability of the PWLL solve and ensures the minimum-norm acquisition function A balances exploration and exploitation of data points. To ensure invertibility of $L_{\mathcal{U}\mathcal{U}} + \tau I$, we require that the underlying graph associated with $W^{(p)}$ is connected. We use a decay schedule for τ as in [27], given by $\tau_t = \tau_0 \left(\varepsilon/\tau_0\right)^{t/(2C)}$ when $t < 2C$ and $\tau_t = 0$ when $t \geq 2C$, and $t = b - 1$ where b is the current active learning round.

5. EXPERIMENTAL RESULTS

We show the improved experimental accuracy of FALL and A-FALL on the Salinas A HSI and a subset of the Pavia University HSI scene [4]. FALL, A-FALL and PWLL-τ were implemented in Python using NumPy and SciPy. The full Salinas scene was captured in Salinas Valley, California, and

Algorithm 1: FALL

Input: HSI $X = \{x_i\}_{i=1}^N$; labeled set $\mathcal{L}$; labels $\{y_i\}_{i\in\mathcal{L}}$; classes $\{1,\dots,C\}$; Regularization $\{\tau_t\}_{t=1}^B$; exponent p; Budget B; Oracle $\mathcal{O}$; Initial Labels $\mathcal{L}_0$;
Output: Predicted labels $\widehat{Y}$

Compute $\ell_p(x_i, x_j)$ for $i, j = 1, \dots, N$ and form weight matrix $W^{(p)}$
Initialize $\mathcal{L}_{\text{curr}} \leftarrow \mathcal{L}_0$
for $b = 1 : B$ **do**
  Solve the graph Poisson system with source induced by $\mathcal{L}_{\text{curr}}$ to obtain γ
  Reweight the graph $(W_\gamma^{(p)})_{ij} = \gamma_i \gamma_j W_{ij}^{(p)}$
  Run PWLL-τ using weights $W_\gamma^{(p)}$, labels $\mathcal{L}_{\text{curr}}$ and τ_b weighting to obtain scores $u : X \to \mathbb{R}^C$
  Select $i^\star = \arg\min_{i\notin\mathcal{L}} \|u(x_i)\|$
  Query $y_{i^\star} = \mathcal{O}(x_{i^\star})$ from $\mathcal{L}$
  Update $\mathcal{L}_{\text{curr}} \leftarrow \mathcal{L}_{\text{curr}} \cup \{x_{i^\star}\}$
Run PWLL-τ with final labels $\mathcal{L}_{\text{curr}}$ and τ_B weighting for $u : X \to \mathbb{R}^C$ and assign $\widehat{y}_i = \arg\max_k u_k(x_i)$
return $\widehat{Y} = \{\hat{y}_i\}_{i=1}^N$

Algorithm 2: A-FALL

Input: HSI $X = \{x_i\}_{i=1}^N$; labeled set $\mathcal{L}$; candidate set $\mathcal{P}$; landmarks m; MDS dimension r; budget B; update period T; oracle $\mathcal{O}$; initial exponent p_0; regularization $\{\tau_b\}_{b=1}^B$; Initial Labels $\mathcal{L}_0$;
Output: Predicted labels $\widehat{Y}$.
Select landmarks $Z = \{z_a\}_{a=1}^m \subset X$ using FPS.
for $p \in \mathcal{P}$ **do**
  Form D from distances $\ell_p(x_i, z_j)$ for each $i = 1, \dots, N$ and $j = 1, \dots, m$.
  Form self-tuned kNN weight matrix $W^{(p)}$ by computing embedding coordinates $\{x_i^{MDS}\}_{i=1}^N$.
Initialize $p_{\text{curr}} \leftarrow p_0$, $\mathcal{L}_{\text{curr}} \leftarrow \mathcal{L}_0$
for $b = 1 : B$ **do**
  if $b \geq T$, $b \bmod T = 0$, *and* $b < B$ **then**
    for $p \in \mathcal{P}$ **do**
      Form S_p and $F_{p,a}^{(\mathcal{R}_a)}$ for each $a \in \mathcal{L}_{\text{curr}}$
      Compute $\mathbf{S}_{\text{ALOO}}(p)$
    Set $p_{\text{curr}} \leftarrow \arg\max_{p\in\mathcal{P}} \mathbf{S}_{\text{ALOO}}(p)$
  Run PWLL-τ using weights $W^{(p_{\text{curr}})}$, labels $\mathcal{L}_{\text{curr}}$ and τ_b weighting to obtain scores $u : X \to \mathbb{R}^C$
  Select $i^\star = \arg\min_{i\notin\mathcal{L}} \|u(x_i)\|$
  Query $y_{i^\star} = \mathcal{O}(x_{i^\star})$ from $\mathcal{L}$
  Update $\mathcal{L}_{\text{curr}} \leftarrow \mathcal{L}_{\text{curr}} \cup \{x_{i^\star}\}$
Run PWLL-τ with final labels $\mathcal{L}_{\text{curr}}$ and τ_B weighting for $u : X \to \mathbb{R}^C$ and assign $\widehat{y}_i = \arg\max_k u_k(x_i)$
return $\widehat{Y} = \{\widehat{y}_i\}_{i=1}^N$

has a 3.7-meter spatial resolution per pixel with 224 spectral bands. The ground truth has 16 classes and a scene size of 512×217 pixels. The Salinas A HSI is a subset of the Salinas dataset, with a scene size of 83×86 pixels and 6 classes. The Pavia University HSI was captured over the University of Pavia, Italy, by the ROSIS sensor. The Pavia University HSI has a scene size of 610×340 pixels with 103 spectral bands, and a spatial resolution of 1.3 meters per pixel. The ground truth has 9 classes. In our experiments, we use a Pavia University subset of size 75×180 with 9 classes, corresponding to rows 253 to 327 and columns 30 to 209 of the original HSI. The performance of our algorithms is assessed using Overall Accuracy (OA, total fraction of correctly labeled pixels) and Average Accuracy (AA, mean per-class accuracies).

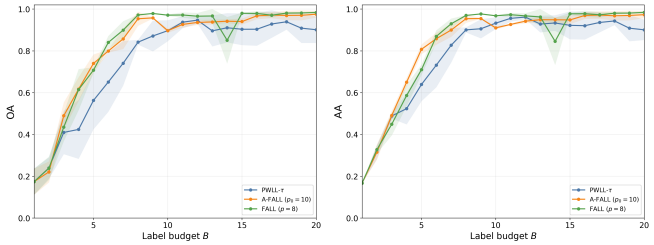
We test FALL and A-FALL against PWLL-τ. We set $\tau_0 = 10^{-3}$, $\varepsilon = 10^{-9}$, and the p-learning update period $T = 10$ in experiments. For A-FALL, we set $\mathcal{P} = \{1.5, 2, 3, 4, 6, 8, 10, 12\}$ and $p_0 = 10$ for Salinas A. Similarly, we set $\mathcal{P} = \{1.25, 1.5, 2, 3, 4, 6, 8, 10, 12, 14\}$ and $p_0 = 4$ for the Pavia subset. Optimal values for p_0 were found via grid search. We set $\mathcal{L}_0 = \{x_R\}$ where x_R is a randomly selected non-background point in X. Experiments were run on an ASUS ROG Strix G15 with 16GB of memory and an AMD Ryzen 7 4800H processor.

Table 1 and Figure 1 show the efficacy of FALL and A-FALL over PWLL-τ, with visualizations in Figure 2. FALL achieves the best OA and AA on Salinas A, while A-FALL obtains similar OA and AA scores and takes only 23.45 seconds to run on average. On Pavia, A-FALL obtains the highest OA and AA with the smallest average runtime of 93.48 seconds, demonstrating its scalability to larger HSI scenes.

Table 1. Comparison of PWLL-τ, FALL, and A-FALL on Salinas A at $B = 20$ and the PaviaU crop at $B = 30$, averaged over 10 seeds. Time is averaged over 10 seeds, measured in seconds. $\pm$ represents 1 standard deviation across seeds.

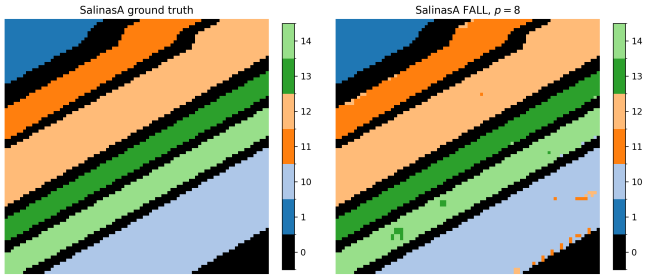
Method	p/p_0	OA	AA	Time
Salinas A, $B = 20$				
PWLL-τ	–	0.9010±0.0600	0.9006±0.0462	**11.67**
FALL	$p = 8$	**0.9837±0.0053**	**0.9841±0.0047**	37.40
A-FALL	$p_0 = 10$	0.9753±0.0159	0.9731±0.0163	23.45
PaviaU Subset, $B = 30$				
PWLL-τ	–	0.8416±0.0366	0.6252±0.0443	130.54
FALL	$p = 8$	0.8918±0.0358	0.5987±0.0685	197.21
A-FALL	$p_0 = 4$	**0.9055±0.0300**	**0.6307±0.0586**	**93.48**

We provide comparisons of ELOO and ALOO p-learning on the Pavia subset and display the differences in OA and AA in Figure 3. At $B = 30$, ELOO achieved an OA of 0.9066 ± 0.0297 and AA of 0.6296 ± 0.0589 with a mean runtime 252.59 s, while ALOO achieved an OA 0.9055 ± 0.03 and AA of 0.6307 ± 0.0586 with a mean runtime of 93.48 s.



(a) OA (b) AA

Fig. 1. Mean OA and AA from $B = 1$ to $B = 20$ on Salinas A. Shaded regions show 1 standard deviation across seeds.

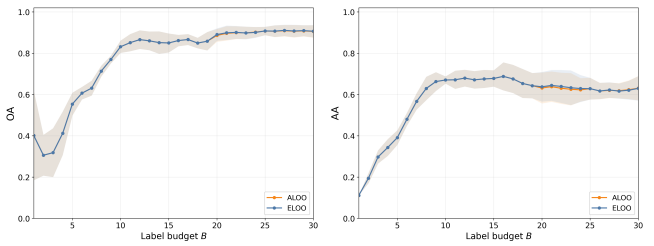


(a) Ground truth (b) FALL prediction

Fig. 2. Salinas A GT vs FALL for $p = 8$, B = 20, best seed.

6. CONCLUSIONS AND FUTURE WORK

The proposed algorithms, FALL and A-FALL, provide improvements in overall accuracy compared to PWLL-τ in the low-label regime. Both algorithms provide a density-aware representation of the data manifold, with A-FALL using approximate Fermat distances via LMDS for greater efficiency. We introduce two methods to learn the Fermat exponent p in our algorithm through cross-validation. In future work, we hope to explore further MDS variants that better preserve local Fermat distances and consider continuous or gradient-based p-learning methods. Further theoretical analysis on how using Fermat distances influences active class discovery is another direction of interest.



(a) OA (b) AA

Fig. 3. A-FALL with ELOO vs ALOO p-learning on the PaviaU subset from $B = 1$ to $B = 30$ over 10 seeds.

7. REFERENCES

[1] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised learning in remote sensing: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 4, pp. 213–247, 2022.

[2] Z. Li, B. Chen, S. Wu, M. Su, J. M. Chen, and B. Xu, "Deep learning for urban land use category classification: A review and experimental assessment," *Remote Sens. Environ.*, vol. 311, Sept. 2024, Art. no. 114290.

[3] M. Ahmad, S. Distefano, A. M. Khan, M. Mazzara, C. Li, H. Li, J. Aryal, Y. Ding, G. Vivone, and D. Hong, "A comprehensive survey for hyperspectral image classification: The evolution from conventional to transformers and Mamba models," *Neurocomputing*, vol. 644, Sept. 2025, Art. no. 130428.

[4] Grupo de Inteligencia Computacional (GIC), "Hyperspectral Remote Sensing Scenes," [Online]. Available: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, 2021, Accessed: Jul. 27, 2026.

[5] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, 2021.

[6] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*. 2003, pp. 912–919, AAAI Press.

[7] J. Calder, B. Cook, M. Thorpe, and D. Slepčev, "Poisson learning: Graph based semi-supervised learning at very low label rates," in *Proceedings of the 37th International Conference on Machine Learning*. 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 1306–1316, PMLR.

[8] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," in *Advances in Neural Information Processing Systems*. 1995, vol. 7, pp. 705–712, MIT Press.

[9] M.-F. Balcan, A. Broder, and T. Zhang, "Margin based active learning," in *Learning Theory*. 2007, vol. 4539 of *Lecture Notes in Computer Science*, pp. 35–50, Springer.

[10] R. M. Castro and R. D. Nowak, "Minimax bounds for active learning," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2339–2353, 2008.

[11] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," *J. Comput. Syst. Sci.*, vol. 75, no. 1, pp. 78–89, 2009.

[12] S. Dasgupta, "Two faces of active learning," *Theor. Comput. Sci.*, vol. 412, no. 19, pp. 1767–1781, 2011.

[13] S. Rajan, J. Ghosh, and M. M. Crawford, "An active learning approach to hyperspectral data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 4, pp. 1231–1242, 2008.

[14] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, 2010.

[15] B. Demir, C. Persello, and L. Bruzzone, "Batch-mode active-learning methods for the interactive classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 1014–1031, 2011.

[16] W. Di and M. M. Crawford, "Active learning via multiview and local proximity co-regularization for hyperspectral image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 618–628, 2011.

[17] D. Tuia, M. Volpi, L. Copa, M. Kanevski, and J. Muñoz-Marí, "A survey of active learning algorithms for supervised remote sensing image classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 606–617, 2011.

[18] S. Patra, K. Bhardwaj, and L. Bruzzone, "A spectral-spatial multicriteria active learning technique for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 12, pp. 5213–5227, 2017.

[19] J. M. Haut, M. E. Paoletti, J. Plaza, J. Li, and A. Plaza, "Active learning with convolutional neural networks for hyperspectral image classification using a new Bayesian approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6440–6461, 2018.

[20] J. M. Murphy and M. Maggioni, "Iterative active learning with diffusion geometry for hyperspectral images," in *2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2018, pp. 1–5.

[21] J. M. Murphy and M. Maggioni, "Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1829–1845, 2019.

[22] M. Maggioni and J. M. Murphy, "Learning by active nonlinear diffusion," *Found. Data Sci.*, vol. 1, no. 3, pp. 271–291, 2019.

[23] J. M. Murphy, "Spatially regularized active diffusion learning for high-dimensional images," *Pattern Recognit. Lett.*, vol. 135, pp. 213–220, 2020.

[24] S. L. Polk, K. Cui, R. J. Plemmons, and J. M. Murphy, "Active diffusion and VCA-assisted image segmentation of hyperspectral images," in *IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 1364–1367.

[25] K. Tripathi and J. M. Murphy, "Learning by evolving nonlinear diffusion for active learning on hyperspectral images," in *2024 14th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. IEEE, 2024, pp. 1–5.

[26] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009, [Online]. Available: https://research.cs.wisc.edu/techreports/2009/TR1648.pdf.

[27] K. Miller and J. Calder, "Poisson reweighted Laplacian uncertainty sampling for graph-based active learning," *SIAM J. Math. Data Sci.*, vol. 5, no. 4, pp. 1160–1190, 2023.

[28] C. D. Howard and C. M. Newman, "Geodesics and spanning trees for Euclidean first-passage percolation," *Ann. Probab.*, vol. 29, no. 2, pp. 577–623, 2001.

[29] M. Alamgir and U. von Luxburg, "Shortest path distance in random k-nearest neighbor graphs," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1031–1038.

[30] S. J. Hwang, S. B. Damelin, and A. O. Hero, III, "Shortest path through random points," *Ann. Appl. Probab.*, vol. 26, no. 5, pp. 2791–2823, 2016.

[31] A. Little, M. Maggioni, and J. M. Murphy, "Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms," *J. Mach. Learn. Res.*, vol. 21, no. 6, pp. 1–66, 2020.

[32] P. Groisman, M. Jonckheere, and F. Sapienza, "Nonhomogeneous Euclidean first-passage percolation and distance learning," *Bernoulli*, vol. 28, no. 1, pp. 255–276, 2022.

[33] A. Little, D. McKenzie, and J. M. Murphy, "Balancing geometry and density: Path distances on high-dimensional data," *SIAM J. Math. Data Sci.*, vol. 4, no. 1, pp. 72–99, 2022.

[34] N. García Trillos, A. Little, D. McKenzie, and J. M. Murphy, "Fermat distances: Metric approximation, spectral convergence, and clustering algorithms," *J. Mach. Learn. Res.*, vol. 25, no. 176, pp. 1–65, 2024.

[35] Sajama and A. Orlitsky, "Estimating and computing density based distance metrics," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 760–767.

[36] A. S. Bijral, N. Ratliff, and N. Srebro, "Semi-supervised learning with density based distances," in *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 2011, pp. 43–50.

[37] D. McKenzie and S. Damelin, "Power weighted shortest paths for clustering Euclidean data," *Found. Data Sci.*, vol. 1, no. 3, pp. 307–327, 2019.

[38] X. Fernández, E. Borghini, G. Mindlin, and P. Groisman, "Intrinsic persistent homology via density-based metric learning," *J. Mach. Learn. Res.*, vol. 24, no. 75, pp. 1–42, 2023.

[39] R. Tan and Y. Zang, "High-dimensional semi-supervised classification via the Fermat distance," 2026, arXiv:2604.23573.

[40] V. de Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems*. 2002, vol. 15, pp. 705–712, MIT Press.

[41] X. Zhang and W. S. Lee, "Hyperparameter learning for graph based semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*. 2006, vol. 19, pp. 1585–1592, MIT Press.

[42] S. Wang, W. Zhou, H. Lu, A. Maleki, and V. Mirrokni, "Approximate leave-one-out for fast parameter tuning in high dimensions," in *Proceedings of the 35th International Conference on Machine Learning*. 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5228–5237, PMLR.

[43] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Trans. Image Process.*, vol. 6, no. 9, pp. 1305–1315, 1997.

[44] F. Dörfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[45] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*. 2004, vol. 17, pp. 1601–1608, MIT Press.