GSSL PROJECT REPORT (SHOULD CHANGE THIS)

ROHIN GILMAN, ALEX JOHNSON, AND KAITLYNN LILLY

Abstract. To be written last

1. Introduction

Graphical semi-supervised learning (GSSL) is a prominent field of research in machine learning as it serves as a powerful technique for dealing with classification problems in which only a small subset of the data is labeled. Incorporating kernel methods into GSSL has emerged as a popular approach in machine learning due to its ability to leverage both labeled and unlabeled data, resulting in improved accuracy and robustness. In kernel methods, the unlabeled data is used to construct a similarity graph, where each node represents an instance and the edges indicate the similarity between them. The graph is then used to propagate the labels from the labeled data to the unlabeled data, resulting in a complete labeling of the data set. As a result, graphical semi-supervised learning using kernel methods is a promising approach for tackling real-world problems in various domains such as image recognition, natural language processing, and bioinformatics.

2. Theoretical Background

2.1. Types of Graphs.

2.1.1. *Proximity Graphs.* Given a set of n points $\{x_1, x_2, \ldots, x_n\}$ in a metric space, a proximity graph is a graph, $G = (V, W)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices representing the points, and W is an adjacency matrix representing the edges connecting pairs of vertices that are "close" to each other, according to some measure of proximity.

Date: June 9, 2023.

To represent the proximity graph G using a weight matrix, we define a kernel function $K\colon X \times X \to \mathbb{R}$ that maps pairs of points in X to a real number, which captures the similarity between the points.

The weight matrix, W, corresponding to the proximity graph G is then defined by $W_{i,j} = K(x_i, x_j)$. The nonzero entries correspond to the similarity between pairs of points that are connected by an edge in G. Diagonal entries, $W_{i,i}$, correspond to the "self-similarity" of each point x_i, and the off-diagonal entries, $W_{i,j}$, correspond to the similarity between pairs of points x_i and x_j. By using a kernel function to define the weights of the edges, proximity graphs can capture underlying geometric relationships between the data points, making them suitable for a wide range of applications in machine learning and data analysis.

2.1.2. *K-Nearest Neighbor.* K-Nearest Neighbor (K-NN) graphs are a type of proximity graph for which edge weights are determined exclusively by the nearest neighbors of a given vertex. Given a set of n vertices $V = \{x_1, x_2, ..., x_n\}$ in a metric space, the K-NN graph $G_k = (V, W)$ is constructed as follows: for each point x_i, its k-nearest neighbors are found according to some distance metric, and edges of weight 1 are added between x_i and each of its k-nearest neighbors. In other words, x_i is adjacent to the k vertices that are closest to it in the data set.

K-NN graphs have several advantages over other types of proximity graphs. One of these advantages is that on average, vertices have relatively low degree, so the weight matrix W is often sparse. This increases the performance of graphical methods. Moreover, K-NN graphs are easy to construct and can be used in a wide range of applications, such as clustering, classification, and anomaly detection.

2.2. **Graph Laplacian.** Given a graph $G = (V, W)$, we can define a graph Laplacian L as a matrix that captures the pairwise relationships between data points on the graph. Specifically, we can construct the unnormalized graph Laplacian L as follows:

$$L = D - W,$$

where D is the diagonal degree matrix. The degree of a vertex is the sum of the weights of the edges adjacent to the vertex. We note that if G has M connected components, then $\text{Null}(L) = M$. Thus, we simply need to determine the degree of the zero eigenvalue of L to determine the number of clusters in the data.

One common variant of the graph Laplacian is the normalized Laplacian, which is defined as:

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2}.$$

The normalized Laplacian satisfies the important property that its eigenvalues are in the range $[0, 2]$, which allows for efficient algorithms and analysis.

2.3. **Types of Kernels.**

2.3.1. *Matérn Family.* The Matérn family is a class of parametric covariance functions that includes a wide range of covariance functions that are characterized by two parameters: a smoothness parameter, ν, and a length scale parameter, ℓ. The smoothness parameter determines how quickly the correlation between two points decays as the distance between them increases, while the length scale parameter controls the range of spatial dependence.

In GSSL, the Matérn family is often used as a graph Laplacian regularization term to enforce smoothness and continuity in the labeling of graph vertices. The graph Laplacian is constructed based on the pairwise similarities or distances between data points, and the Matérn family is used to model the spatial dependence in these pairwise similarities or distances. By including the Matérn family as a regularization term in the graph Laplacian, GSSL algorithms can effectively leverage the spatial structure of the data to improve the quality of the labeling.

2.3.2. *Green's Function of Elliptic Differential Operators.* A Green's function is a solution to the equation:

$$\mathcal{L}G(x, y) = \delta(x - y),$$

where $\mathcal{L}$ is an elliptic differential operator, δ is the Dirac delta function, and $G(x, y)$ is the Green's function. If we have a linear elliptic differential equation of the form

$$\mathcal{L}u(x) = f(x),$$

with boundary conditions, we can solve for the solution $u(x)$ using the Green's function as follows:

$$u(x) = \int G(x, y)f(y)\mathrm{d}y,$$

where the integral is taken over the domain of the equation.

Green's functions of elliptic differential operators provide a natural way to define diffusion processes on graphs in GSSL, where the information from the labeled points diffuses through the graph to the unlabeled points. This diffusion process can be described using a diffusion kernel, which is defined in terms of the graph Laplacian, an elliptic differential operator on the graph. The Green's function of the graph Laplacian can be used to construct the diffusion kernel, which in turn can be used to define a diffusion process that spreads the information from the labeled points to the unlabeled points. This diffusion process can then be used to make predictions for the unlabeled data points, based on the information that has diffused from the labeled points.

2.4. **GSSL Algorithms.**

2.4.1. *The Probit Method.* The probit method is a statistical technique used for modeling binary response variables. We assume

$$y_j = \mathrm{sign}(f(x_j) + \epsilon_j),$$

for some latent function f, where $\epsilon_j \overset{\text{iid}}{\sim} \psi$, for some probability density ψ. Assume ψ is symmetric, so that

$$\begin{aligned}
\mathbb{P}(y_j = +1 \mid f) &= \mathbb{P}(f(x_j) + \epsilon_j \geq 0) \\
&= \mathbb{P}(\epsilon_j \geq -f(x_j)) \\
&= \int_{-f(x_j)}^{\infty} \psi(t)\mathrm{d}t \\
&= \int_{-\infty}^{f(x_j)} \psi(t)\mathrm{d}t \\
&= \Psi(f(x_j)) \\
&= \Psi(y_j f(x_j)),
\end{aligned}$$

where Ψ is the cumulative distribution function (CDF) of ψ. By the same calculation, we obtain $\mathbb{P}(y_j = -1 \mid f) = \Psi(y_j f(x_j))$. Thus, $\mathbb{P}(y_j \mid f) = \Psi(y_j f(x_j))$. Since the ϵ_j are independent and identically distributed, we can write

$$\mathbb{P}(y \mid f) = \prod_{j=1}^{n} \Psi(y_j f(x_j)).$$

For a function f, we want our loss function to be large when the signs of $f(x_j)$ and y_j disagree, so we define our probit loss:

$$L(f) = -\log(\mathbb{P}(y \mid f)) = -\sum_{j=1}^{n} \Psi(y_j f(x_j)).$$

Finally, we add a regularization term based on the RKHS norm to define our optimization problem.

3. Methods

Write about what it is we are actually going to do

4. Results

5. Conclusion