Higher Order Poisson Regularization for Graph-Based Semi-Supervised Learning

Dingjun Bian

Advisor: Jeff Calder

School of Mathematics
University of Minnesota

UMN Honors Thesis

May 4, 2021

Outline

Outline

A Quick Introduction to Learning

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
   - Does not make use of label information

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
   - Does not make use of label information
   - Learns structures in high dimensional data using only the unlabeled data features

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
   - Does not make use of label information
   - Learns structures in high dimensional data using only the unlabeled data features
   - Clustering (Customer Segmentation and Recommendation System)

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
    - Makes use of labeled data to infer a function for future predictions
    - Classification (Self-Driving Cars)
    - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
    - Does not make use of label information
    - Learns structures in high dimensional data using only the unlabeled data features
    - Clustering (Customer Segmentation and Recommendation System)
3. **Semi-Supervised Machine Learning (SSL)**

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
    - Makes use of labeled data to infer a function for future predictions
    - Classification (Self-Driving Cars)
    - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
    - Does not make use of label information
    - Learns structures in high dimensional data using only the unlabeled data features
    - Clustering (Customer Segmentation and Recommendation System)
3. **Semi-Supervised Machine Learning (SSL)**
    - Makes use of both label information and unlabeled data features

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
   - Does not make use of label information
   - Learns structures in high dimensional data using only the unlabeled data features
   - Clustering (Customer Segmentation and Recommendation System)
3. **Semi-Supervised Machine Learning (SSL)**
   - Makes use of both label information and unlabeled data features
   - Classification (More on this later)

A Quick Introduction to Learning

There are three major types of machine learning algorithm:

1. **Fully Supervised Machine Learning**
   - Makes use of labeled data to infer a function for future predictions
   - Classification (Self-Driving Cars)
   - Regression (Stock price prediction, Algorithmic trading)
2. **Unsupervised Machine Learning**
   - Does not make use of label information
   - Learns structures in high dimensional data using only the unlabeled data features
   - Clustering (Customer Segmentation and Recommendation System)
3. **Semi-Supervised Machine Learning (SSL)**
   - Makes use of both label information and unlabeled data features
   - Classification (More on this later)
   - Regression (Draw Inferences for Demographic Information from limited Census Data)

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels.

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels. The goal is to learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n. \tag{1}$$

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels. The goal is to learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n. \tag{1}$$

Semi-supervised learning: In semi-supervised learning, we are additionally given a (usually large) amount of unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$.

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels. The goal is to learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n. \tag{1}$$

Semi-supervised learning: In semi-supervised learning, we are additionally given a (usually large) amount of unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$. Goal is to use the unlabeled data to aid the learning.

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels. The goal is to learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n. \tag{1}$$

Semi-supervised learning: In semi-supervised learning, we are additionally given a (usually large) amount of unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$. Goal is to use the unlabeled data to aid the learning.

1. **Inductive learning:** Learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n.$$

Mathematical Framework to Learning

Fully supervised: In fully supervised learning, we are given training data (x_i, y_i) for $i = 1, \ldots, n$, where $x_i \in \mathcal{X}$ are the data points and $y_i \in \mathcal{Y}$ are the known labels. The goal is to learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n. \tag{1}$$

Semi-supervised learning: In semi-supervised learning, we are additionally given a (usually large) amount of unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$. Goal is to use the unlabeled data to aid the learning.

1. **Inductive learning:** Learn a function

$$u : \mathcal{X} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n.$$

2. **Transductive learning:** Learn a function

$$u : \{x_1, x_2, \ldots, x_{n+m}\} \to \mathcal{Y} \quad \text{for which } u(x_i) \approx y_i \text{ for } i = 1, \ldots, n$$

Motivation for Semi-Supervised Learning

Motivation for Semi-Supervised Learning

Why do we want to use semi-supervised learning?

Motivation for Semi-Supervised Learning

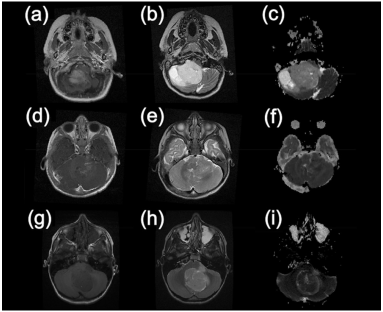
Why do we want to use semi-supervised learning?

It is oftentimes **expensive and labor intensive** to obtain labeled data, and we normally have an **abundance** of unlabeled data.

Motivation for Semi-Supervised Learning

Why do we want to use semi-supervised learning?

It is oftentimes **expensive and labor intensive** to obtain labeled data, and we normally have an **abundance** of unlabeled data.



Heuristic Reasons for Semi Supervised Learning to Work

Heuristic Reasons for Semi Supervised Learning to Work

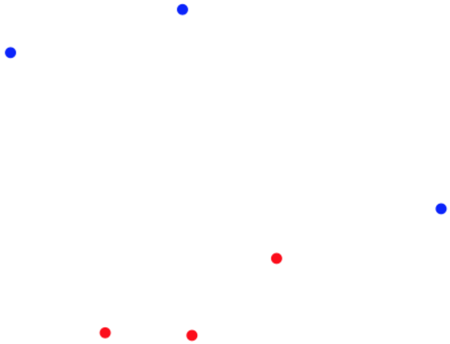


Figure: Three labels per class is given.

Heuristic Reasons for Semi-Supervised Learning to Work

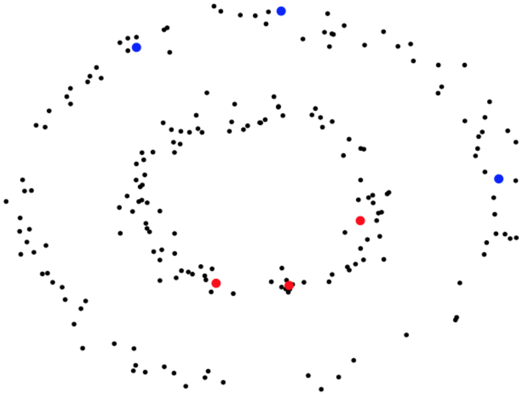


Figure: When both labeled and unlabeled data is given.

Outline

Mathematics of Graph-Based Semi-Supervised Learning

Mathematics of Graph-Based Semi-Supervised Learning

Given:

Mathematics of Graph-Based Semi-Supervised Learning

Given:

1. Graph $(\mathcal{X}, \mathcal{W})$ where

Mathematics of Graph-Based Semi-Supervised Learning

Given:

1. Graph $(\mathcal{X}, \mathcal{W})$ where
   - $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
   - $\mathcal{W} = (w_{xy})_{x,y \in \mathcal{X}}$ are the **nonnegative** edge weights.
   - $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.

Mathematics of Graph-Based Semi-Supervised Learning

Given:

1. Graph $(\mathcal{X}, \mathcal{W})$ where
   - $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
   - $\mathcal{W} = (w_{xy})_{x,y \in \mathcal{X}}$ are the **nonnegative** edge weights.
   - $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.

2. Labeled (or observed) vertices are a subset $\Gamma \subseteq \mathcal{X}$.

Mathematics of Graph-Based Semi-Supervised Learning

Given:

1. Graph $(\mathcal{X}, \mathcal{W})$ where
   - $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
   - $\mathcal{W} = (w_{xy})_{x,y\in\mathcal{X}}$ are the **nonnegative** edge weights.
   - $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.
2. Labeled (or observed) vertices are a subset $\Gamma \subseteq \mathcal{X}$.
3. We are given a labelling function $g : \Gamma \to \mathbf{R}^k$ such that the i^{th} class has label vector $g(x) = e_i = (0, \cdots, 0, 1, 0, \cdots, 0)$.

Task: Extend the labels from Γ to the entire graph $\mathcal{X}$.

Mathematics of Graph-Based Semi-Supervised Learning

Given:

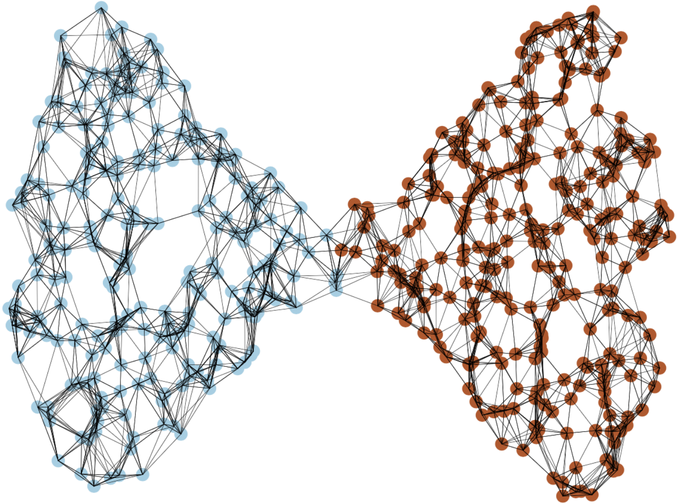
1. Graph $(\mathcal{X}, \mathcal{W})$ where
   - $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
   - $\mathcal{W} = (w_{xy})_{x,y \in \mathcal{X}}$ are the **nonnegative** edge weights.
   - $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.
2. Labeled (or observed) vertices are a subset $\Gamma \subseteq \mathcal{X}$.
3. We are given a labelling function $g : \Gamma \to \mathbf{R}^k$ such that the i^{th} class has label vector $g(x) = e_i = (0, \cdots, 0, 1, 0, \cdots, 0)$.

Task: Extend the labels from Γ to the entire graph $\mathcal{X}$.

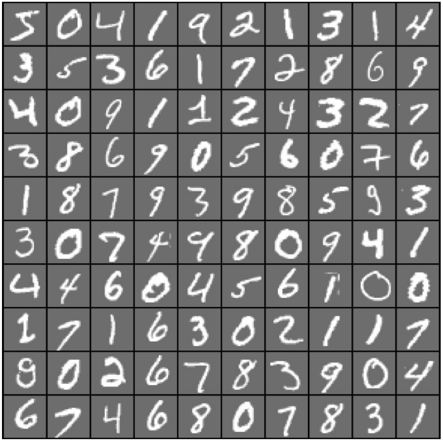
Semi-supervised smoothness assumption

Similar points $x, y \in \mathcal{X}$ in **high density** regions of the graph should have similar labels.

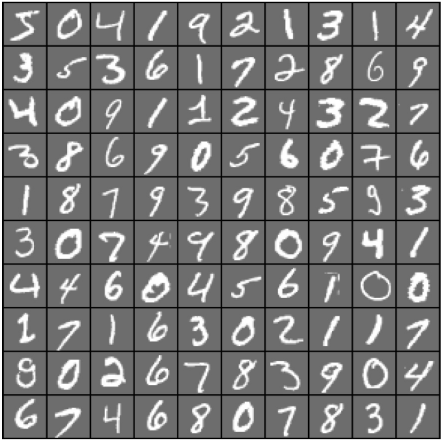
Example graph



MNIST (70,000 28×28 pixel images of digits 0-9)



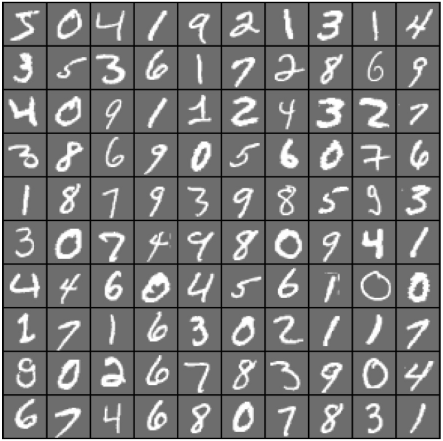
MNIST (70,000 28×28 pixel images of digits 0-9)



- Each image is a datapoint

$$x \in \mathbf{R}^{28\times 28} = \mathbf{R}^{784}.$$

MNIST (70,000 28×28 pixel images of digits 0-9)



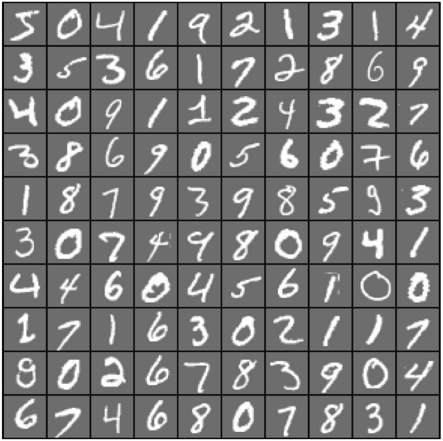
- Each image is a datapoint

$$x \in \mathbf{R}^{28\times 28} = \mathbf{R}^{784}.$$

- Geometric weights:

$$w_{xy} = \Phi\left(\frac{|x-y|}{\varepsilon}\right).$$

MNIST (70,000 28×28 pixel images of digits 0-9)



- Each image is a datapoint

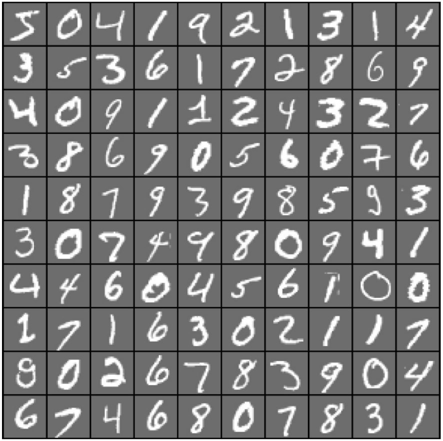
$$x \in \mathbf{R}^{28\times 28} = \mathbf{R}^{784}.$$

- Geometric weights:

$$w_{xy} = \Phi\left(\frac{|x-y|}{\varepsilon}\right).$$

An example of Φ is $\Phi(x) = e^{-x}$.

MNIST (70,000 28×28 pixel images of digits 0-9)



- Each image is a datapoint

$$x \in \mathbf{R}^{28\times 28} = \mathbf{R}^{784}.$$

- Geometric weights:

$$w_{xy} = \Phi\left(\frac{|x-y|}{\varepsilon}\right).$$

An example of Φ is $\Phi(x) = e^{-x}$.

- k-nearest neighbor graph:

$$w_{xy} = \Phi\left(\frac{|x-y|}{\varepsilon_k(x)}\right).$$

Outline

Laplacian Regularization

Laplacian regularized semi-supervised learning solves the Laplace equation

$$\begin{cases} \mathcal{L}u = 0 & \text{in } \mathcal{X} \setminus \Gamma, \\ u = g & \text{on } \Gamma, \end{cases}$$

where $u : \mathcal{X} \to \mathbb{R}^k$, and $\mathcal{L}$ is the graph Laplacian

$$\mathcal{L}u(x) = \sum_{y \in \mathcal{X}} w_{xy}(u(x) - u(y)).$$

Laplacian Regularization

Laplacian regularized semi-supervised learning solves the Laplace equation

$$\begin{cases} \mathcal{L}u = 0 & \text{in } \mathcal{X} \setminus \Gamma, \\ u = g & \text{on } \Gamma, \end{cases}$$

where $u : \mathcal{X} \to \mathbb{R}^k$, and $\mathcal{L}$ is the graph Laplacian

$$\mathcal{L}u(x) = \sum_{y \in \mathcal{X}} w_{xy}(u(x) - u(y)).$$

The label decision for vertex $x \in \mathcal{X}$ is determined by the largest component of $u(x)$

$$\ell(x) = \operatorname*{argmax}_{j \in \{1,\ldots,k\}} \{u_j(x)\}.$$

References:

- Original work [Zhu et al., 2003]
- Learning [Zhou et al., 2005, Ando and Zhang, 2007]
- Manifold ranking [He et al., 2006, Zhou et al., 2011, Xu et al., 2011]

Variational interpretation

Laplace learning is equivalent to the variational problem

$$\min_{u:\mathcal{X}\to\mathbb{R}^k} \left\{ \sum_{x,y\in\mathcal{X}} w_{xy}|u(x)-u(y)|^2 \,:\, u(x)=g(x) \text{ for all } x\in\Gamma \right\}.$$

Variational interpretation

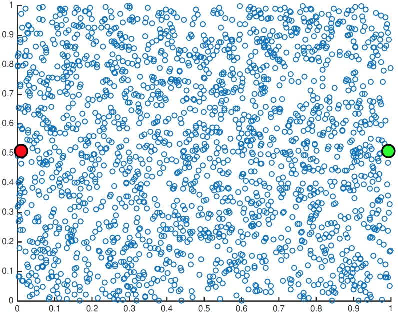
Laplace learning is equivalent to the variational problem

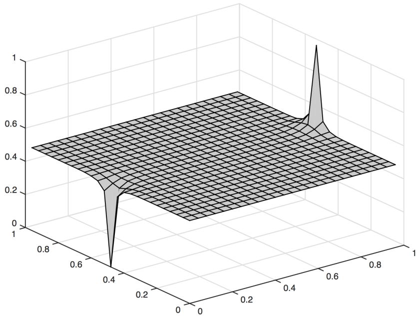
$$\min_{u:\mathcal{X}\to\mathbb{R}^k} \left\{ \sum_{x,y\in\mathcal{X}} w_{xy}|u(x)-u(y)|^2 \,:\, u(x)=g(x) \text{ for all } x\in\Gamma \right\}.$$

Many soft-constrained versions have been proposed

$$\min_{u:\mathcal{X}\to\mathbb{R}^k} \left\{ \sum_{x,y\in\mathcal{X}} w_{xy}|u(x)-u(y)|^2 + \lambda \sum_{x\in\Gamma} \ell(u(x),g(x)) \right\}.$$

Ill-posed with small amount of labeled data

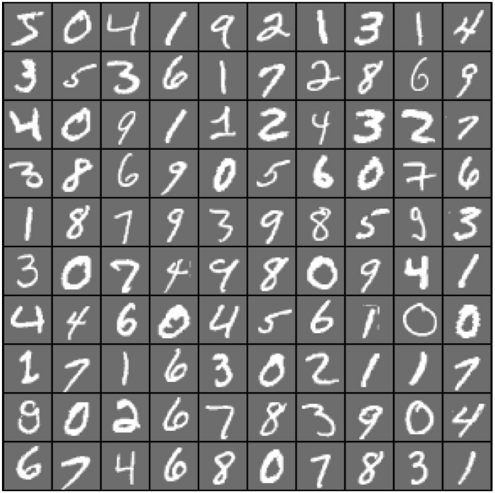




- Graph is $n = 10^5$ i.i.d. random variables uniformly drawn from $[0, 1]^2$.
- $w_{xy} = 1$ if $|x - y| < 0.01$ and $w_{xy} = 0$ otherwise.
- Two labels: $g(x) = 0$ at the Red point and $g(x) = 1$ at the Green point.

[Nadler et al., 2009]

MNIST (70,000 28×28 pixel images of digits 0-9)



[Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.]

Laplace learning on MNIST at low label rates

# Labels per class	**1**	**2**	**3**	**4**	**160**
Laplace Learning	16.1 (6.2)	28.2 (10.3)	42.0 (12.4)	57.8 (12.3)	97.0 (0.1)
Nearest Neighbor	65.4 (5.2)	74.2 (3.3)	77.8 (2.6)	80.7 (2.0)	92.4 (0.2)

- Average accuracy over 100 trials with standard deviation in brackets.
- Nearest neighbor is geodesic graph-nearest neighbor.

Recent work

The low-label rate problem was originally identified in [Nadler 2009].

A lot of recent work has attempted to address this issue with new graph-based classification algorithms at low label rates.

- Higher-order regularization: [Zhou and Belkin, 2011], [Dunlop et al., 2019]
- p-Laplace regularization: [Alaoui et al., 2016], [Calder 2018,2019], [Slepcev & Thorpe 2019]
- Re-weighted Laplacians: [Shi et al., 2017], [Calder & Slepcev, 2019]
- Centered kernel method: [Mai & Couillet, 2018]
- Poisson learning: [Calder, Cook, Thorpe, Slepcev, 2020]

Outline

Higher Order Poisson Learning

We propose the higher order Poisson (HOP) learning

$$\boxed{\mathcal{L}^m u(x) = \sum_{y \in \Gamma} (g(y) - \overline{g})\delta_{xy},}$$

subject to $\sum_{x \in \mathcal{X}} u(x) = 0$, where $\overline{g} = \frac{1}{|\Gamma|}\sum_{y \in \Gamma} g(y)$ and $m \in \mathbf{Z}^+$.

Higher Order Poisson Learning

We propose the higher order Poisson (HOP) learning

$$\boxed{\mathcal{L}^m u(x) = \sum_{y\in\Gamma}(g(y)-\overline{g})\delta_{xy},}$$

subject to $\sum_{x\in\mathcal{X}} u(x) = 0$, where $\overline{g} = \frac{1}{|\Gamma|}\sum_{y\in\Gamma} g(y)$ and $m \in \mathbf{Z}^+$.

The label decision is the same as before:

$$\ell(x) = \underset{j\in\{1,\ldots,k\}}{\operatorname{argmax}}\,\{u_j(x)\}.$$

Variational Interpretation

Variational Interpretation

The higher order Poisson learning problem is equivalent to the variational problem

$$(2) \qquad \min_{u \in \ell_0^2(\mathcal{X})} \left\{ \frac{1}{2}(u, \mathcal{L}^m u) - \sum_{x \in \Gamma} (g(x) - \overline{g}) \cdot u(x) \right\}.$$

where $\overline{g} = \frac{1}{|\Gamma|} \sum_{x \in \Gamma} g(x)$ is the average label vector.

Variational Interpretation

The higher order Poisson learning problem is equivalent to the variational problem

$$\min_{u \in \ell_0^2(\mathcal{X})} \left\{ \frac{1}{2}(u, \mathcal{L}^m u) - \sum_{x \in \Gamma} (g(x) - \overline{g}) \cdot u(x) \right\}. \tag{2}$$

where $\overline{g} = \frac{1}{|\Gamma|} \sum_{x \in \Gamma} g(x)$ is the average label vector.

Theorem 1 (Existence and Uniqueness of Solution for HOP Learning)

Assume that $\mathcal{G}$ is connected. Then there exists a unique minimizer $u \in \ell_0^2(\mathcal{X})$ of (2) and furthermore it satisfies the Euler-Lagrange equation for higher order Poisson learning

$$\mathcal{L}^m u(x) = \sum_{y \in \Gamma} (g(y) - \overline{g}) \delta_{xy},$$

subject to $\sum_{x \in \mathcal{X}} u(x) = 0$, where $\overline{g} = \frac{1}{|\Gamma|} \sum_{y \in \Gamma} g(y)$ and $m \in \mathbf{Z}^+$.

Preliminary Results

Preliminary Results

Lemma 2

Suppose that the graph Laplacian operator $\mathcal{L}$ is defined on a connected graph $\mathcal{G}$. The kernel of the graph Laplacian operator is the set of constant function on $\mathcal{X}$. In other words, we have

$$\ker(\mathcal{L}) = \{v \in \ell^2(\mathcal{X}) : v(x) = v(y) \text{ for all } x, y \in \mathcal{X}\}.$$

Preliminary Results

Lemma 2

Suppose that the graph Laplacian operator $\mathcal{L}$ is defined on a connected graph $\mathcal{G}$. The kernel of the graph Laplacian operator is the set of constant function on $\mathcal{X}$. In other words, we have

$$\ker(\mathcal{L}) = \{v \in \ell^2(\mathcal{X}) : v(x) = v(y) \text{ for all } x, y \in \mathcal{X}\}.$$

Lemma 3

The range of the graph Laplacian operator is the set of unweighted mean-zero functions on $\mathcal{X}$. In other words, we have

$$\operatorname{range}(\mathcal{L}) = \mathcal{L}(\mathcal{X}) = \{v \in \ell^2(\mathcal{X}) : \sum_{i=1}^{n} v(x_i) = 0\},$$

where $\mathcal{L}$ is the graph Laplacian operator and $\mathcal{X}$ is explicitly written out as $\mathcal{X} := \{x_1, \cdots, x_n\}$ with $n \in \mathbf{Z}^+$.

Preliminary Results

Lemma 4

Let $\mathcal{L}$ be the graph Laplacian operator, $\mathcal{X}$ be explicitly written as $\mathcal{X} := \{x_1, \cdots, x_n\}$ with $n \in \mathbf{Z}^+$, and $m \in \mathbf{Z}^+$ be given. Then we have

$$\ker(\mathcal{L}^m) = \ker(\mathcal{L}) = \{v \in \ell^2(\mathcal{X}) : v(x) = v(y) \text{ for all } x, y \in \mathcal{X}\},$$

and

$$\operatorname{range}(\mathcal{L}^m) = \operatorname{range}(\mathcal{L}) = \{v \in \ell^2(\mathcal{X}) : \sum_{i=1}^{n} v(x_i) = 0\}.$$

In other words, the kernel and range of the higher order Laplacian is the same as the kernel and range of the graph Laplacian.

Sketch of the Proof for the Theorem

"Proof:" Consider the Euler Lagrange equation for HOP learning problem

$$\mathcal{L}^m u(x) = \sum_{y \in \Gamma} (g(y) - \overline{g})\delta_{xy} = \begin{cases} g(x) - \overline{g}, & \text{when } x \in \Gamma, \\ 0, & \text{when } x \notin \Gamma. \end{cases}$$

Thus the existence of solution as range of $\mathcal{L}^m$ is the set of mean zero functions.

The solution is unique up to a constant function as the kernel of $\mathcal{L}^m$ is a set of constant functions.

The variational interpretation then comes naturally by taking the variations of the energy term

$$E(u) = \frac{1}{2}(u, \mathcal{L}^m u) - \sum_{x \in \Gamma} (g(x) - \overline{g}) \cdot u(x). \quad \square$$

Outline

Algorithm Development

To solve the Euler-Lagrange equation for higher order Poisson learning

$$\mathcal{L}^m u(x) = \sum_{y \in \Gamma} (g(y) - \overline{g})\delta_{xy},$$

we need to construct the inverse matrix $(\mathcal{L}^m)^{-1}$. This can be done through spectral numerical approximation.

Outline

Algorithm Development

Theorem 5 (Spectral Numerical Approximation)

The solution of the Euler-Lagrange equation for higher order Poisson learning

$$\mathcal{L}^m u(x) = \sum_{y\in\Gamma}(g(y)-\overline{g})\delta_{xy}$$

can be approximated by

$$u(x) \approx (\mathcal{L}_N^m)^{-1}\sum_{y\in\Gamma}(g(y)-\overline{g})\delta_{xy}$$

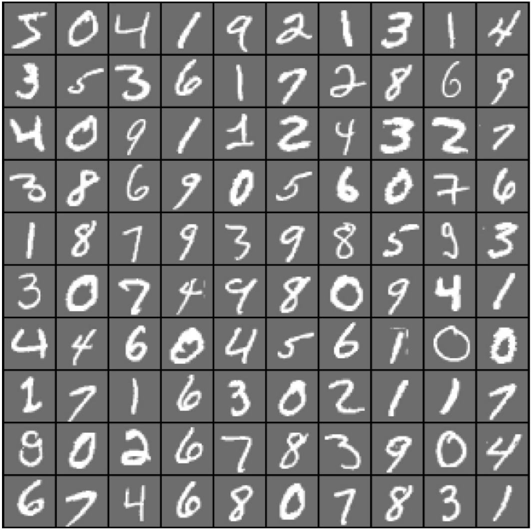
where

$$(\mathcal{L}_N^m)^{-1} = V_N \Lambda_{N,m} V_N^T$$

with V_N being the matrix of first N (in terms of the ascending order of eigenvalues) eigenvectors as its columns and $\Lambda_{N,m} =$ diag$((\lambda_i^{-m})_{i=1,\cdots,N})$ being the diagonal matrix with diagonal entries equal to the inverse of the first N eigenvalues.

Outline

MNIST

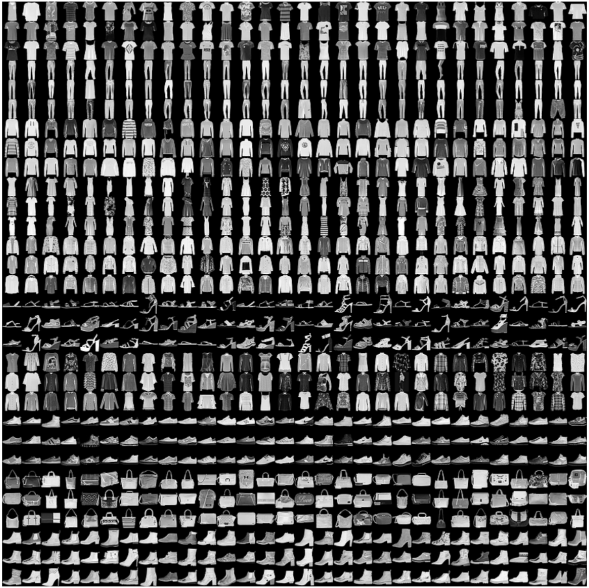


Simulation Results (MNIST)

Figure: Mean(Standard Deviation) Classification Accuracy over 100 Trials

# Labels per class	**1**	**2**	**3**	**4**	**5**
HOP with $N = 20$ & $m = 1.0$	**91.6 (3.9)**	**94.3 (1.6)**	**95.0 (0.8)**	95.1 (0.6)	95.3 (0.5)
HOP with $N = 30$ & $m = 1.0$	90.9 (3.8)	94.3 (1.5)	94.9 (1.1)	**95.3 (0.9)**	**95.6 (0.8)**
Poisson	90.2 (4.0)	93.6 (1.6)	94.5 (1.1)	94.9 (0.8)	95.3 (0.7)

Fashion-MNIST

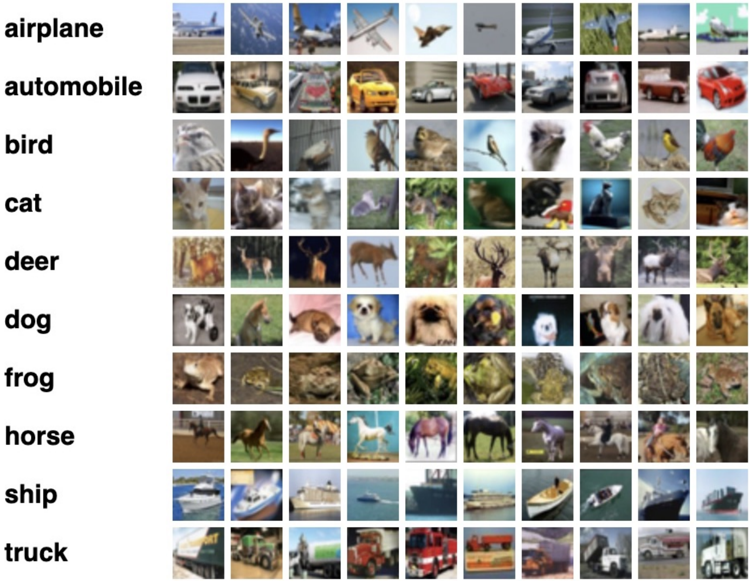


Simulation Results (Fashion-MNIST)

Figure: Mean(Standard Deviation) Classification Accuracy over 100 Trials

# Labels per class	**1**	**2**	**3**	**4**	**5**
HOP with $N = 10$ & $m = 0.4$	**61.7 (5.2)**	65.6 (4.1)	68.5 (2.6)	69.5 (2.2)	69.7 (2.1)
HOP with $N = 20$ & $m = 0.7$	60.7 (4.8)	**65.8 (4.4)**	69.4 (2.8)	70.6 (2.6)	71.4 (2.3)
HOP with $N = 30$ & $m = 0.8$	59.9 (4.7)	65.6 (4.2)	**69.5 (2.7)**	70.8 (2.5)	72.0 (2.3)
HOP with $N = 30$ & $m = 0.7$	59.6 (4.7)	65.5 (4.2)	69.4 (2.7)	**70.9 (2.5)**	**72.1 (2.2)**
Poisson	60.8 (4.6)	**66.1 (3.9)**	**69.6 (2.6)**	**71.2 (2.2)**	**72.4 (2.3)**

Cifar-10

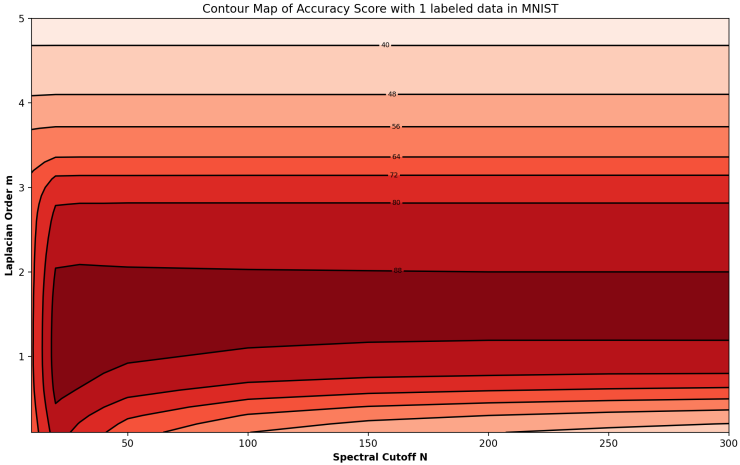

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

Simulation Results (Cifar-10)

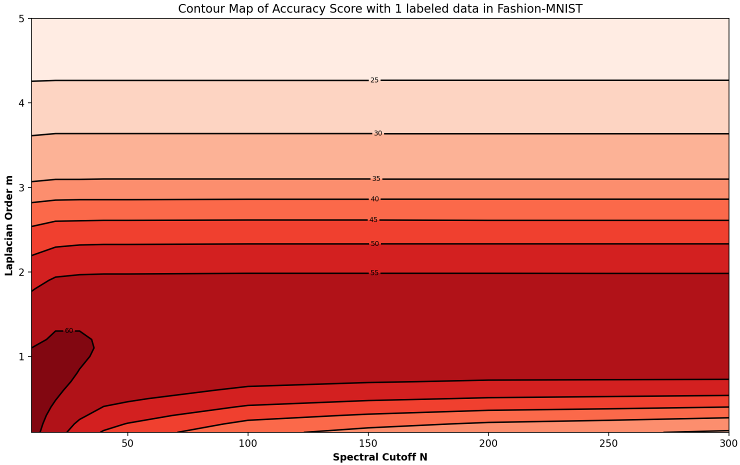
Figure: Mean(Standard Deviation) Classification Accuracy over 100 Trials

# Labels per class	**1**	**2**	**3**	**4**	**5**
HOP with $N = 20$ & $m = 0.3$	**41.8 (5.5)**	47.7 (5.3)	51.0 (3.9)	53.1 (3.1)	54.4 (2.7)
HOP with $N = 20$ & $m = 0.1$	41.7 (5.5)	**47.8 (5.3)**	**51.0 (3.8)**	**53.2 (3.1)**	**54.5 (2.7)**
Poisson	40.7 (5.5)	46.5 (5.1)	49.9 (3.4)	52.3 (3.1)	53.8 (2.6)

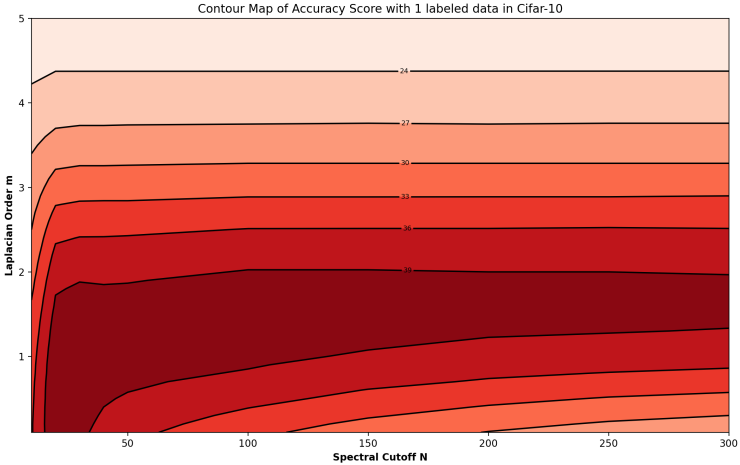
Contour Graphs for Simulation Results (MNIST)



Contour Graphs for Simulation Results (Fashion-MNIST)



Contour Graphs for Simulation Results (Cifar-10)



Outline

Current and Future Works

Current and Future Works

We observe that:

Current and Future Works

We observe that:

- The lower the spectral cutoff N is, the better result we have in terms of classification accuracy in general (Spectral Embedding and Clustering).
- The combination of the spectral cutoff N and the Laplacian operator order m plays a great role in the classfication accuracy score.

Current and Future Works

We observe that:

- The lower the spectral cutoff N is, the better result we have in terms of classification accuracy in general (Spectral Embedding and Clustering).
- The combination of the spectral cutoff N and the Laplacian operator order m plays a great role in the classfication accuracy score.

Future works may want to investigate for a criterion to find the best combination between the spectral cutoff N and the Laplacian operator order m.

References

References:

1. J. Calder, D. Slepčev, D., and M. Thorpe. **Rates of convergence for Laplacian semi-supervised learning with low label rates.** *arXiv:2006.02765*, 2020.
2. J. Calder, B. Cook, M. Thorpe, and D. Slepčev. **Poisson Learning: Graph based semi-supervised learning at very low label rates.** *International Conference on Machine Learning (ICML), PMLR 119:1306–1316*, 2020.
3. D. Bian. **Higher Order Poisson Regularization for Graph-Based Semi-Supervised Learning.** University of Minnesota Honors Thesis in Mathematics, 2021.

Code: `https://github.com/DingjunB/Honors-Thesis.git`

Outline

Alamgir, M. and Luxburg, U. V. (2011).
Phase transition in the family of p-resistances.
In *Advances in Neural Information Processing Systems*, pages 379–387.

Amghibech, S. (2003).
Eigenvalues of the discrete p-Laplacian for graphs.
Ars Combinatoria, 67:283–302.

Ando, R. K. and Zhang, T. (2007).
Learning on graph with Laplacian regularization.
In *Advances in Neural Information Processing Systems*, pages 25–32.

Bridle, N. and Zhu, X. (2013).
p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data.
In *Eleventh Workshop on Mining and Learning with Graphs (MLG2013)*.

Bruna, J. and Mallat, S. (2013).
Invariant scattering convolution networks.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1872–1886.

Bühler, T. and Hein, M. (2009).
Spectral clustering based on the graph p-Laplacian.
In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88. ACM.

Calder, J. (2019).
Consistency of Lipschitz learning with infinite unlabeled data and finite labeled data.
To appear in SIAM Journal on Mathematics of Data Science.

Calder, J. (2018).
The game theoretic p-Laplacian and semi-supervised learning with few labels.
Nonlinearity, 32(1).

Calder, J. and Slepčev, D. (2018).
Properly-weighted graph Laplacian for semi-supervised learning.
arXiv:1810.04351.
https://arxiv.org/abs/1810.04351.

Chapelle, O., Scholkopf, B., and Zien, A. (2006).
Semi-supervised learning.
MIT.

Chaudhari, P. and Soatto, S. (2018).
Stochastic gradient descent peforms variational inference, converges to limit cycles for deep networks.
arXiv:1710.11029.

El Alaoui, A., Cheng, X., Ramdas, A., Wainwright, M. J., and Jordan, M. I. (2016).
Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning.
In *Conference on Learning Theory*, pages 879–906.

Finlay, C., Abbasi, B., Calder, J., and Oberman, A. M. (2018).
Lipschitz regularized Deep Neural Networks generalize and are adversarially robust.
arXiv:1808.09540.

Flores, M., Calder, J., and Lerman, G. (2019).
Algorithms for Lp-based semi-supervised learning on graphs.
arXiv:1901.05031.

He, J., Li, M., Zhang, H.-J., Tong, H., and Zhang, C. (2006).
Generalized manifold-ranking-based image retrieval.
IEEE Transactions on Image Processing, 15(10):3170–3177.

Kyng, R., Rao, A., Sachdeva, S., and Spielman, D. A. (2015).
Algorithms for Lipschitz learning on graphs.
In *Conference on Learning Theory*, pages 1190–1223.

Luo, D., Huang, H., Ding, C., and Nie, F. (2010).
On the eigenvectors of p-Laplacian.
Machine Learning, 81(1):37–51.

Luxburg, U. v. and Bousquet, O. (2004).
Distance-based classification with lipschitz functions.
Journal of Machine Learning Research, 5(Jun):669–695.

Nadler, B., Srebro, N., and Zhou, X. (2009).
Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data.
Advances in Neural Information Processing Systems, 22:1330–1338.

Shi, Z., Osher, S., and Zhu, W. (2017).
Weighted nonlocal Laplacian on interpolation from sparse data.
Journal of Scientific Computing, 73(2-3):1164–1177.

Slepčev, D. and Thorpe, M. (2019).
Analysis of p-Laplacian regularization in semisupervised learning.
SIAM Journal on Mathematical Analysis, 51(3):2085–2120.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014).
Dropout: a simple way to prevent neural networks from overfitting.
The journal of machine learning research, 15(1):1929–1958.

Wang, Y., Cheema, M. A., Lin, X., and Zhang, Q. (2013).
Multi-manifold ranking: Using multiple features for better image retrieval.
In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 449–460. Springer.

Xu, B., Bu, J., Chen, C., Cai, D., He, X., Liu, W., and Luo, J. (2011).
Efficient manifold ranking for image retrieval.
In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 525–534. ACM.

Yang, C., Zhang, L., Lu, H., Ruan, X., and Yang, M.-H. (2013).
Saliency detection via graph-based manifold ranking.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3166–3173.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017).
Understanding deep learning requires rethinking generalization.
ICLR.

Zhou, D., Huang, J., and Schölkopf, B. (2005).
Learning from labeled and unlabeled data on a directed graph.
In *Proceedings of the 22nd International Conference on Machine Learning*, pages 1036–1043. ACM.

Zhou, D. and Schölkopf, B. (2005).
Regularization on discrete spaces.
In *Joint Pattern Recognition Symposium*, pages 361–368. Springer.

Zhou, X., Belkin, M., and Srebro, N. (2011).
An iterated graph Laplacian approach for ranking on manifolds.
In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 877–885. ACM.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003).
Semi-supervised learning using Gaussian fields and harmonic functions.
In *Proceedings of the 20th International Conference on Machine learning (ICML-03)*, pages 912–919.