PDE-inspired methods for graph-based semi-supervised learning

Jeff Calder

School of Mathematics
University of Minnesota

IMA Data Science Seminar
September 14, 2021

Joint work with:
Brendan Cook (UMN), Gilad Lerman (UMN), Peter J. Olver (UMN), Mauricio Flores Rios (Amazon), Dejan Slepčev (CMU), Matthew Thorpe (Manchester), and Katrina Yezzi-Woodley (UMN)

Research supported by NSF-DMS 1713691, 1944925, and the Alfred P. Sloan Foundation

Outline

Outline

Quick intro 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

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

Example: Automated image captioning

A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background

A little **girl** sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

A giraffe standing in a forest with **trees** in the background.

[Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep learning. Nature, 2015.]

Applications

Why is semi-supervised learning useful?

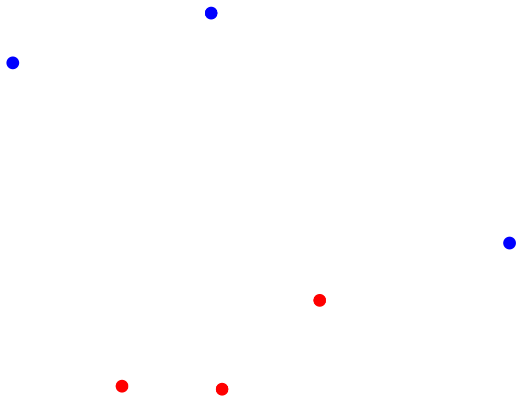
It is expensive to label data, and we have an abundance of unlabeled data.

Brief list of example applications:

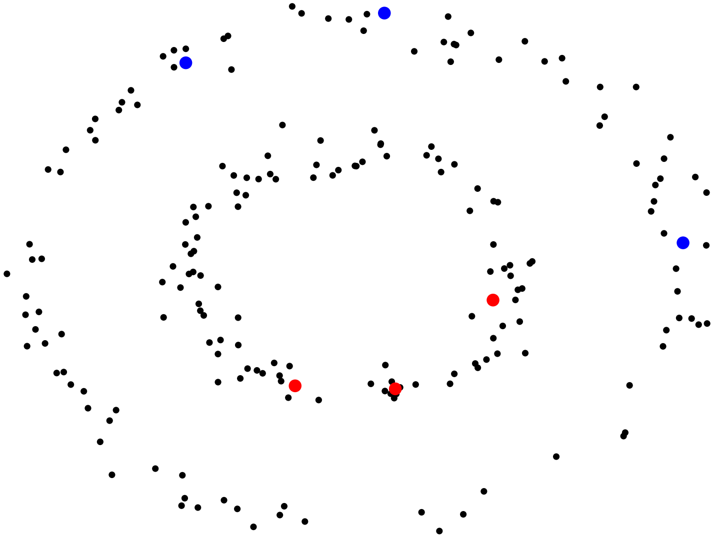
1. Speech recognition
2. Webpage classification
3. Inferring protein structure from sequencing

A great introductory book [Chapelle et al., 2006].

Why semi-supervised?



Why semi-supervised?



Outline

Graph-based semi-supervised learning

Model:

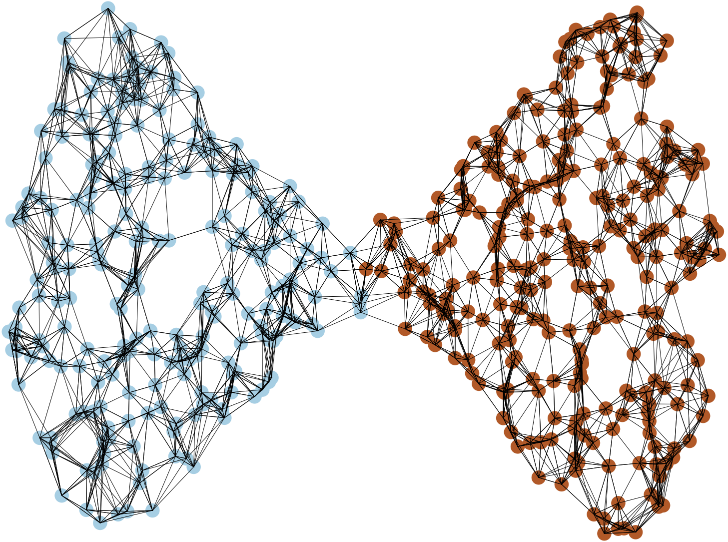
1. Data (labeled and unlabeled) is a graph $(\mathcal{X}, \mathcal{W})$.
   - $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
   - $\mathcal{W} = (w_{xy})_{x,y\in\mathcal{X}}$ are the nonnegative edge weights.
   - $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.
2. Labeled (or observed) vertices are a subset $\Gamma \subset \mathcal{X}$.
3. We given a labelling function $g : \Gamma \to \mathbb{R}$.

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

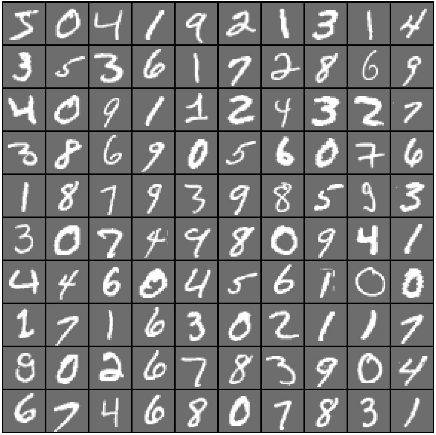
Semi-supervised smoothness assumption

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)



- Each image is a datapoint

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

- Geometric weights:

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

- k-nearest neighbor graph:

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

Clustering MNIST

https://divamgupta.com

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

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

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]

Label propagation

The solution of Laplace learning satisfies

$$\mathcal{L}u(x) = \sum_{y \in \mathcal{X}} w_{xy}(u(x) - u(y)) = 0. \qquad (y \in \mathcal{X} \setminus \Gamma)$$

Re-arranging, we see that u satisfies the mean-value property

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

Label propagation [Zhu 2005] iterates

$$u^{k+1}(x) = \frac{\sum_{y \in \mathcal{X}} w_{xy} u^k(y)}{\sum_{y \in \mathcal{X}} w_{xy}}.$$

and at convergence is equivalent to Laplace learning.

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

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

Random Walk Interpretation

Laplace learning also has a random walk interpretation

$$u(x) = \mathbb{E}[g(X_\tau)],$$

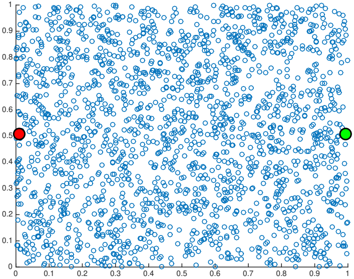
where $X_0 = x, X_1, X_2, X_3, \ldots$ is a random walk on $\mathcal{X}$ and

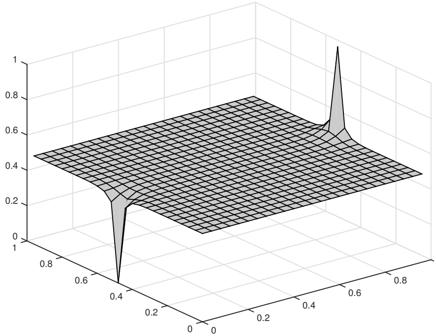
$$\tau = \inf\{k \geq 0 \,:\, X_k \in \Gamma\}.$$

The random walk satisfies

$$\mathbb{P}(X_{k+1} = y | X_k = x) = \frac{w_{xy}}{\sum_{z\in\mathcal{X}} w_{xz}}.$$

Ill-posed with small amount of labeled data

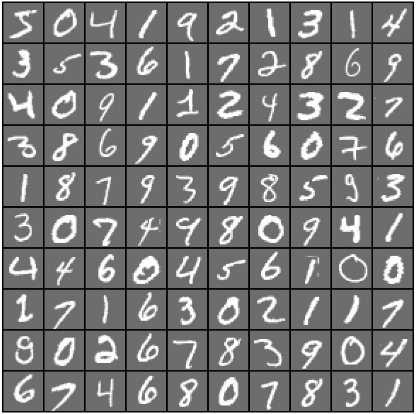




- Graph is $n = 10^5$ i.i.d. random variables uniformly drawn from $[0,1]^2$.
- $w_{xy} = 1$ if $|x - y| < 0.01$ and $w_{xy} = 0$ otherwise.
- Two labels: $g(x) = 0$ at the Red point and $g(x) = 1$ at the Green point.

[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]

Continuum perspective

Random Geometric Graph: Assume the vertices of the graph are

$$\mathcal{X}_n = \{x_1, \ldots, x_n\}$$

where $x_1, \ldots, x_n$ are a sequence of i.i.d. random variables on $\Omega \subset \mathbb{R}^d$ with positive density ρ, and the weights are given by

$$w_{xy} = \Phi\left(\frac{|x-y|}{\varepsilon}\right), \tag{2}$$

where $\Phi : [0,\infty) \to [0,1]$ is smooth with compact support. In particular, we assume

$$\begin{cases} \Phi(t) \geq 1, & \text{if } 0 \leq t \leq 1 \\ \Phi(t) = 0, & \text{if } t > 2 \\ \Phi(t) \geq 0, & \text{for all } t \geq 0. \end{cases}$$

Graph Connectivity

We say the graph $\mathcal{X}_n$ is connected if for every $x, y \in \mathcal{X}_n$ there is a path $y_1, y_2, \ldots, y_m \in \mathcal{X}_n$ with $y_1 = x$, $y_m = y$ and $w_{y_i, y_{i-1}} > 0$ for all $i = 2, \ldots, m$.

Fact: The graph is connected with probability at least

$$1 - n \exp\left(-Cn\varepsilon^d\right),$$

provided $n\varepsilon^d \geq c$.

The Laplace learning problem

$$\begin{cases} \mathcal{L}u(x) = 0, & \text{if } x \in \mathcal{X}_n \setminus \Gamma \\ u(x) = g(x), & \text{if } x \in \Gamma, \end{cases} \tag{3}$$

admits a unique solution when the graph is connected.

1. Uniqueness: Maximum principle (or strong convexity of graph Dirichlet energy).
2. Existence: The Perron method (or construct as minimizer of Dirichlet energy).

Continuum limit of graph Laplacian

Recall

$$\mathcal{L}u(x) = \sum_{i=1}^{n} \Phi\left(\frac{|x_i - x|}{\varepsilon}\right)(u(x_i) - u(x)).$$

Taking expectations we have

$$\begin{aligned}
\mathbb{E}[\mathcal{L}u(x)] &= n\int_{B(x,2\varepsilon)} \Phi\left(\frac{|y-x|}{\varepsilon}\right)(u(y) - u(x))\rho(y)\,dy \\
&= n\varepsilon^{d}\int_{B(0,2)} \Phi\left(|z|\right)(u(x+\varepsilon z) - u(x))\rho(x+\varepsilon z)\,dz \\
&= C_\Phi n\varepsilon^{d+2}\left(\frac{\rho(x)}{2}\Delta u(x) + \nabla\rho(x)\cdot\nabla u(x)\right) + O(n\varepsilon^{d+3}) \\
&= C_\Phi n\varepsilon^{d+2}\rho^{-1}\mathrm{div}(\rho^2\nabla u) + O(n\varepsilon^{d+3}).
\end{aligned}$$

Concentration of measure

Theorem (Bernstein's inequality)

Let $Y_1, \ldots, Y_n$ *be* *i.i.d.* *with mean* $\mu = \mathbb{E}[Y_i]$ *and variance* $\sigma^2 = \mathbb{E}[(Y_i - \mathbb{E}[Y_i])^2]$*, and assume* $|Y_i| \leq M$ *almost surely for all* i*. Then for any* $t > 0$

$$\mathbb{P}\left(\left|\sum_{i=1}^n Y_i - n\mu\right| > nt\right) \leq 2\exp\left(-\frac{nt^2}{2\sigma^2 + 4Mt/3}\right). \tag{4}$$

Here, $Y_i = \Phi\left(\frac{|x_i - x|}{\varepsilon}\right)(u(x_i) - u(x))$ so $|Y_i| \leq C\varepsilon$ and

$$\sigma^2 \leq \int_{B(x,2\varepsilon)} \Phi\left(\frac{|y-x|}{\varepsilon}\right)^2 (u(y) - u(x))^2 \rho(y)^2 \, dy \leq C\varepsilon^{d+2}.$$

Hence we have

$$\mathbb{P}(|\mathcal{L}u(x) - \mathbb{E}[\mathcal{L}u(x)]| > n\varepsilon^{d+2}\lambda) \leq 2\exp\left(-cn\varepsilon^{d+2}\lambda^2\right)$$

provided $0 < \lambda \leq \varepsilon^{-1}$.

Pointwise consistency of graph Laplacian

Combining the asymptotic expansion of $\mathbb{E}[\mathcal{L}u]$ and the concentration of measure we have

$$\mathbb{P}\left(\left|\mathcal{L}u(x) - C_\Phi n\varepsilon^{d+2}\rho^{-1}\mathrm{div}(\rho^2\nabla u)\right| > Cn\varepsilon^{d+2}(\lambda+\varepsilon)\right) \leq 2\exp\left(-cn\varepsilon^{d+2}\lambda^2\right)$$

for any $u \in C^3(\Omega)$.

Essentially this says that

$$\frac{1}{n\varepsilon^{d+2}}\mathcal{L}u(x) = C_\Phi\rho^{-1}\mathrm{div}(\rho^2\nabla u) + O(\varepsilon)$$

with very high probability.

Continuum perspective

From this continuum perspective, Laplace learning is a discretization of

$$(5) \qquad \begin{cases} \Delta u + 2\nabla \log \rho \cdot \nabla u = 0 & \text{in } \Omega \\ u = g & \text{on } \Gamma \\ \dfrac{\partial u}{\partial \nu} = 0 & \text{on } \partial\Omega, \end{cases}$$

where Ω is the support of ρ.

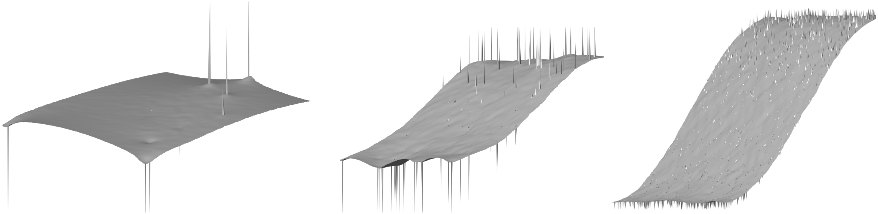
Laplace's equation is ill-posed without some restrictions on the boundary Γ:

- The Perron method requires Γ satisfy an exterior sphere condition.
- Sobolev space methods require boundary regularity to define the trace on Γ.

Takeaway: Laplace learning performs poorly for low labeling rates because it is a discretization of an ill-posed PDE.

Spikes in Laplacian regularized learning

Label function: $g(x) = \cos(x_1)$.



10 labels | 100 labels | 1000 labels

Q1 How many labels do we need to ensure that spikes do not form?

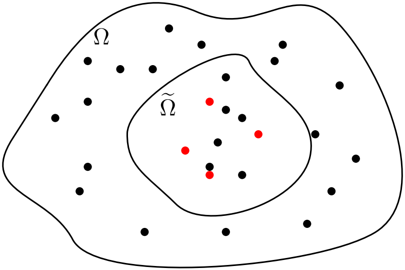
Q2 Why does Laplace learning perform poorly at low label rates?

- Are the spikes too localized? Do they propagate information globally?

Q3 How should we propagate labels in a stable and informative way?

Model for labeled data

Model 1. Let $\beta \in (0,1]$ and $\widetilde{\Omega} \subset\subset \Omega$. Each $x_i \in \widetilde{\Omega}$ is selected as training data independently with probability β. Let $\Gamma_n =$ training data.



The Laplacian learning problem is

$$(6) \quad \begin{cases} \mathcal{L}u_n(x) = 0, & \text{if } x \in \mathcal{X}_n \setminus \Gamma_n \\ u_n(x) = g(x), & \text{if } x \in \Gamma_n, \end{cases}$$

where $g : \Omega \to \mathbb{R}$ is Lipschitz and

$$\mathcal{X}_n = \{x_1, x_2, \ldots, x_n\}.$$

Main result

The continuum PDE is

$$
\begin{cases} \mathrm{div}(\rho^2 \nabla u) = 0 & \text{in } \Omega \setminus \widetilde{\Omega} \\ u = g & \text{on } \widetilde{\Omega} \\ \nabla u \cdot \mathbf{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{7}
$$

Theorem (C.-Slepcev-Thorpe, 2020)

Let $u_n : \mathcal{X}_n \to \mathbb{R}$ *be the solution of* (6), *and let* $u \in C^3(\overline{\Omega})$ *be the solution of* (7). *If* $\beta \geq \varepsilon^2$ *and* $\varepsilon \leq \lambda \leq c$ *then*

$$
\max_{x \in \mathcal{X}_n} |u_n(x) - u(x)| \leq C \left(\frac{\varepsilon}{\sqrt{\beta}} \log\left(\frac{\sqrt{\beta}}{\varepsilon} \right) + \lambda \right) \tag{8}
$$

holds with probability at least $1 - Cn \exp\left(-cn\varepsilon^{d+2}\lambda^2\right)$.

The negative result

Theorem (C.-Slepcev-Thorpe, 2020)

Assume that $\beta = \beta_n \to 0^+$ and $\varepsilon = \varepsilon_n \to 0^+$ satisfy

(9) $$\beta_n \ll \varepsilon_n^2, \quad \textit{and} \quad n\varepsilon_n^d \gg \log(n).$$

Then, with probability one, the sequence u_n is pre-compact in TL^2 and any convergent subsequence converges to a constant.

Summary: Laplace learning propagates labels well when

$$\text{Label rate } = \beta \gg \varepsilon^2.$$

Below this label rate, spikes form and the solution is degenerate.

Outline

ℓ_p-based Laplacian regularization

For any $p < \infty$:

(10) $$\min_{u:\mathcal{X}\to\mathbb{R}} \sum_{x,y\in\mathcal{X}} w_{xy}^p |u(x) - u(y)|^p \quad \text{subject to } u(x) = g(x) \text{ for all } x \in \Gamma.$$

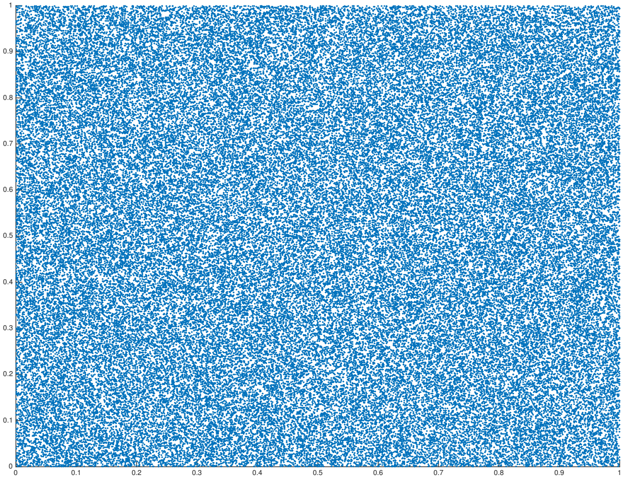
We can send $p \to \infty$:

(11) $$\min_{u:\mathcal{X}\to\mathbb{R}} \max_{x,y\in\mathcal{X}} \{w_{xy}|u(x) - u(y)|\} \quad \text{subject to } u(x) = g(x) \text{ for all } x \in \Gamma.$$

References:

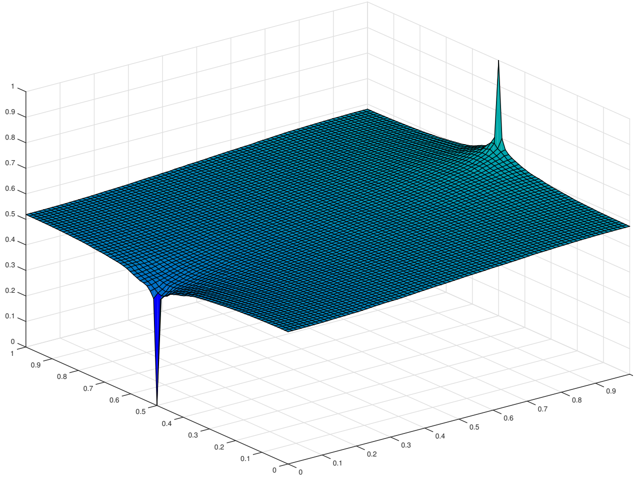
- Finite p:
  [Bridle and Zhu, 2013][Alamgir and Luxburg, 2011][El Alaoui et al., 2016]
- $p = \infty$: [Kyng et al., 2015] [Luxburg and Bousquet, 2004]
- Absolutely minimal Lipschitz extensions: Aronsson et al., 2004

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



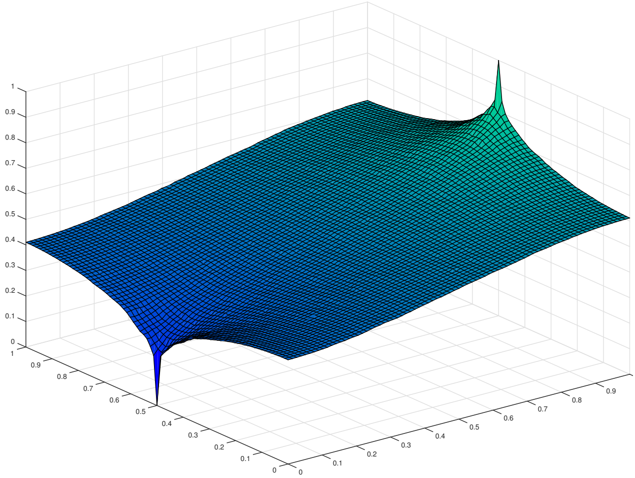
$p = 2$

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



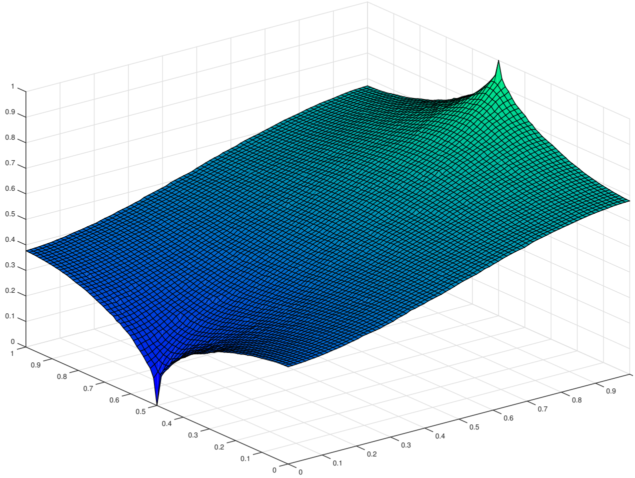
$p = 2$

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



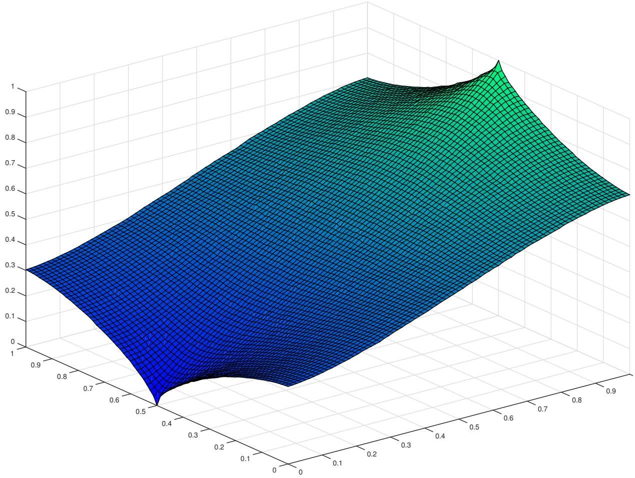
$p = 2.5$

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



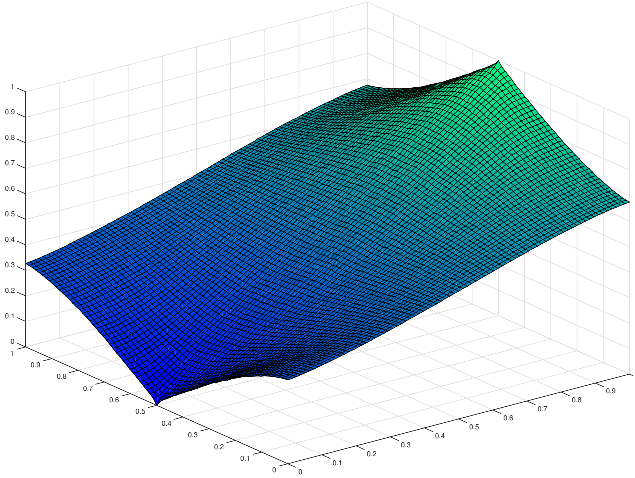
$p = 3$

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



$p = 5$

p-Laplacian learning: $n = 10^5$ points, $h = 10^{-2}$



$p = \infty$

$p = \infty$ minimizers nonunique

$p = \infty$ minimizers nonunique

$p = \infty$ minimizers nonunique

$p = \infty$ minimizers nonunique

For $p = \infty$ we take the minimizer that is smallest in the lexicographic ordering.

For $x, y \in \mathbb{R}^n$ with

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n$$

we say $x \preceq y$ in the lexicographic ordering if $x = y$ or

$$\exists j,\ x_j < y_j \text{ and } \forall i < j,\ x_i = y_i.$$

For general $x, y \in \mathbb{R}^n$ we sort components from smallest to largest before comparing.

$p = \infty$ minimizers nonunique

For $p = \infty$ we take the minimizer that is smallest in the lexicographic ordering.

For $x, y \in \mathbb{R}^n$ with

$$x_1 \leq x_2 \leq \cdots \leq x_n \quad \text{and} \quad y_1 \leq y_2 \leq \cdots \leq y_n$$

we say $x \preceq y$ in the lexicographic ordering if $x = y$ or

$$\exists j,\ x_j < y_j \text{ and } \forall i < j,\ x_i = y_i.$$

For general $x, y \in \mathbb{R}^n$ we sort components from smallest to largest before comparing.

Graph Laplacians

p-**Laplacian learning:**

$$\min_{u:\mathcal{X}_n\to\mathbb{R}} J_p(u) = \sum_{x,y\in\mathcal{X}_n} w_{xy}^p |u(x)-u(y)|^p \quad \text{subject to } u(x)=g(x) \text{ for } x\in\Gamma\subset\mathcal{X}_n$$

The minimizer $u:\mathcal{X}_n\to\mathbb{R}$ satisfies

$$\sum_{y\in\mathcal{X}_n} w_{xy}^p |u(y)-u(x)|^{p-2}(u(y)-u(x)) = 0 \quad \text{for } x\in\mathcal{X}_n\setminus\Gamma$$

and $u(x)=g(x)$ for $x\in\Gamma$.

References on graph p-Laplacian:

- [Zhou and Schölkopf, 2005] [Amghibech, 2003] [Bühler and Hein, 2009] [Luo et al., 2010]

Graph Laplacian as $p \to \infty$

Note that solutions of

$$\sum_{y \in \mathcal{X}_n} w_{xy}^p |u(y) - u(x)|^{p-2}(u(y) - u(x)) = 0$$

satisfy

$$\left(\sum_{\substack{y \in \mathcal{X}_n \\ u(y) \geq u(x)}} w_{xy}^p (u(y) - u(x))^{p-1} \right)^{1/p} = \left(\sum_{\substack{y \in \mathcal{X}_n \\ u(y) < u(x)}} w_{xy}^p (u(x) - u(y))^{p-1} \right)^{1/p}.$$

Send $p \to \infty$ to get

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

or

$$\mathcal{L}_\infty u(x) := \max_{y \in \mathcal{X}_n} w_{xy}(u(y) - u(x)) + \min_{y \in \mathcal{X}_n} w_{xy}(u(y) - u(x)) = 0.$$

Graph Laplacians

Lipschitz learning

$$\min_{u:\mathcal{X}_n\to\mathbb{R}} J_\infty(u) = \max_{x,y\in\mathcal{X}_n} w_{xy}|u(x)-u(y)| \quad \text{subject to } u(x)=g(x) \text{ for } x\in\Gamma\subset\mathcal{X}_n$$

The lex-minimizer $u:\mathcal{X}_n\to\mathbb{R}$ satisfies

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

where

$$\mathcal{L}_\infty u(x) := \max_{y\in\mathcal{X}_n} w_{xy}(u(y)-u(x)) + \min_{y\in\mathcal{X}_n} w_{xy}(u(y)-u(x)) = 0.$$

Reference:

1. [Kyng et al., 2015], [Calder, 2019]

Game theoretic p-Laplacian

Another natural way to regularize the 2-Laplacian is to add a small ∞-Laplace term:

$$\mathcal{L}u + \varepsilon \mathcal{L}_\infty u = 0.$$

At the continuum, we can expand the p-Laplacian

$$\Delta_p u := \mathrm{div}(|\nabla u|^{p-2}\nabla u) = |\nabla u|^{p-2}(\Delta u + (p-2)\Delta_\infty u).$$

So p-harmonic functions also satisfy

$$\Delta u + (p-2)\Delta_\infty u = 0.$$

which is called the game-theoretic, or homogeneous, p-Laplace equation.

Game theoretic p-Laplacian

We define the graph game theoretic p-Laplacian

$$\mathcal{L}_p u = \frac{1}{d_n}\mathcal{L}u + \lambda(p-2)\mathcal{L}_\infty u$$

where $d_n(x) = \sum_{y\in\mathcal{X}_n} w_{xy}$ and $\lambda = \lambda(\Phi)$.

A similar, but different, definition appears in [Manfredi et al, 2015.]

The game theoretic p-Laplacian for semi-supervised learning

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

Here,

$$\mathcal{X}_n = \Gamma \cup \{x_1, x_2, \ldots, x_n\}$$

where $x_1, x_2, \ldots, x_n$ i.i.d.$\sim \rho$ on $\mathbb{T}^d$ and $\Gamma \subset \mathbb{T}^d$ is a fixed collection of label points.

Game theoretic p-Laplacian

Theorem ([Calder, 2018])

Let $d < p < \infty$*, and suppose that* $\varepsilon_n \to 0$ *such that*

$$\lim_{n\to\infty} \frac{n\varepsilon_n^q}{\log(n)} = \infty, \tag{12}$$

where $q = \max\{d+4, 3d/2\}$*. Then with probability one*

$$u_n \longrightarrow u \quad \text{uniformly as} \quad n \to \infty, \tag{13}$$

where $u \in C^{0,\frac{p-d}{p-1}}(\mathbb{T}^d)$ *is the unique viscosity solution of the weighted* p*-Laplace equation*

$$\begin{cases} \mathrm{div}\left(\rho^2|\nabla u|^{p-2}\nabla u\right) = 0 & \text{in } \mathbb{T}^d \setminus \Gamma \\ u = g & \text{on } \Gamma. \end{cases} \tag{14}$$

Calder, J. (2018). **The game theoretic p-Laplacian and semi-supervised learning with few labels.** *Nonlinearity*, 32(1).
Flores, M., Calder, J., and Lerman, G. (2019). **Algorithms for Lp-based semi-supervised learning on graphs**. *arXiv:1901.05031*.

Outline

Lipschitz learning

Lipschitz learning solves

$$\min_{u:\mathcal{X}\to\mathbb{R}} \max_{x,y\in\mathcal{X}} \{w_{xy}|u(x)-u(y)|\} \quad \text{subject to } u(x)=g(x) \text{ for all } x\in\Gamma. \tag{15}$$

In the continuum this is consistent with the ∞-Laplace equation

$$\Delta_\infty u := \sum_{i,j=1}^{d} u_{x_i x_j} u_{x_i} u_{x_j} = 0$$

which is well-posed for any boundary Γ.

- Lipschitz learning proposed in [Kyng et al., 2015, Luxburg and Bousquet, 2004] and studied in [El Alaoui et al., 2016].
- Discrete to continuum convergence was proved in [Calder, 2019].
- Continuum PDE does not depend on ρ!

Lipschitz learning

Lipschitz learning can be re-tuned to be sensitive to ρ:

(16) $$\min_{u:\mathcal{X}\to\mathbb{R}} \max_{x,y\in\mathcal{X}} \{w_{xy}\, d(x)^\alpha d(y)^\alpha |u(x)-u(y)|\} \quad \text{subject to } u(x)=g(x) \text{ for all } x\in\Gamma,$$

where $d(x)=\sum_{y\in\mathcal{X}} w_{xy}$ is the degree (i.e., kernel density estimator)

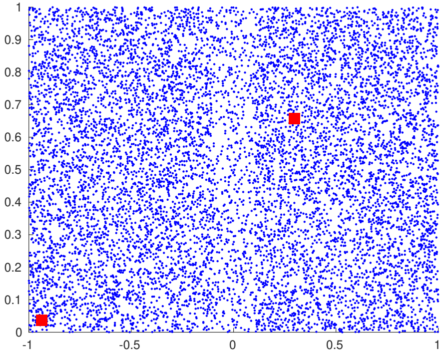
In the continuum this is consistent with the ∞-Laplace equation

$$\Delta_\infty u + 2\alpha\nabla\log\rho\cdot\nabla u = 0.$$

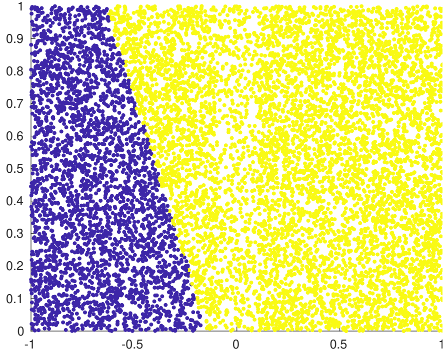
- The additional term $\nabla\log\rho\cdot\nabla u$ is an advection term that propagates labels along the gradient of the distribution.
- Continuum PDE is sensitive to distribution for $\alpha\neq 0$.

Calder, J. (2019). **Consistency of Lipschitz learning with infinite unlabeled data and finite labeled data.** *SIAM Journal on Mathematics of Data Science* 1(4):780–812.

Synthetic classification demo

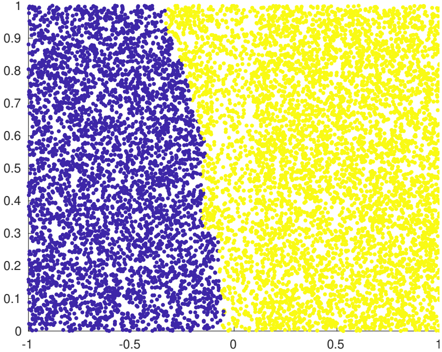


Synthetic classification demo



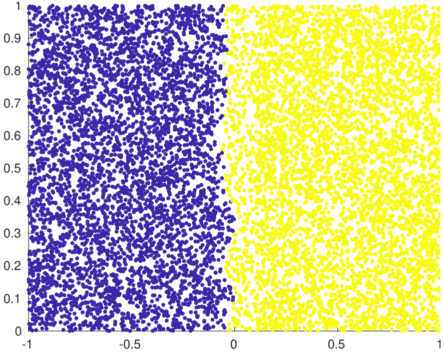
$\alpha = 0$, (81.5%)

Synthetic classification demo



$\alpha = 1$, (93.44%)

Synthetic classification demo



$\alpha = 2$, (98.88%)

Outline

Weighted Non-Local Laplacian (WNLL)

The Weighted Non-Local Laplacian (WNLL) [Shi et al., 2017] solves

$$\min_{u:\mathcal{X}\to\mathbb{R}} \sum_{x\in\mathcal{X}\setminus\Gamma} \sum_{y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2 + \mu \sum_{x\in\Gamma}\sum_{y\in\mathcal{X}} w_{xy}(g(x)-u(y))^2, \tag{17}$$

subject to $u = g$ on Γ, where

$$\mu = \text{Ratio of unlabeled to labeled data.}$$

Weighted Non-Local Laplacian (WNLL)

The Weighted Non-Local Laplacian (WNLL) [Shi et al., 2017] solves

$$\min_{u:\mathcal{X}\to\mathbb{R}} \sum_{x\in\mathcal{X}\setminus\Gamma}\sum_{y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2 + \mu \sum_{x\in\Gamma}\sum_{y\in\mathcal{X}} w_{xy}(g(x)-u(y))^2, \tag{18}$$

subject to $u = g$ on Γ, where

$$\mu = \text{Ratio of unlabeled to labeled data.}$$

- Gives better performance for few labels.
- However, still consistent with Laplace's equation in the continuum.
- [Calder and Slepčev, 2018] showed WNLL ill-posed for very low label rate problems.

Outline

Properly weighted Laplacian

Define

$$\gamma(x) = 1 + \left(\frac{r_0}{\mathrm{dist}(x,\Gamma)}\right)^{\alpha}, \tag{19}$$

where $\mathrm{dist}(x,\Gamma)$ denotes the Euclidean distance from x to the closest point in Γ.

For $\zeta > 1$ we set

$$\gamma(x) = \min\{\gamma(x), \zeta\}. \tag{20}$$

The Properly-Weighted learning problem is

$$\min_{u:\mathcal{X}\to\mathbb{R}} \sum_{x,y\in\mathcal{X}_n} \gamma(x) w_{xy} |u(x) - u(y)|^2 \quad \text{subject to } u(x) = g(x) \text{ for all } x \in \Gamma. \tag{21}$$

Calder, J. and Slepčev, D. (2019). **Properly-weighted graph Laplacian for semi-supervised learning**. *Applied Mathematics and Optimization: Special Issue on Optimization in Data Science*, 82:1111–1159.

Singularly-Weighted Sobolev Spaces

Recall

$$\gamma(x) = 1 + \left(\frac{r_0}{\mathrm{dist}(x,\Gamma)}\right)^{\alpha}, \tag{22}$$

For $u \in H^1(\Omega)$ we define

$$[u]^2_{H^1_\gamma(\Omega)} = \int_\Omega \gamma |\nabla u|^2 \, dx, \tag{23}$$

and

$$\|u\|^2_{H^1_\gamma(\Omega)} = \|u\|^2_{L^2(\Omega)} + [u]^2_{H^1_\gamma(\Omega)}. \tag{24}$$

We define

$$H^1_\gamma(\Omega) = \left\{ u \in H^1(\Omega) \, : \, \|u\|_{H^1_\gamma(\Omega)} < \infty \right\}. \tag{25}$$

We also denote by $H^1_{\gamma,0}(\Omega)$ the closure of $C^\infty_c(\overline{\Omega} \setminus \Gamma)$ in $H^1_\gamma(\Omega)$.

Traces at points

Theorem (Trace Theorem [Calder and Slepčev, 2018])

Assume $\alpha > d-2$. *Then the trace operator* $\mathsf{Tr} : H^1_\gamma(\Omega) \to \mathbb{R}^\Gamma$ *is bounded, and satisfies* $\mathsf{Tr}[u](x) = u(x)$ *whenever* u *is continuous at* $x \in \Gamma$. *Furthermore, for every* $u, v \in H^1_\gamma(\Omega)$ *with* $\|u-v\|^{2/(\alpha+2)}_{L^2(\Omega)} \leq R/2$ *we have*

$$|\,\mathsf{Tr}[u] - \mathsf{Tr}[v]| \leq C(1 + [u]_{H^1_\gamma(\Omega)} + [v]_{H^1_\gamma(\Omega)})\|u-v\|^{1-d/(\alpha+2)}_{L^2(\Omega)}. \tag{26}$$

It follows from the trace theorem that the continuum problem

$$\text{minimize } \frac{1}{2}\int_\Omega \gamma|\nabla u|^2\rho^2\,dx \text{ over } \left\{u \in H^1_\gamma(\Omega) \text{ and } u = g \text{ on } \Gamma\right\}. \tag{27}$$

is well-posed.

Lemma (Trace estimate)

Write $B_r = B(0,r)$*. If* $\alpha > d-2$ *then for any* $u \in C^\infty(B_1)$

$$|u(0) - (u)_r|^2 \leq C r^{\alpha+2-d} \int_{B_r} |x|^{-\alpha} |\nabla u|^2 \, dx,$$

where $(u)_r := \fint_{B_r} u \, dx$*.*

Sketch of proof.

By the Poincaré inequality

$$\begin{aligned} \fint_{B_r} (u - (u)_r)^2 \, dx &\leq C r^2 \fint_{B_r} |\nabla u|^2 \, dx \\ &\leq C r^2 \fint_{B_r} |x|^{-\alpha} r^\alpha |\nabla u|^2 \, dx \\ &= C r^{\alpha+2-d} \int_{B_r} |x|^{-\alpha} |\nabla u|^2 \, dx \ldots \end{aligned}$$

□

Euler-Lagrange Equation

The Euler-Lagrange equation satisfied by minimizers of the continuum problem is

$$
\begin{cases}
-\text{div}(\gamma\rho^2\nabla u) = 0 & \text{in } \Omega \setminus \Gamma \\
u = g & \text{on } \Gamma \\
\dfrac{\partial u}{\partial \nu} = 0 & \text{on } \partial\Omega.
\end{cases} \tag{28}
$$

Theorem ([Calder and Slepčev, 2018])

Let $\alpha > d-2$. The elliptic equation (28) has a unique weak solution $u \in H^1_\gamma(\Omega)$. Furthermore, $u \in C(\overline{\Omega}) \cap C^{2,\sigma}_{loc}(\overline{\Omega} \setminus \Gamma)$ and satisfies for every $0 < \beta < \alpha + 2 - d$

$$
|u(x) - u(y)| \leq C(\beta)|x-y|^\beta \quad (x \in \overline{\Omega}, y \in \Gamma). \tag{29}
$$

Proof uses the barrier $u(x) = |x|^{\alpha+2-d}$, which solves div $\left(|x|^{-\alpha}\nabla u\right) = 0$.

Continuity near labels

Let u_n be the unique solution of

$$\min_{u:\mathcal{X}_n\to\mathbb{R}} \sum_{x,y\in\mathcal{X}_n} \gamma_n(x) w_{xy} |u(x)-u(y)|^2 \quad \text{subject to } u(x)=g(x) \text{ for all } x\in\Gamma. \tag{30}$$

where we recall $\gamma_n(x) = \min\{\gamma(x), \zeta_n\}$.

Theorem (Hölder estimate [Calder and Slepčev, 2018])

Assume $\alpha > d-2$, $\zeta_n \geq 1+\varepsilon_n^{-\alpha}$ and fix $0<\beta<\alpha+2-d$. For each $z\in\Gamma$ the event that

$$|u_n(x)-u_n(z)| \leq C|x-z|^\beta + Cn^{1/2}\varepsilon_n^{1+\alpha/2} \tag{31}$$

holds for all $x\in\mathcal{X}_n$ occurs with probability at least $1-C\exp\left(-cn\varepsilon_n^{d+4}+\log(n)\right)$.

Proof uses barrier $|x|^{2+\alpha-d}$ adapted to graph.

Continuum limit

Theorem ([Calder and Slepčev, 2018])

Let $\alpha > d - 2$, $d \geq 3$ *and* $\varepsilon_n \to 0$, $\zeta_n \to \infty$ *such that*

$$\lim_{n\to\infty} \frac{n\varepsilon_n^d}{\log(n)} = \infty \quad \textit{and} \quad \lim_{n\to\infty} \frac{\zeta_n}{n\varepsilon_n^2} = \infty.$$

Then almost surely $u_n \to u$ *(in* TL^2*) where* u *is the solution of the continuum problem*

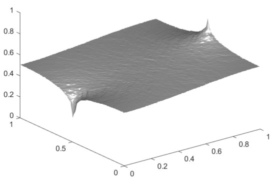
$$\textit{minimize } \frac{1}{2}\int_\Omega \gamma|\nabla u|^2\rho^2\,dx \textit{ over } \left\{u \in H^1_\gamma(\Omega) \text{ and } u = g \text{ on } \Gamma\right\}.$$

- Proof uses Gamma-convergence and TL^2 spaces developed by Slepčev and Trillos
- Can use PDE-arguments to upgrade to uniform convergence provided

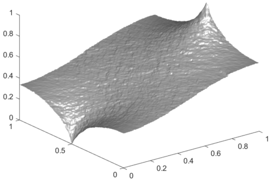
(32)
$$\left(\frac{\log(n)}{n}\right)^{1/(d+4)} \ll \varepsilon_n \ll \left(\frac{1}{n}\right)^{1/(\alpha+2)}.$$

Calder, J. and Slepčev, D. (2019). **Properly-weighted graph Laplacian for semi-supervised learning**. *Applied Mathematics and Optimization: Special Issue on Optimization in Data Science*, 82:1111–1159.

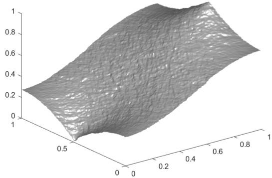
Solution profiles



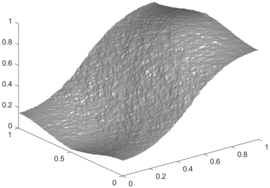
(a) $\alpha = 0$



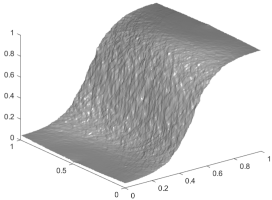
(b) $\alpha = 0.5, \zeta = 50n\varepsilon^2$



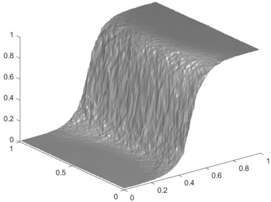
(c) $\alpha = 1, \zeta = 50n\varepsilon^2$



(d) $\alpha = 2, \zeta = 50n\varepsilon^2$

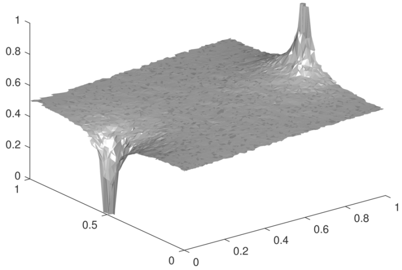


(e) $\alpha = 5, \zeta = 10^3 n\varepsilon^2$

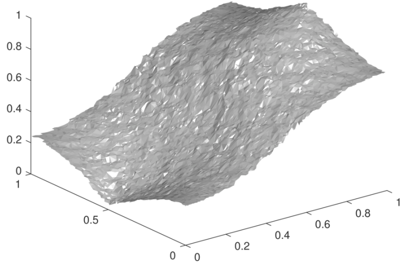


(f) $\alpha = 10, \zeta = 10^5 n\varepsilon^2$

Comparison with WNLL

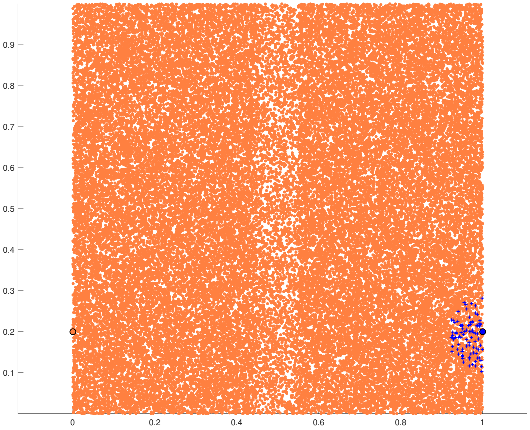


(g) WNLL [Shi et al., 2017]



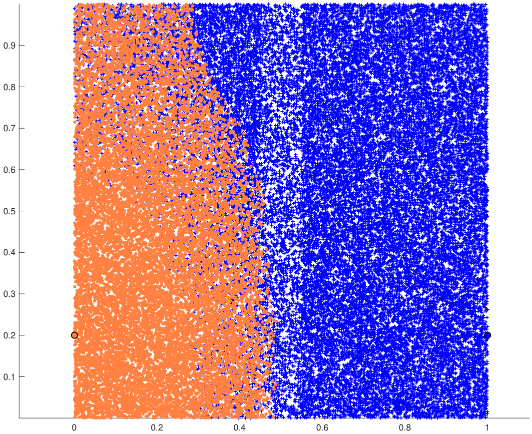
(h) PW Laplacian ($\alpha = 2$)

Classification: $p = 2$ Laplacian, $n = 50,000$ points



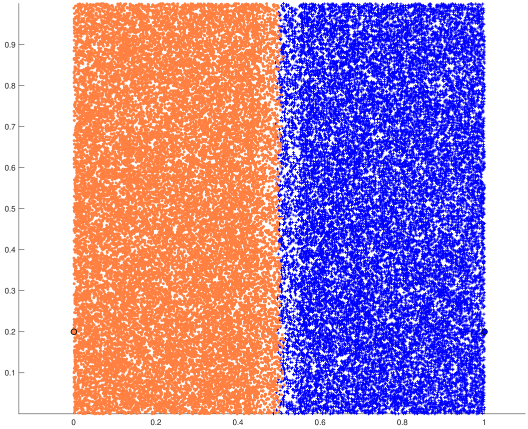
49.8% error

Classification: WNLL [Shi et al., 2017], $n = 50,000$ points



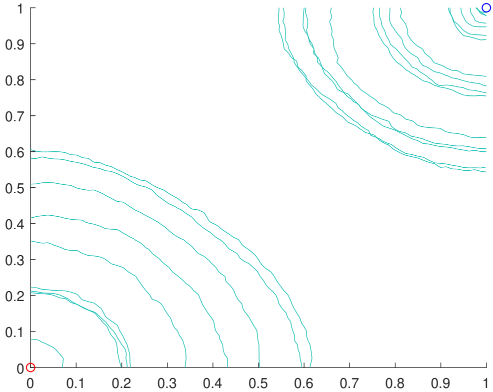
11% error

Classification: PW-Laplacian, $n = 50,000$ points

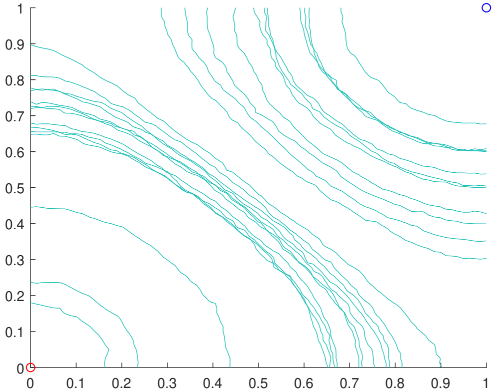


0.25% error

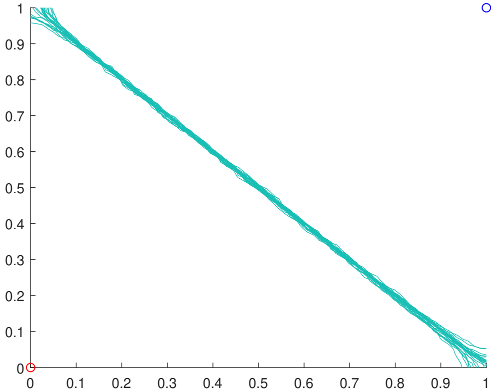
Classification: $p = 2$ Laplacian, $n = 10^5$ points



Classification: WNLL [Shi et al., 2017], $n = 10^5$ points



Classification: PW-Laplacian, $n = 10^5$ points



Outline

Poisson learning

We propose to replace Laplace learning

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

with Poisson learning

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

subject to $\sum_{x \in \mathcal{X}} d(x)u(x) = 0$, where $\overline{g} = \frac{1}{|\Gamma|} \sum_{y \in \Gamma} g(y)$.

In both cases, the label decision is the same:

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

Poisson learning

We propose to replace Laplace learning

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

with Poisson learning

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

subject to $\sum_{x \in \mathcal{X}} d(x)u(x) = 0$, where $\overline{g} = \frac{1}{|\Gamma|}\sum_{y \in \Gamma} g(y)$.

For Poisson learning, unbalanced class sizes can be incorporated:

$$\ell(x) = \operatorname*{argmax}_{j \in \{1,\ldots,k\}} \left\{ \frac{p_j}{n_j} u_j(x) \right\},$$

p_j = Fraction of data in class j

n_j = Fraction of training data from class j.

The random walk interpretation

Let X_0^x, X_1^x, X_2^x be a random walk on $\mathcal{X}$ starting from $x \in \mathcal{X}$, and define

$$u_T(x) := \mathbb{E}\left[\sum_{k=0}^{T} \frac{1}{d(x)} \sum_{y\in\Gamma} (g(y) - \overline{g}) \mathbb{1}_{\{X_k^y = x\}}\right], \quad \text{where } \overline{g} = \frac{1}{|\Gamma|}\sum_{y\in\Gamma} g(y).$$

Theorem (C.-Cook-Thorpe-Slepcev, 2020)

For every $T \geq 0$ we have

$$u_{T+1}(x) = u_T(x) + \frac{1}{d(x)}\left(\sum_{y\in\Gamma}(g(y) - \overline{g})\delta_{xy} - \mathcal{L}u_T(x)\right).$$

If the graph G is connected and the Markov chain induced by the random walk is aperiodic, then $u_T \to u$ as $T \to \infty$, where $u : \mathcal{X} \to \mathbb{R}$ is the solution of

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

satisfying $\sum_{x\in\mathcal{X}} d(x)u(x) = 0$.

The variational interpretation

Consider the variational problem

$$(33) \qquad \min_{u\in\ell_0^2(\mathcal{X})} \left\{ \sum_{x,y\in\mathcal{X}} w_{xy}|u(x)-u(y)|^2 - \sum_{x\in\Gamma}(g(x)-\overline{g})\cdot u(x)\right\},$$

where $\overline{g} = \frac{1}{|\Gamma|}\sum_{x\in\Gamma} g(x)$.

Theorem (C.-Cook-Thorpe-Slepcev, 2020)

Assume G is connected. Then there exists a unique minimizer $u\in\ell_0^2(\mathcal{X})$ of (33)*, and furthermore, u satisfies the Poisson equation*

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

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.

The continuum perspective

Manifold assumption: Let $x_1, \dots, x_n$ be a sequence of i.i.d. random variables with density ρ supported on a d-dimensional compact, closed, and connected Riemannian manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, where $d \ll D$. Fix a finite set of points $\Gamma \subset \mathcal{M}$ and set

$$\mathcal{X}_n := \underbrace{\{x_1, \dots, x_n\}}_{\text{Unlabeled}} \cup \underbrace{\Gamma}_{\text{Labeled}} .$$

Conjecture

Let $n \to \infty$ and $\varepsilon = \varepsilon_n \to 0$ so that $\lim_{n\to\infty} \frac{n\varepsilon_n^{d+2}}{\log n} = \infty$. Let u_n be the solution of the Poisson learning problem

$$\left(\frac{2}{\sigma_\eta n \varepsilon_n^{d+2}}\right) \mathcal{L} u_n(x) = \sum_{y\in\Gamma} (g(y) - \overline{g})(n\delta_{xy}) \quad \text{for } x \in \mathcal{X}_n.$$

Then with probability one $u_n \to u$ locally uniformly on $\mathcal{M} \setminus \Gamma$ as $n \to \infty$, where $u \in C^\infty(\mathcal{M} \setminus \Gamma)$ is the solution of the Poisson equation

$$-\,\mathrm{div}_{\mathcal{M}}\left(\rho^2 \nabla_{\mathcal{M}} u\right) = \sum_{y\in\Gamma} (g(y) - \overline{g})\delta_y \quad \text{on } \mathcal{M}.$$

Spectral representation

Theorem

The solution of the Poisson learning equation

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

is given by

$$u(x) = \sum_{y\in\Gamma}\sum_{k=2}^{n}(g(y)-\overline{g})\lambda_k^{-1}v_k(x)v_k(y),$$

where $v_1, v_2, \ldots, v_n$ *are the normalized eigenvectors of* $\mathcal{L}$*, with corresponding eigenvalues* $0=\lambda_1<\lambda_2\leq\cdots\leq\lambda_n$.

Proof of the conjecture reduces to spectral convergence. We proved $O(\varepsilon)$ spectral convergence rates in the $C^{0,1}$ sense:

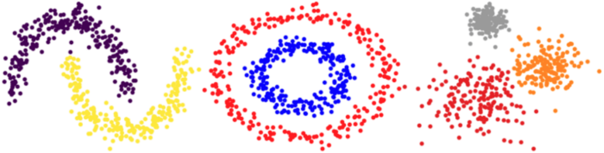
J. Calder, N. Garcia Trillos, and M. Lewicka, **Lipschitz regularity of graph Laplacians on random data clouds**, *arXiv:2007.06679*, 2020.

Outline

Outline

GraphLearning Python Package

Graph-based Clustering and Semi-Supervised Learning



This python package is devoted to efficient implementations of modern graph-based learning algorithms for both semi-supervised learning and clustering. The package implements many popular datasets (currently MNIST, FashionMNIST, cifar-10, and WEBKB) in a way that makes it simple for users to test out new algorithms and rapidly compare against existing methods.

This package reproduces experiments from the paper

Calder, Cook, Thorpe, Slepcev. Poisson Learning: Graph Based Semi-Supervised Learning at Very Low Label Rates., Proceedings of the 37th International Conference on Machine Learning, PMLR 119:1306-1316, 2020.

Installation

Install with

```
pip install graphlearning
```

https://github.com/jwcalder/GraphLearning

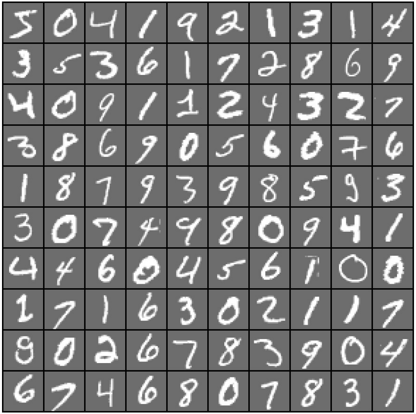
Algorithmic details

Algorithm 1 Poisson Learning

Input: $\mathbf{W}, \mathbf{F} = [y_1, y_2, \ldots, y_m], T$
$\mathbf{D} \leftarrow \text{diag}(\mathbf{W}\mathbb{1})$
$\mathbf{L} \leftarrow \mathbf{D} - \mathbf{W}$
$\mathbf{c} \leftarrow \frac{1}{m}\mathbf{F}\mathbb{1}$
$\mathbf{B} \leftarrow [\mathbf{F} - \mathbf{c}, \textbf{zeros}(k, n-m)]$
$\mathbf{U} \leftarrow \textbf{zeros}(n, k)$
for $i = 1$ **to** T **do**
    $\mathbf{U} \leftarrow \mathbf{U} + \mathbf{D}^{-1}(\mathbf{B}^T - \mathbf{L}\mathbf{U})$
end for
$\ell_i \leftarrow \operatorname{argmax}_{1 \leq j \leq k} \mathbf{U}_{ij}$
return: $\boldsymbol{\ell} := [\ell_1, \ell_2, \ldots, \ell_n]$

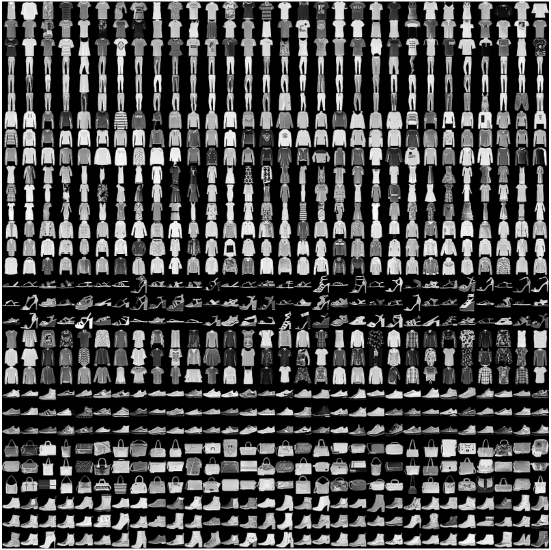
We only need about $T = 100$ iterations on MNIST, FashionMNIST, CIFAR-10, to get good results. CPU Time: 4 seconds on CPU, 1 second on GPU.

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



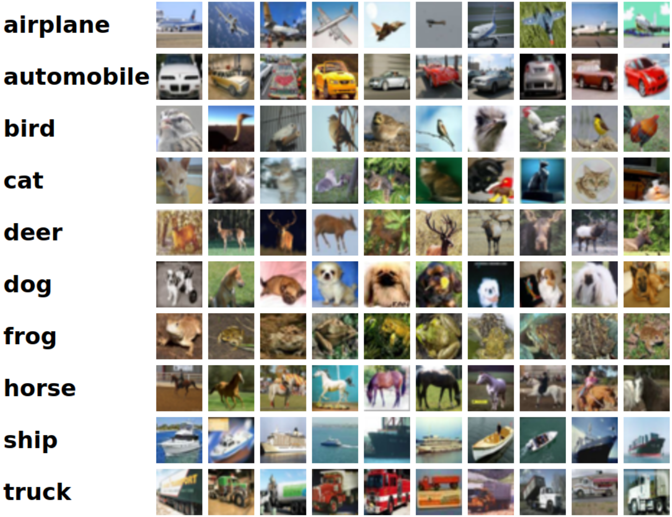
[Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition.” Proceedings of the IEEE, 86(11):2278-2324, November 1998.]

FashionMNIST (70,000 28×28 images of fashion items)



[Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." arXiv:1708.07747 (2017).]

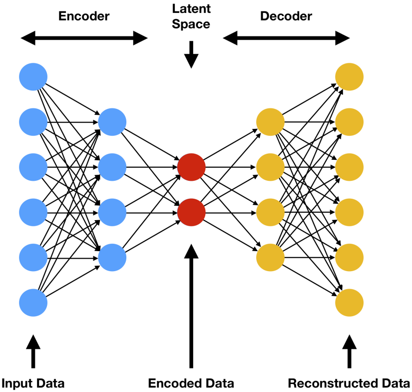
CIFAR-10



[Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).]

Autoencoders

For each dataset, we build the graph by training autoencoders.



www.compthree.com

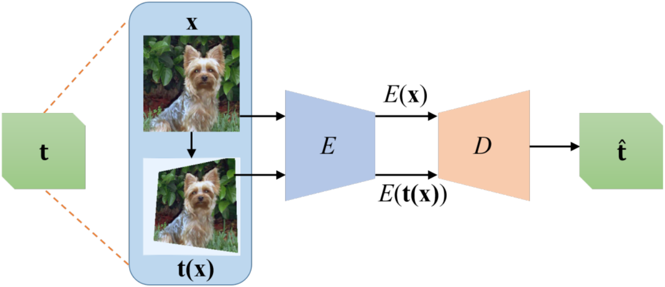
Autoencoders are "Nonlinear versions of PCA"

Building graphs from autoencoders

For MNIST and FashionMNIST, we use a 4-layer variational autoencoder with 30 latent variables:

[Kingma and Welling. Auto-encoding variational Bayes. ICML 2014]

For CIFAR-10, we use the autoencoding framework from [Zhang et al. AutoEncoding Transformations (AET), CVPR 2019] with 12,288 latent variables.



First comparison

We compared against many other graph-based learning algorithms

- Laplace/Label propagation: [Zhu et al., 2003]
- Graph nearest neighbor (using Dijkstra)
- Lazy random walks: [Zhou et al., 2004]
- Mutli-class MBO: [Garcia-Cardona et al., 2014]
- Centered kernel method: [Mai & Couillet, 2018]
- Sparse Label Propagation: [Jung et al., 2016]
- Weighted Nonlocal Laplacian (WNLL): [Shi et al., 2017]
- p-Laplace regularization: [Flores et al. 2019]

MNIST results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	16.1 (6.2)	28.2 (10.3)	42.0 (12.4)	57.8 (12.3)	69.5 (12.2)
Nearest Neighbor	65.4 (5.2)	74.2 (3.3)	77.8 (2.6)	80.7 (2.0)	82.1 (2.0)
Random Walk	66.4 (5.3)	76.2 (3.3)	80.0 (2.7)	82.8 (2.3)	84.5 (2.0)
MBO	19.4 (6.2)	29.3 (6.9)	40.2 (7.4)	50.7 (6.0)	59.2 (6.0)
Centered Kernel	19.1 (1.9)	24.2 (2.3)	28.8 (3.4)	32.6 (4.1)	35.6 (4.6)
Sparse Label Prop.	14.0 (5.5)	14.0 (4.0)	14.5 (4.0)	18.0 (5.9)	16.2 (4.2)
WNLL	55.8 (15.2)	82.8 (7.6)	90.5 (3.3)	93.6 (1.5)	94.6 (1.1)
p-Laplace	72.3 (9.1)	86.5 (3.9)	89.7 (1.6)	90.3 (1.6)	91.9 (1.0)
Poisson	**90.2 (4.0)**	**93.6 (1.6)**	**94.5 (1.1)**	**94.9 (0.8)**	**95.3 (0.7)**

FashionMNIST results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	18.4 (7.3)	32.5 (8.2)	44.0 (8.6)	52.2 (6.2)	57.9 (6.7)
Nearest Neighbor	46.6 (4.7)	53.5 (3.6)	57.2 (3.0)	59.3 (2.6)	61.1 (2.8)
Random Walk	49.0 (4.4)	55.6 (3.8)	59.4 (3.0)	61.6 (2.5)	63.4 (2.5)
MBO	15.7 (4.1)	20.1 (4.6)	25.7 (4.9)	30.7 (4.9)	34.8 (4.3)
Centered Kernel	11.8 (0.4)	13.1 (0.7)	14.3 (0.8)	15.2 (0.9)	16.3 (1.1)
Sparse Label Prop.	14.1 (3.8)	16.5 (2.0)	13.7 (3.3)	13.8 (3.3)	16.1 (2.5)
WNLL	44.6 (7.1)	59.1 (4.7)	64.7 (3.5)	67.4 (3.3)	70.0 (2.8)
p-Laplace	54.6 (4.0)	57.4 (3.8)	65.4 (2.8)	68.0 (2.9)	68.4 (0.5)
Poisson	**60.8 (4.6)**	**66.1 (3.9)**	**69.6 (2.6)**	**71.2 (2.2)**	**72.4 (2.3)**

Compare to clustering result of 67.2% [McConville et al., 2019]

CIFAR-10 results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	10.4 (1.3)	11.0 (2.1)	11.6 (2.7)	12.9 (3.9)	14.1 (5.0)
Nearest Neighbor	33.1 (4.3)	37.3 (4.1)	39.7 (3.0)	41.7 (2.8)	43.0 (2.5)
Random Walk	36.4 (4.9)	42.0 (4.4)	45.1 (3.3)	47.5 (2.9)	49.0 (2.6)
MBO	14.2 (4.1)	19.3 (5.2)	24.3 (5.6)	28.5 (5.6)	33.5 (5.7)
Centered Kernel	15.4 (1.6)	16.9 (2.0)	18.8 (2.1)	19.9 (2.0)	21.7 (2.2)
Sparse Label Prop.	11.8 (2.4)	12.3 (2.4)	11.1 (3.3)	14.4 (3.5)	11.0 (2.9)
WNLL	16.6 (5.2)	26.2 (6.8)	33.2 (7.0)	39.0 (6.2)	44.0 (5.5)
p-Laplace	26.0 (6.7)	35.0 (5.4)	42.1 (3.1)	48.1 (2.6)	49.7 (3.8)
Poisson	**40.7 (5.5)**	**46.5 (5.1)**	**49.9 (3.4)**	**52.3 (3.1)**	**53.8 (2.6)**

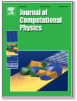
Compare to clustering result of 41.2% [Mukherjee et al., ClusterGAN, CVPR 2019].

Outline

Volume constrained semi-supervised learning

ELSEVIER

Journal of Computational Physics
Volume 354, 1 February 2018, Pages 288-310



Auction dynamics: A volume constrained MBO scheme

Matt Jacobs, Ekaterina Merkurjev, Selim Esedoḡlu

Show more

https://doi.org/10.1016/j.jcp.2017.10.036

Get rights and content

Classification results can be improved by incorporating prior knowledge of class sizes through volume constraints.

PoissonMBO: Volume constrained Poisson learning

Observation 1: The Poisson learning iteration with a fixed time step

$$u_{T+1}(x) = u_T(x) + dt\left(\sum_{y\in\Gamma}(g(y) - \overline{g})\delta_{ij} - \mathcal{L}u_T(x)\right)$$

is **volume preserving**. That is $\sum_{x\in\mathcal{X}} u_{T+1}(x) = \sum_{x\in\mathcal{X}} u_T(x)$.

Observation 2: We can easily perform a volume constrained label projection

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

We adjust the weights s_j to grow/shrink each region to achieve the correct class sizes.

Named after the Merriman-Bence-Osher (MBO) scheme for curvature motion, which has been used before in graph-based learning [Garcia, et al., 2014, Jacobs et al., 2018].

MNIST results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	16.1 (6.2)	28.2 (10.3)	42.0 (12.4)	57.8 (12.3)	69.5 (12.2)
WNLL	55.8 (15.2)	82.8 (7.6)	90.5 (3.3)	93.6 (1.5)	94.6 (1.1)
p-Laplace	72.3 (9.1)	86.5 (3.9)	89.7 (1.6)	90.3 (1.6)	91.9 (1.0)
VolumeMBO	89.9 (7.3)	95.6 (1.9)	96.2 (1.2)	96.6 (0.6)	96.7 (0.6)
Poisson	90.2 (4.0)	93.6 (1.6)	94.5 (1.1)	94.9 (0.8)	95.3 (0.7)
PoissonMBO	**96.5 (2.6)**	**97.2 (0.1)**	**97.2 (0.1)**	**97.2 (0.1)**	**97.2 (0.1)**
# Labels per class	**10**	**20**	**40**	**80**	**160**
Laplace/LP	91.3 (3.7)	95.8 (0.6)	96.5 (0.2)	96.8 (0.1)	97.0 (0.1)
WNLL	95.6 (0.5)	96.1 (0.3)	96.3 (0.2)	96.4 (0.1)	96.3 (0.1)
p-Laplace	94.0 (0.8)	95.1 (0.4)	95.5 (0.1)	96.0 (0.2)	96.2 (0.1)
VolumeMBO	96.9 (0.2)	97.0 (0.1)	97.1 (0.1)	97.2 (0.1)	**97.3 (0.1)**
Poisson	95.9 (0.4)	96.3 (0.3)	96.6 (0.2)	96.8 (0.1)	96.9 (0.1)
PoissonMBO	**97.2 (0.1)**	**97.2 (0.1)**	**97.2 (0.1)**	**97.2 (0.1)**	97.2 (0.1)

FashionMNIST results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	18.4 (7.3)	32.5 (8.2)	44.0 (8.6)	52.2 (6.2)	57.9 (6.7)
WNLL	44.6 (7.1)	59.1 (4.7)	64.7 (3.5)	67.4 (3.3)	70.0 (2.8)
p-Laplace	54.6 (4.0)	57.4 (3.8)	65.4 (2.8)	68.0 (2.9)	68.4 (0.5)
VolumeMBO	54.7 (5.2)	61.7 (4.4)	66.1 (3.3)	68.5 (2.8)	70.1 (2.8)
Poisson	60.8 (4.6)	66.1 (3.9)	69.6 (2.6)	71.2 (2.2)	72.4 (2.3)
PoissonMBO	**62.0 (5.7)**	**67.2 (4.8)**	**70.4 (2.9)**	**72.1 (2.5)**	**73.1 (2.7)**
# Labels per class	**10**	**20**	**40**	**80**	**160**
Laplace/LP	70.6 (3.1)	76.5 (1.4)	79.2 (0.7)	80.9 (0.5)	**82.3 (0.3)**
WNLL	74.4 (1.6)	77.6 (1.1)	79.4 (0.6)	80.6 (0.4)	81.5 (0.3)
p-Laplace	73.0 (0.9)	76.2 (0.8)	78.0 (0.3)	79.7 (0.5)	80.9 (0.3)
VolumeMBO	74.4 (1.5)	77.4 (1.0)	79.5 (0.7)	**81.0 (0.5)**	82.1 (0.3)
Poisson	75.2 (1.5)	77.3 (1.1)	78.8 (0.7)	79.9 (0.6)	80.7 (0.5)
PoissonMBO	**76.1 (1.4)**	**78.2 (1.1)**	**79.5 (0.7)**	80.7 (0.6)	81.6 (0.5)

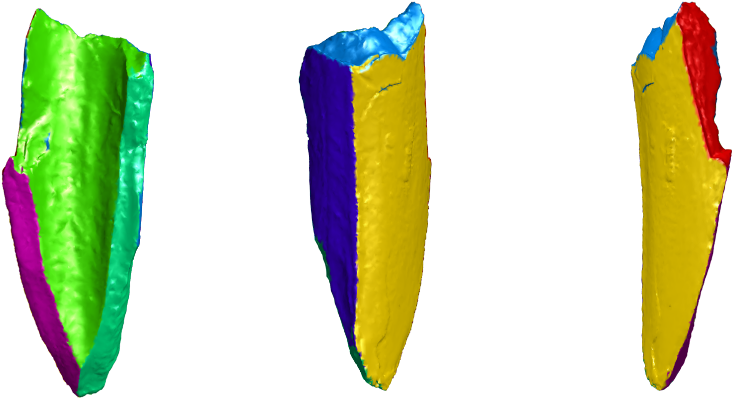
CIFAR-10 results

Table: Average (standard deviation) classification accuracy over 100 trials.

# Labels per class	**1**	**2**	**3**	**4**	**5**
Laplace/LP	10.4 (1.3)	11.0 (2.1)	11.6 (2.7)	12.9 (3.9)	14.1 (5.0)
WNLL	16.6 (5.2)	26.2 (6.8)	33.2 (7.0)	39.0 (6.2)	44.0 (5.5)
p-Laplace	26.0 (6.7)	35.0 (5.4)	42.1 (3.1)	48.1 (2.6)	49.7 (3.8)
VolumeMBO	38.0 (7.2)	46.4 (7.2)	50.1 (5.7)	53.3 (4.4)	55.3 (3.8)
Poisson	40.7 (5.5)	46.5 (5.1)	49.9 (3.4)	52.3 (3.1)	53.8 (2.6)
PoissonMBO	**41.8 (6.5)**	**50.2 (6.0)**	**53.5 (4.4)**	**56.5 (3.5)**	**57.9 (3.2)**
# Labels per class	**10**	**20**	**40**	**80**	**160**
Laplace/LP	21.8 (7.4)	38.6 (8.2)	54.8 (4.4)	62.7 (1.4)	66.6 (0.7)
WNLL	54.0 (2.8)	60.3 (1.6)	64.2 (0.7)	66.6 (0.6)	68.2 (0.4)
p-Laplace	56.4 (1.8)	60.4 (1.2)	63.8 (0.6)	66.3 (0.6)	68.7 (0.3)
VolumeMBO	59.2 (3.2)	61.8 (2.0)	63.6 (1.4)	64.5 (1.3)	65.8 (0.9)
Poisson	58.3 (1.7)	61.5 (1.3)	63.8 (0.8)	65.6 (0.6)	67.3 (0.4)
PoissonMBO	**61.8 (2.2)**	**64.5 (1.6)**	**66.9 (0.8)**	**68.7 (0.6)**	**70.3 (0.4)**

Outline

Application: Segmenting broken bone fragments

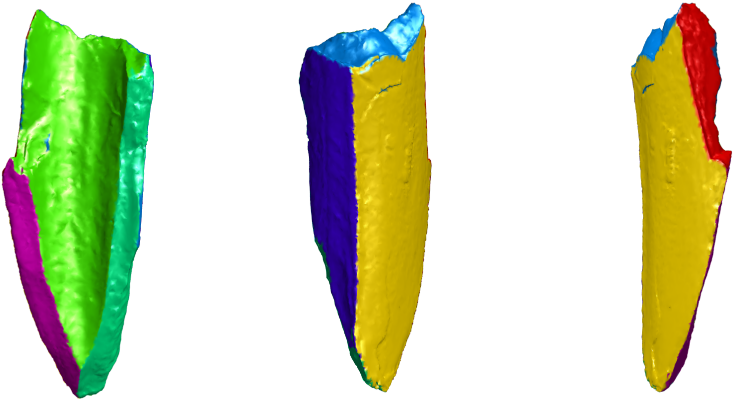


AMAAZE consortium for mathematics and anthropology: `https://amaaze.umn.edu/`

Main collaborators: Peter J. Olver and Katrina Yezzi-Woodley (Anthropology)

REU students: Math: David Floeder, Anthropology: Paige Cody, Chloe Siewert
Math Graduate students: Riley O'Neill, Brendan Cook

Application: Segmenting broken bone fragments

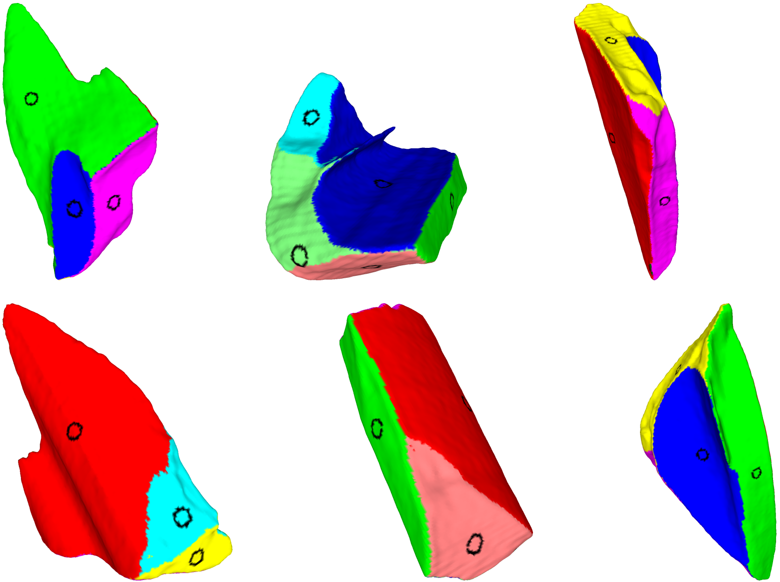


Graph-based clustering with weights

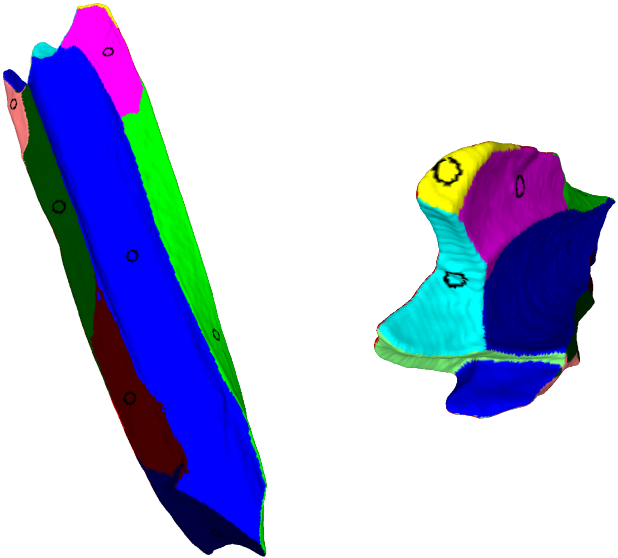
$$w_{ij} = \exp\left(-C|\mathbf{n}_i - \mathbf{n}_j|^p\right).$$

between nearby points on the mesh, where $\mathbf{n}_i$ is the outward normal vector at vertex i.

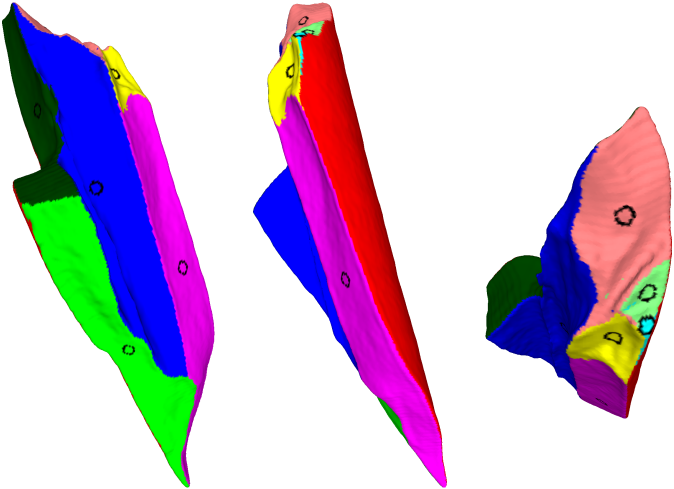
Mesh Segmentation via Poisson Learning



Mesh Segmentation via Poisson Learning

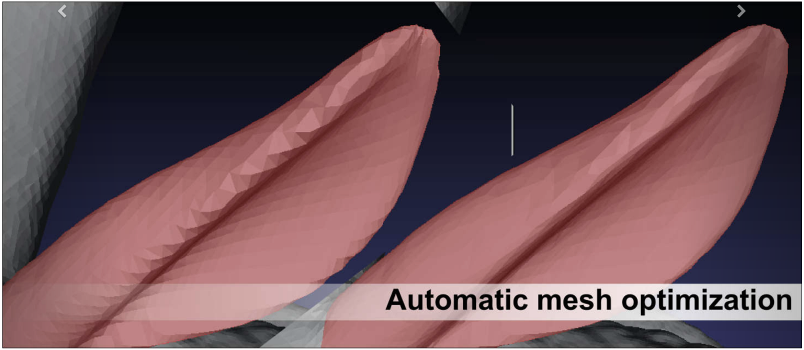


Mesh Segmentation via Poisson Learning



AMAAZE MeshLab plugins

MeshLab the open source system for processing and editing 3D triangular meshes. It provides a set of tools for editing, cleaning, healing, inspecting, rendering, texturing and converting meshes. It offers features for processing raw data produced by 3D digitization tools/devices and for preparing models for 3D printing.



`https://amaaze.umn.edu/software`

Outline

Current/Future Work

1. Poisson learning
   - Directed graphs, clustering
   - Continuum limit
   - Asymptotic consistency
2. Rates of convergence for p-Laplacian regularization
   - Including other graphs, like stochastic block models
3. Graph convolutional networks for semi-supervised learning
   - [Kipf & Welling, ICLR 2017]
4. Few-shot semi-supervised learning
   - H. Huang, J. Zhang, J. Zhang, Q. Wu, C. Xu. **PTN: A Poisson Transfer Network for Semi-supervised Few-shot Learning.** To appear in proceedings of AAAI 2021 (arXiv preprint:2012.10844).

References

References:

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

2. Calder, J. (2019). **Consistency of Lipschitz learning with infinite unlabeled data and finite labeled data.** *SIAM Journal on Mathematics of Data Science* 1(4):780–812.

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

4. J. Calder, D. Slepčev, D., and M. Thorpe. **Rates of convergence for Laplacian semi-supervised learning with low label rates.** *arXiv:2006.02765*, 2020.

5. 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.

Code: `https://github.com/jwcalder/GraphLearning` (pip install graphlearning)

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.