New directions in graph-based learning: Active learning, Hamilton-Jacobi equations on graphs, and elliptic regularity

Jeff Calder

School of Mathematics
University of Minnesota

IPAM Hamilton-Jacobi Opening Day
March 9, 2020

Outline

Outline

1. Introduction and background

2. New directions
   - Hamilton-Jacobi equations on graphs
   - Active Learning
   - Elliptic regularity

Quick intro to learning

Fully supervised: Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, learn a function

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

Quick intro to learning

Fully supervised: Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, 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: Given additional unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$, use both the labeled and unlabeled data to learn f.

Quick intro to learning

Fully supervised: Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, 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: Given additional unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$, use both the labeled and unlabeled data to learn f.

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

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

Quick intro to learning

Fully supervised: Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, 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: Given additional unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$, use both the labeled and unlabeled data to learn f.

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

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

Quick intro to learning

Fully supervised: Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$, 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: Given additional unlabeled data $x_{n+1}, \ldots, x_{n+m}$ for $m \geq 1$, use both the labeled and unlabeled data to learn f.

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

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

Unsupervised learning: Algorithms that use only the unlabeled data $x_1, \ldots, x_n$, such as clustering.

Example: Automated image captioning

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.]

Example: Automated image captioning fail

[Andrej Karpathy's NeuralTalk]

Graph-based learning

In semi-supervised and unsupervised learning, we often build 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.

Graph-based learning

In semi-supervised and unsupervised learning, we often build 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.

The graph Laplacian:

$$\mathcal{L}u(x) = \sum_{y\in\mathcal{X}} w_{xy}(u(y) - u(x)) = 0. \quad (u : \mathcal{X} \to \mathbb{R}^k)$$

- Laplacian regularized semi-supervised learning [Zhu et al., (2003)]
  - Propagate labels on a graph by harmonic extension.

Graph-based learning

In semi-supervised and unsupervised learning, we often build 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.

The graph Laplacian:

$$\mathcal{L}u(x) = \sum_{y\in\mathcal{X}} w_{xy}(u(y) - u(x)) = 0. \quad (u : \mathcal{X} \to \mathbb{R}^k)$$

- Laplacian regularized semi-supervised learning [Zhu et al., (2003)]
  - Propagate labels on a graph by harmonic extension.
- Spectral clustering [Shi and Malik (2000)] [Ng, Jordan, and Weiss (2002)]
  - Embed a graph into $\mathbb{R}^k$ by projecting onto eigenspaces of $\mathcal{L}$.

Graph-based learning

In semi-supervised and unsupervised learning, we often build a graph $(\mathcal{X}, \mathcal{W})$:

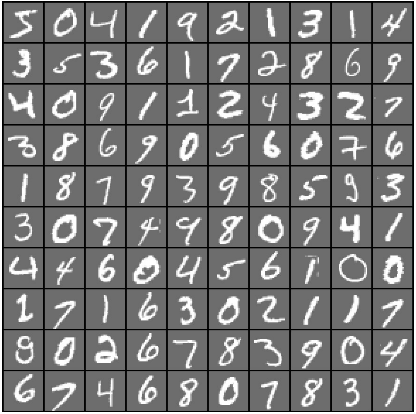
- $\mathcal{X} \subset \mathbb{R}^d$ are the vertices and
- $\mathcal{W} = (w_{xy})_{x,y\in\mathcal{X}}$ are the nonnegative edge weights.
- $w_{xy} \approx 1$ if x, y similar, and $w_{xy} \approx 0$ when dissimilar.

The graph Laplacian:

$$\mathcal{L}u(x) = \sum_{y\in\mathcal{X}} w_{xy}(u(y) - u(x)) = 0. \qquad (u : \mathcal{X} \to \mathbb{R}^k)$$

- Laplacian regularized semi-supervised learning [Zhu et al., (2003)]
  - Propagate labels on a graph by harmonic extension.
- Spectral clustering [Shi and Malik (2000)] [Ng, Jordan, and Weiss (2002)]
  - Embed a graph into $\mathbb{R}^k$ by projecting onto eigenspaces of $\mathcal{L}$.
- Laplacian eigenmaps [Belkin and Niyogi (2003)], Diffusion maps [Coifman and Lafon (2006)]

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.]

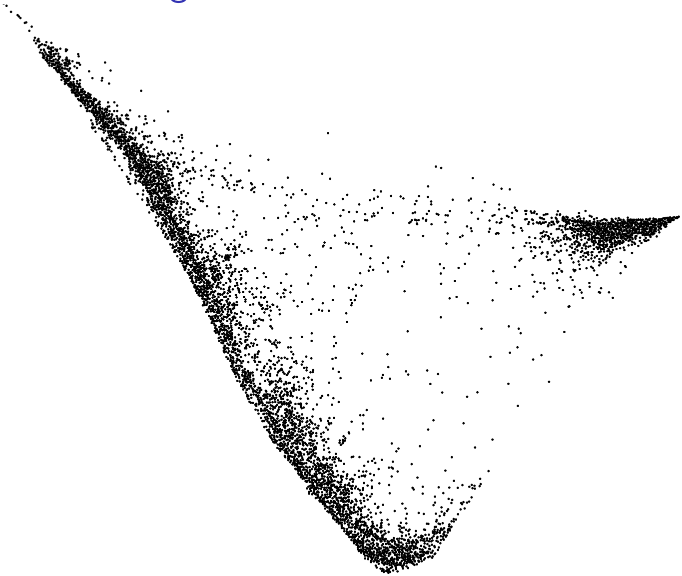
Laplace learning on MNIST

# Labels/class	**10**	**50**	**100**	**500**	**1000**
Laplace	93.2 (2.3)	96.9 (0.1)	97.1 (0.1)	97.6 (0.1)	97.7 (0.0)

Average accuracy over 10 trials with standard deviation in brackets.

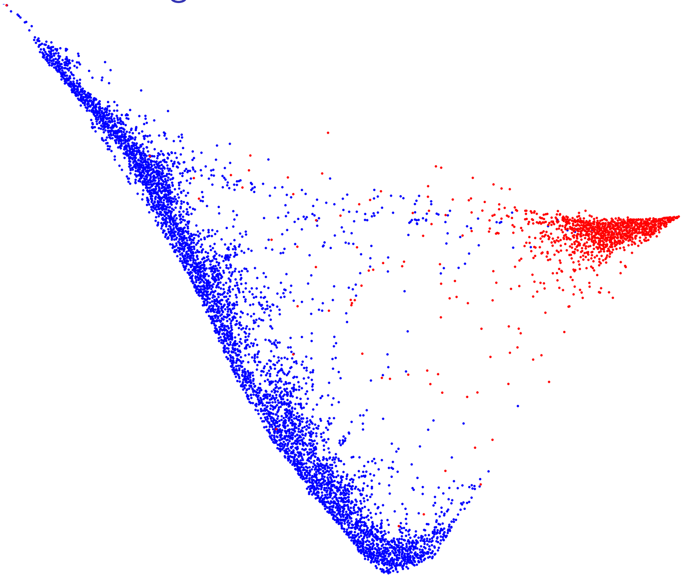
Weight matrix constructed with the Scattering Transform [Bruna and Mallat, 2013].

Spectral embedding: MNIST



Digits 1 and 2 from MNIST visualized with spectral projection

Spectral embedding: MNIST



Digits 1 (blue) and 2 (red) from MNIST visualized with spectral projection

Random geometric graph (ε-ball 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

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

where $\eta : [0,\infty) \to [0,1]$ is smooth with compact support.

Random geometric graph (ε-ball graph)

Assume the vertices of the graph are

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

where $x_1, \dots, x_n$ 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

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

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

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

Manifold assumption: Also common to assume $x_1, \dots, x_n$ are supported on a smooth manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$.

k-nearest neighbor graph

Let

$$\varepsilon_k(x) = \text{Distance from } x \text{ to } k^{\text{th}} \text{ nearest neighbor.}$$

- Non-symmetric (or directed) k-nn graph

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

k-nearest neighbor graph

Let

$$\varepsilon_k(x) = \text{Distance from } x \text{ to } k^{\text{th}} \text{ nearest neighbor.}$$

- Non-symmetric (or directed) k-nn graph

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

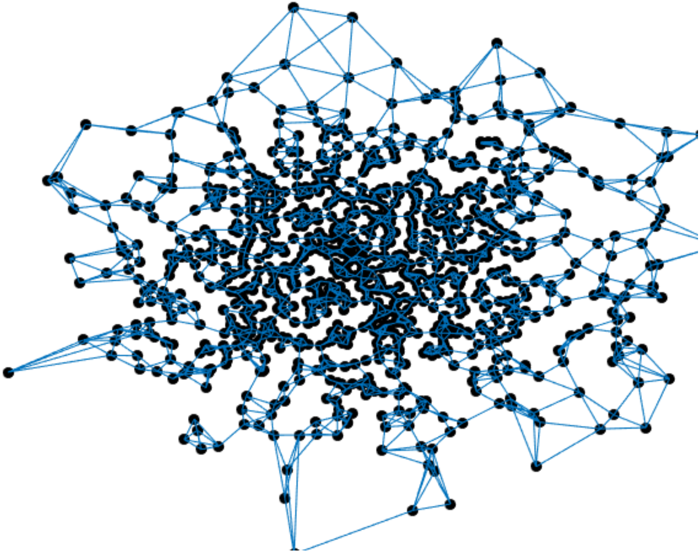
- Various ways to symmetrize:

$$w_{xy} = \eta\left(\frac{|x-y|}{\varepsilon_k(x)}\right) + \eta\left(\frac{|x-y|}{\varepsilon_k(y)}\right)$$
$$w_{xy} = \eta\left(\frac{|x-y|}{\min\{\varepsilon_k(x), \varepsilon_k(x)\}}\right)$$
$$w_{xy} = \eta\left(\frac{|x-y|}{\max\{\varepsilon_k(x), \varepsilon_k(x)\}}\right)$$

Synthetic Gaussian Data

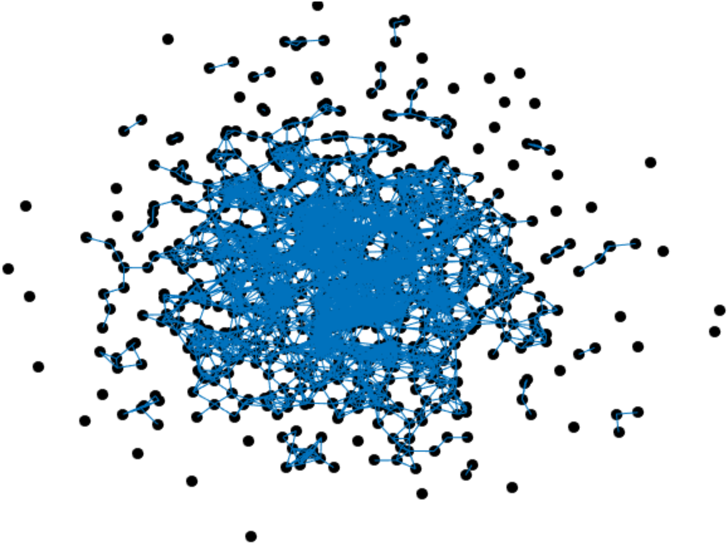
Synthetic Gaussian Data

k-nn graph



$k = 5$, Sparsity $\leq 1\%$

Random geometric graph



$\varepsilon = 0.25$, Sparsity $\sim 1.7\%$, Disconnected graph

Continuum limits in graph-based learning

The limit is taken jointly as $n \to \infty$ and $\varepsilon \to 0$.

- Early work [Hein et al., 2007] established pointwise consistency for smooth functions, with high probability

$$\mathcal{L}u = \rho^{-1}\text{div}\left(\rho^2 \nabla u\right) + O(\varepsilon).$$

Continuum limits in graph-based learning

The limit is taken jointly as $n \to \infty$ and $\varepsilon \to 0$.

- Early work [Hein et al., 2007] established pointwise consistency for smooth functions, with high probability

$$\mathcal{L}u = \rho^{-1}\text{div}\left(\rho^2 \nabla u\right) + O(\varepsilon).$$

- Also early spectral convergence results [Von Lusburg et al, 2008] showed that the eigenvalues and eigenvectos of $\mathcal{L}$ converge to those of a weighted Laplace-Beltrami operator on $\mathcal{M}$, though without convergence rates.

Continuum limits in graph-based learning

The limit is taken jointly as $n \to \infty$ and $\varepsilon \to 0$.

- Early work [Hein et al., 2007] established pointwise consistency for smooth functions, with high probability

$$\mathcal{L}u = \rho^{-1}\mathrm{div}\left(\rho^2 \nabla u\right) + O(\varepsilon).$$

- Also early spectral convergence results [Von Lusburg et al, 2008] showed that the eigenvalues and eigenvectos of $\mathcal{L}$ converge to those of a weighted Laplace-Beltrami operator on $\mathcal{M}$, though without convergence rates.

- Γ-convergence framework developed in [Trillos & Slepčev 2016] for variational convergence.
  - Continuum limit for total variation on graphs [Trillos & Slepčev 2016].
  - Spectral convergence rates [Trillos et al., 2018], [Calder & Trillos 2019].
  - Many other applications.

Continuum limits in graph-based learning

The limit is taken jointly as $n \to \infty$ and $\varepsilon \to 0$.

- Early work [Hein et al., 2007] established pointwise consistency for smooth functions, with high probability

$$\mathcal{L}u = \rho^{-1}\mathrm{div}\left(\rho^2 \nabla u\right) + O(\varepsilon).$$

- Also early spectral convergence results [Von Lusburg et al, 2008] showed that the eigenvalues and eigenvectos of $\mathcal{L}$ converge to those of a weighted Laplace-Beltrami operator on $\mathcal{M}$, though without convergence rates.

- Γ-convergence framework developed in [Trillos & Slepčev 2016] for variational convergence.
    - Continuum limit for total variation on graphs [Trillos & Slepčev 2016].
    - Spectral convergence rates [Trillos et al., 2018], [Calder & Trillos 2019].
    - Many other applications.

- Maximum principle and vicosity solution approach [Calder, 2018].

Convergence rates

Eigenmode	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Eigenvalue	2	2	2	6	6	6	6	6	12	12	12	12	12	12	12
E.value rate	2.4	2.6	3.1	2.3	2.3	2.5	2.6	3	2.1	2.1	2.2	2.3	2.4	2.8	3.3
E.vector rate	2.3	2.3	2.3	2.2	2.2	2.2	2.3	2.7	2.2	2.1	2.1	2.2	2.2	2.3	2.5

Table: Rates of convergence of the form $O(\varepsilon^b)$ (value of b is shown) for eigenvalues and eigenvectors of the graph Laplacian on the 2-sphere. Errors are averaged over 100 trials with n ranging from $n = 500$ to $n = 10^5$.

Rates of convergence for

$$\varepsilon = \left(\frac{\log n}{n}\right)^{1/(d+2)}.$$

Sharpest known convergence rates are $O(\varepsilon)$ [Calder & Trillos 2019].

Outline

Some new directions

Previous/current work has focused on continuum limits: $F_{n,\varepsilon} \to F$, which gives

- Well-posedness/stability results
- New understandings of algorithms

Some new directions

Previous/current work has focused on continuum limits: $F_{n,\varepsilon} \to F$, which gives

- Well-posedness/stability results
- New understandings of algorithms

New directions should use the developed tools to

- Establish performance guarantees for algorithms
  - Do we actually recover the correct labels?
- Develop new algorithms with performance guarantees

Some new directions

Previous/current work has focused on continuum limits: $F_{n,\varepsilon} \to F$, which gives

- Well-posedness/stability results
- New understandings of algorithms

New directions should use the developed tools to

- Establish performance guarantees for algorithms
  - Do we actually recover the correct labels?
- Develop new algorithms with performance guarantees

Some new directions

1. Hamilton-Jacobi equations on graphs
2. Active learning
3. Elliptic regularity for graph Laplacians

Outline

Hamilton-Jacobi equations on graphs

The graph eikonal equation gives a "nearest neighbor" classifier:

$$
\begin{cases} \min_{y \sim x} \{\nabla u(x,y) + w_{xy}\} = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = 0, & \text{if } x \in \Gamma. \end{cases} \tag{3}
$$

Here, $\nabla u(x,y) = u(y) - u(x)$ and $w_{xy} = |x - y|$.

Hamilton-Jacobi equations on graphs

The graph eikonal equation gives a "nearest neighbor" classifier:

$$\begin{cases} \min_{y \sim x} \{\nabla u(x,y) + w_{xy}\} = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = 0, & \text{if } x \in \Gamma. \end{cases} \tag{3}$$

Here, $\nabla u(x,y) = u(y) - u(x)$ and $w_{xy} = |x - y|$.

Performance on MNIST

# Labels/class	**10**	**50**	**100**	**500**	**1000**
Laplace (14 sec.)	93.2 (2.3)	96.9 (0.1)	97.1 (0.1)	97.6 (0.1)	97.7 (0.0)
eikonal (0.3 sec.)	82.3 (1.0)	89.0 (0.5)	90.6 (0.4)	93.4 (0.1)	93.7 (0.1)

Hamilton-Jacobi equations on graphs

The graph eikonal equation gives a "nearest neighbor" classifier:

$$\begin{cases} \min_{y \sim x} \{\nabla u(x,y) + w_{xy}\} = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = 0, & \text{if } x \in \Gamma. \end{cases} \tag{3}$$

Here, $\nabla u(x,y) = u(y) - u(x)$ and $w_{xy} = |x - y|$.

Performance on MNIST

# Labels/class	**10**	**50**	**100**	**500**	**1000**
Laplace (14 sec.)	93.2 (2.3)	96.9 (0.1)	97.1 (0.1)	97.6 (0.1)	97.7 (0.0)
eikonal (0.3 sec.)	82.3 (1.0)	89.0 (0.5)	90.6 (0.4)	93.4 (0.1)	93.7 (0.1)

Question: How can we incorporate information about the data distribution ρ into Hamilton-Jacobi equations on graphs for classification?

Hamilton-Jacobi equations on graphs

The graph eikonal equation gives a "nearest neighbor" classifier:

$$\begin{cases} \min\limits_{y\sim x} \{\nabla u(x,y) + w_{xy}\} = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = 0, & \text{if } x \in \Gamma. \end{cases} \tag{3}$$

Here, $\nabla u(x,y) = u(y) - u(x)$ and $w_{xy} = |x - y|$.

Performance on MNIST

# Labels/class	**10**	**50**	**100**	**500**	**1000**
Laplace (14 sec.)	93.2 (2.3)	96.9 (0.1)	97.1 (0.1)	97.6 (0.1)	97.7 (0.0)
eikonal (0.3 sec.)	82.3 (1.0)	89.0 (0.5)	90.6 (0.4)	93.4 (0.1)	93.7 (0.1)

Question: How can we incorporate information about the data distribution ρ into Hamilton-Jacobi equations on graphs for classification?

Question: Are more general HJ-equations $H(x, \nabla u) = 0$ useful?

Outline

Active learning

Main question: Which data points should be queried for labels?

Active learning

Main question: Which data points should be queried for labels?

Using PageRank to choose labeled points

# Labels/class	**10**	**50**	**100**	**500**	**1000**
Laplace	93.2 (2.3)	96.9 (0.1)	97.1 (0.1)	97.6 (0.1)	97.7 (0.0)
PR Laplace	95.4 (0.0)	97.2 (0.0)	97.3 (0.0)	97.4 (0.0)	97.3 (0.0)
eikonal	82.3 (1.0)	89.0 (0.5)	90.6 (0.4)	93.4 (0.1)	93.7 (0.1)
PR eikonal	85.6 (0.0)	92.4 (0.0)	93.6 (0.0)	95.1 (0.0)	95.0 (0.0)

Inverse problem?

To formulate the active learning problem, consider Laplace learning with $\Gamma \subset \mathcal{X}$ labels

(4)
$$\begin{cases} \mathcal{L}u(x) = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u(x) = g(x), & \text{if } x \in \Gamma, \end{cases}$$

and add another label at $z \in \mathcal{X}$:

(5)
$$\begin{cases} \mathcal{L}u_z(x) = 0, & \text{if } x \in \mathcal{X} \setminus \Gamma \\ u_z(x) = g(x), & \text{if } x \in \Gamma \cup \{z\}. \end{cases}$$

We should choose z to minimize $\|u_z - g\|$?

Can we do this efficiently, and under what models for g?

Can the connection to continuum PDEs or Hamilton-Jacobi equations be utilized?

Elliptic regularity on graphs

Some of the most useful tools in PDE theory are regularity results.

Elliptic regularity on graphs

Some of the most useful tools in PDE theory are regularity results.

For the p-Laplacian on a random geometric graph we have the following:

Theorem (Calder, 2018)

If $\mathcal{L}_p u = 0$ *and* $p > d$*, then for every* $0 < \alpha < \frac{p-d}{p-1}$ *there exists* $C, \delta > 0$ *such that*

$$|u(x) - u(y)| \leq C(|x-y|^\alpha + \varepsilon^\alpha) \tag{6}$$

holds for all $x, y \in \mathcal{X}_n$ *with probability at least*

$$1 - \exp\left(-\delta n \varepsilon^q + C\log(n)\right),$$

where $q = \max\{d+4, 3d/2\}$.

Elliptic regularity on graphs

Some of the most useful tools in PDE theory are regularity results.

For the p-Laplacian on a random geometric graph we have the following:

Theorem (Calder, 2018)

If $\mathcal{L}_p u = 0$ and $p > d$, then for every $0 < \alpha < \frac{p-d}{p-1}$ there exists $C, \delta > 0$ such that

$$|u(x) - u(y)| \leq C(|x-y|^\alpha + \varepsilon^\alpha) \tag{6}$$

holds for all $x, y \in \mathcal{X}_n$ with probability at least

$$1 - \exp\left(-\delta n \varepsilon^q + C\log(n)\right),$$

where $q = \max\{d+4, 3d/2\}$.

A similar result is implicit in [Slepčev & Thrope, 2019].

Outline

Elliptic regularity on graphs

Question: Given a random geometric graph model, how regular are solutions of graph Poisson equations

$$\mathcal{L}u(x) = f(x) \quad \text{for } x \in \mathcal{X}? \tag{7}$$

Regularity can be Hölder, Lipschitz, $C^{k,\alpha}$ or Sobolev spaces.

Elliptic regularity on graphs

Question: Given a random geometric graph model, how regular are solutions of graph Poisson equations

$$\mathcal{L}u(x) = f(x) \quad \text{for } x \in \mathcal{X}? \tag{7}$$

Regularity can be Hölder, Lipschitz, $C^{k,\alpha}$ or Sobolev spaces.

A very preliminary result in the manifold setting:

Theorem (Calder, Lewicka, Trillos 2020)

With probability greater than $1 - n^k \exp(-cn\varepsilon^{d+4})$ *solutions of* (7) *satisfy*

$$|u(x) - u(y)| \leq C(\|f\|_\infty + \|u\|_\infty)(|x-y| + \varepsilon)$$

for all $x, y \in \mathcal{X} \cap \mathcal{M}$.

The proof of the theorem uses stochastic coupling of random walks.

Elliptic regularity on graphs

Question: Given a random geometric graph model, how regular are solutions of graph Poisson equations

$$\mathcal{L}u(x) = f(x) \quad \text{for } x \in \mathcal{X}? \tag{7}$$

Regularity can be Hölder, Lipschitz, $C^{k,\alpha}$ or Sobolev spaces.

A very preliminary result in the manifold setting:

Theorem (Calder, Lewicka, Trillos 2020)

With probability greater than $1 - n^k \exp(-cn\varepsilon^{d+4})$ *solutions of* (7) *satisfy*

$$|u(x) - u(y)| \leq C(\|f\|_\infty + \|u\|_\infty)(|x-y| + \varepsilon)$$

for all $x, y \in \mathcal{X} \cap \mathcal{M}$.

The proof of the theorem uses stochastic coupling of random walks.

A direct application is L^∞ spectral convergence rates.

My chalkboard tutorial talk is based off Chapter 5 in the Calculus of Variations lecture notes available on my personal website:

http://www-users.math.umn.edu/~jwcalder/CalculusOfVariations.pdf