Graph-Based Learning: Theory and Applications

Jeff Calder

School of Mathematics
University of Minnesota

Mathematics of Machine Learning Course
Brigham Young University
March 5, 2021

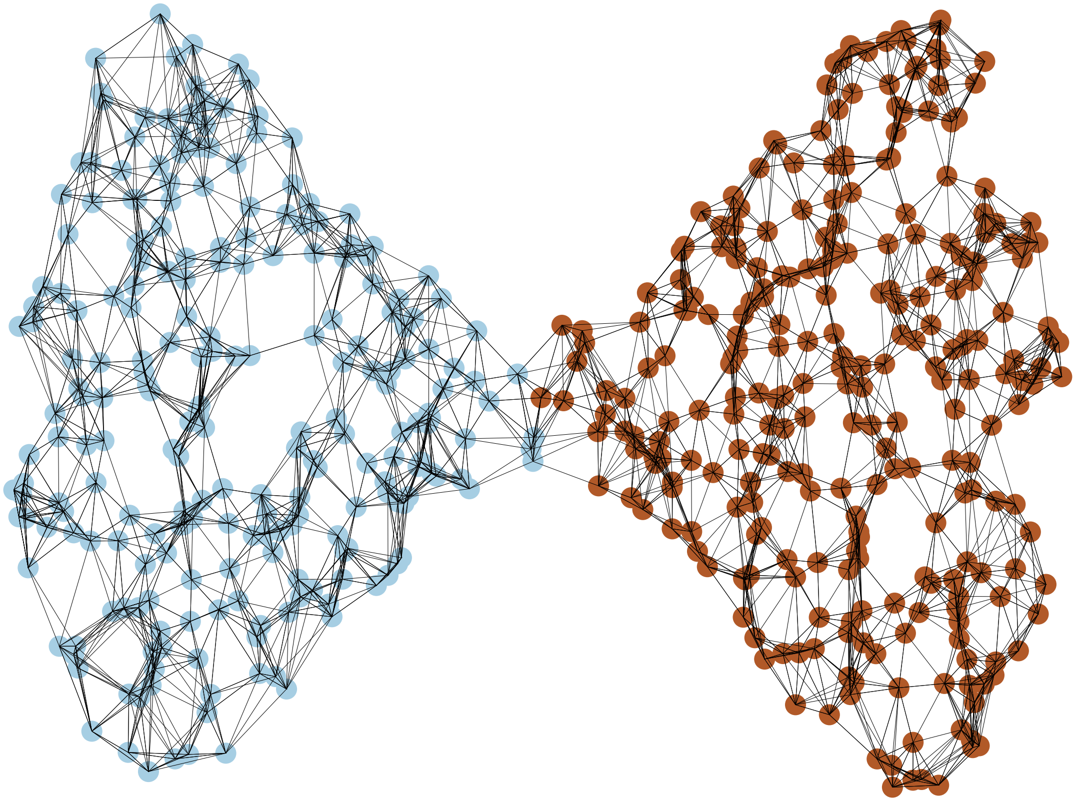
Outline

Outline

Graph-based learning

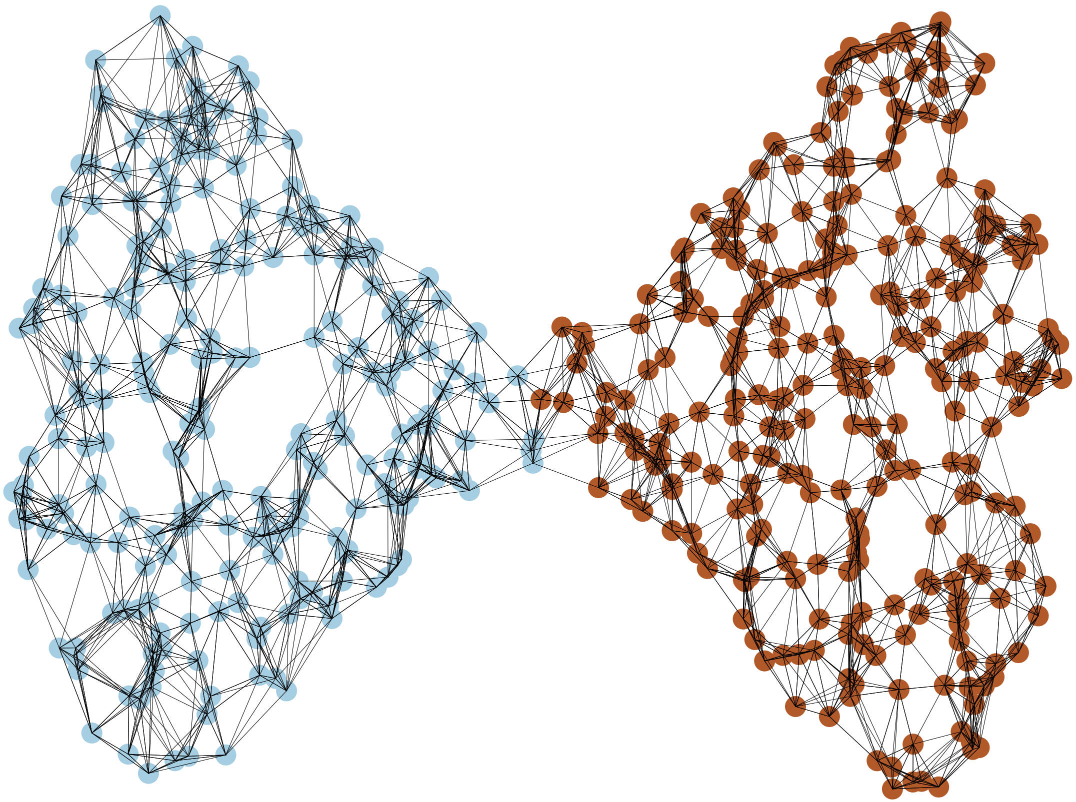
Let $(\mathcal{X}, \mathcal{W})$ be a graph.

- $\mathcal{X} \subset \mathbb{R}^d$ are the vertices.
- $\mathcal{W} = (w_{xy})_{x,y\in\mathcal{X}}$ are nonnegative edge weights.
- w_{xy} is large when x and y are similar, and small or $w_{xy} = 0$ otherwise.

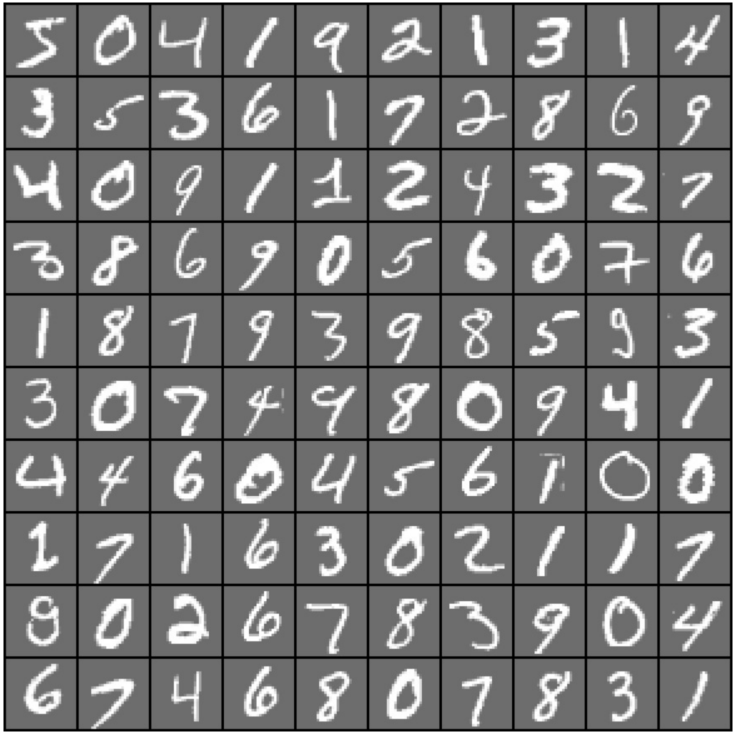


Some common graph-based learning tasks

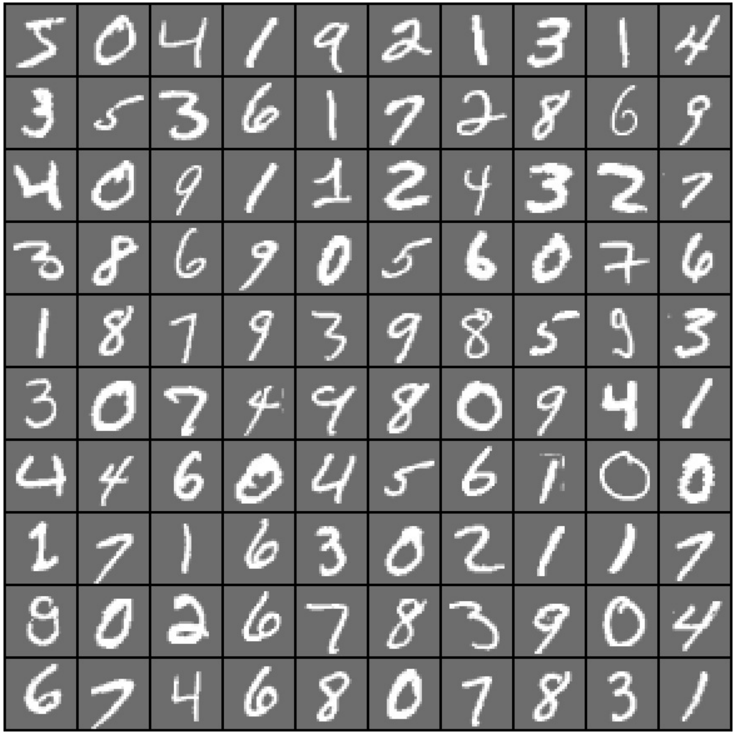
1. Clustering (grouping similar datapoints)
2. Semi-supervised learning (propagating labels)
3. Dimension reduction (spectral embeddings)



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



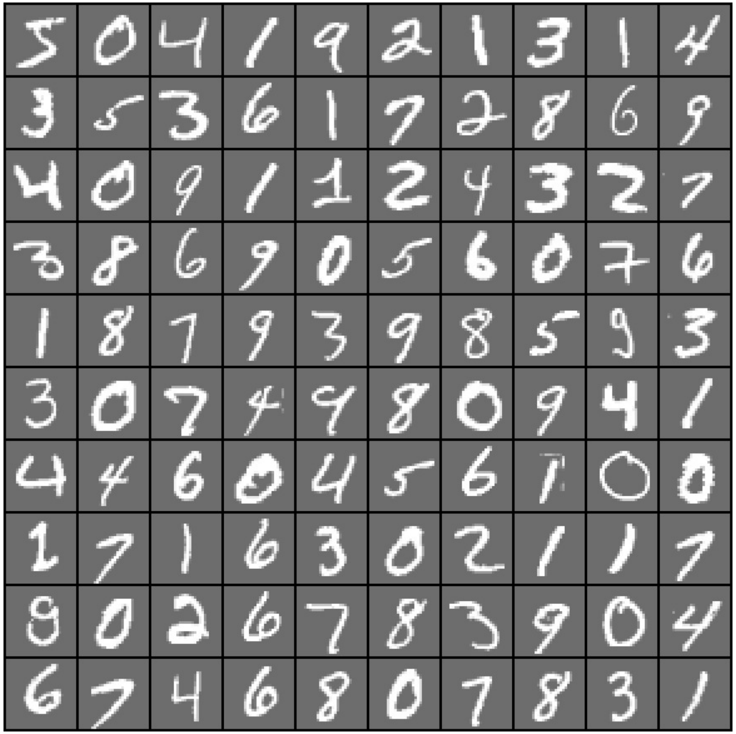
MNIST (70,000 28×28 pixel images of digits 0-9)



- Each image is a datapoint

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

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



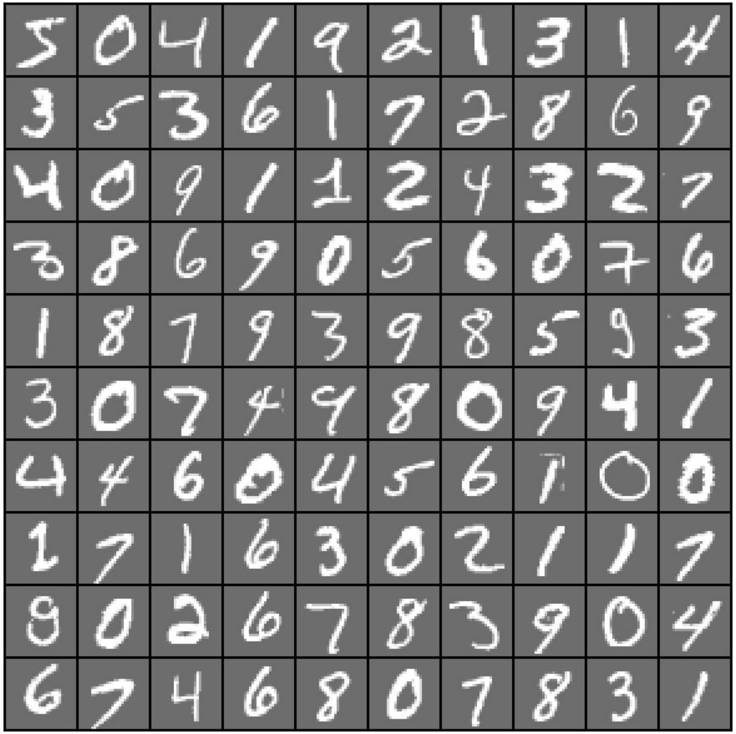
- Each image is a datapoint

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

- Geometric weights:

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

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:

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

- k-nearest neighbor graph:

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

Clustering MNIST

https://divamgupta.com

Graph cuts

Question: How do we cluster graph data?

Graph cuts

Question: How do we cluster graph data?

Consider binary clustering (two classes). We can try to minimize a graph cut energy

$$\text{(Min-Cut)} \qquad \min_{A \subset \mathcal{X}} \mathrm{Cut}(A) := \sum_{\substack{x,y \in \mathcal{X} \\ x \in A, y \notin A}} w_{xy}.$$

Graph cuts

Question: How do we cluster graph data?

Consider binary clustering (two classes). We can try to minimize a graph cut energy

$$\text{(Min-Cut)} \qquad \min_{A \subset \mathcal{X}} \mathrm{Cut}(A) := \sum_{\substack{x,y \in \mathcal{X} \\ x \in A, y \notin A}} w_{xy}.$$

Tends to produce unbalanced classes (e.g., $A = \{x\}$).

Graph cuts

Question: How do we cluster graph data?

Consider binary clustering (two classes). We can try to minimize a graph cut energy

$$\text{(Balanced-Cut)} \qquad \min_{A \subset \mathcal{X}} \frac{\mathrm{Cut}(A)}{\mathrm{Vol}(A)\mathrm{Vol}(\mathcal{X} \setminus A)},$$

where

$$\mathrm{Vol}(A) = \sum_{x \in A} \sum_{y \in X} w_{xy}.$$

Graph cuts

Question: How do we cluster graph data?

Consider binary clustering (two classes). We can try to minimize a graph cut energy

$$\text{(Balanced-Cut)} \qquad \min_{A \subset \mathcal{X}} \frac{\mathrm{Cut}(A)}{\mathrm{Vol}(A)\mathrm{Vol}(\mathcal{X} \setminus A)},$$

where

$$\mathrm{Vol}(A) = \sum_{x \in A} \sum_{y \in X} w_{xy}.$$

Gives good clusterings but very computationally hard (NP-hard).

Spectral clustering

For $A \subset \mathcal{X}$ set

$$u(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise.} \end{cases}$$

Spectral clustering

For $A \subset \mathcal{X}$ set

$$u(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise.} \end{cases}$$

Then we have

$$\mathrm{Cut}(A) = \sum_{\substack{x,y\in\mathcal{X} \\ x\in A, y\notin A}} w_{xy} = \frac{1}{2}\sum_{x,y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2$$

and

$$\mathrm{Vol(A)} = \sum_{x,y\in\mathcal{X}} w_{xy}u(x).$$

Spectral clustering

For $A \subset \mathcal{X}$ set

$$u(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{otherwise.} \end{cases}$$

Then we have

$$\mathrm{Cut}(A) = \sum_{\substack{x,y \in \mathcal{X} \\ x \in A, y \notin A}} w_{xy} = \frac{1}{2} \sum_{x,y \in \mathcal{X}} w_{xy}(u(x) - u(y))^2$$

and

$$\mathrm{Vol(A)} = \sum_{x,y \in \mathcal{X}} w_{xy} u(x).$$

This allow us to write the balanced cut problem as

$$\min_{u:\mathcal{X} \to \{0,1\}} \frac{\displaystyle\sum_{x,y \in \mathcal{X}} w_{xy}(u(x) - u(y))^2}{\displaystyle\sum_{x,y,x',y' \in \mathcal{X}} u(x) w_{xy} (1 - u(y')) w_{x'y'}}.$$

Spectral clustering

Consider solving the similar, relaxed, problem

$$\min_{\substack{u:\mathcal{X}\to\mathbb{R} \\ \sum_{x\in\mathcal{X}} u(x)\neq 0}} \frac{\displaystyle\sum_{x,y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2}{\displaystyle\sum_{x\in\mathcal{X}} u(x)^2}.$$

Spectral clustering

Consider solving the similar, relaxed, problem

$$\min_{\substack{u:\mathcal{X}\to\mathbb{R} \\ \sum_{x\in\mathcal{X}} u(x)\neq 0}} \frac{\displaystyle\sum_{x,y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2}{\displaystyle\sum_{x\in\mathcal{X}} u(x)^2}.$$

The solution is the smallest non-trivial eigenvector (Fiedler vector) of the graph Laplacian

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

Spectral clustering

Consider solving the similar, relaxed, problem

$$\min_{\substack{u:\mathcal{X}\to\mathbb{R} \\ \sum_{x\in\mathcal{X}} u(x)\neq 0}} \frac{\displaystyle\sum_{x,y\in\mathcal{X}} w_{xy}(u(x)-u(y))^2}{\displaystyle\sum_{x\in\mathcal{X}} u(x)^2}.$$

The solution is the smallest non-trivial eigenvector (Fiedler vector) of the graph Laplacian

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

Binary spectral clustering:

1. Compute Fiedler vector $u : \mathcal{X} \to \mathbb{R}$.
2. Set $A = \{x \in \mathcal{X} \ : \ u(x) > 0\}$.

Spectral clustering

Spectral clustering: To cluster into k groups:

1. Compute first k eigenvectors of the graph Laplacian $\mathcal{L}$:

$$u_1, \ldots, u_k : \mathcal{X} \to \mathbb{R}.$$

Spectral clustering

Spectral clustering: To cluster into k groups:

1. Compute first k eigenvectors of the graph Laplacian $\mathcal{L}$:

$$u_1, \dots, u_k : \mathcal{X} \to \mathbb{R}.$$

2. Define the spectral embedding $\Psi : \mathcal{X} \to \mathbb{R}^k$ by

$$\Psi(x) = (u_1(x), u_2(x), \dots, u_k(x)).$$

Spectral clustering

Spectral clustering: To cluster into k groups:

1. Compute first k eigenvectors of the graph Laplacian $\mathcal{L}$:
$$u_1, \dots, u_k : \mathcal{X} \to \mathbb{R}.$$
2. Define the spectral embedding $\Psi : \mathcal{X} \to \mathbb{R}^k$ by
$$\Psi(x) = (u_1(x), u_2(x), \dots, u_k(x)).$$
3. Cluster the point cloud $\mathcal{Y} = \Psi(\mathcal{X})$ with your favorite clustering algorithm (often k-means).

Spectral methods in data science

Spectral methods are widely used for dimension reduction and clustering in data science and machine learning.

- Spectral clustering [Shi and Malik (2000)] [Ng, Jordan, and Weiss (2002)]
- Laplacian eigenmaps [Belkin and Niyogi (2003)]
- Diffusion maps [Coifman and Lafon (2006)]

Outline

Graph-based semi-supervised learning

Given:

- Graph $(\mathcal{X}, \mathcal{W})$
- Labeled nodes $\Gamma \subset \mathcal{X}$ and labels $g : \Gamma \to \mathbb{R}^k$,
- The i^{th} class has label vector $g(x) = e_i = (0, , \dots, 0, 1, 0, \dots, 0)$.

Graph-based semi-supervised learning

Given:

- Graph $(\mathcal{X}, \mathcal{W})$
- Labeled nodes $\Gamma \subset \mathcal{X}$ and labels $g : \Gamma \to \mathbb{R}^k$,
- The i^{th} class has label vector $g(x) = e_i = (0, , \ldots, 0, 1, 0, \ldots, 0)$.

Task: Extend the labels to the rest of the graph $\mathcal{X} \setminus \Gamma$.

Graph-based semi-supervised learning

Given:

- Graph $(\mathcal{X}, \mathcal{W})$
- Labeled nodes $\Gamma \subset \mathcal{X}$ and labels $g : \Gamma \to \mathbb{R}^k$,
- The i^{th} class has label vector $g(x) = e_i = (0, , \ldots, 0, 1, 0, \ldots, 0)$.

Task: Extend the labels to the rest of the graph $\mathcal{X} \setminus \Gamma$.

Semi-supervised: Goal is to use both the labeled and unlabeled data to get good performance with far fewer labels than required by fully-supervised learning.

Graph-based semi-supervised learning

Given:

- Graph $(\mathcal{X}, \mathcal{W})$
- Labeled nodes $\Gamma \subset \mathcal{X}$ and labels $g : \Gamma \to \mathbb{R}^k$,
- The i^{th} class has label vector $g(x) = e_i = (0, , \dots, 0, 1, 0, \dots, 0)$.

Task: Extend the labels to the rest of the graph $\mathcal{X} \setminus \Gamma$.

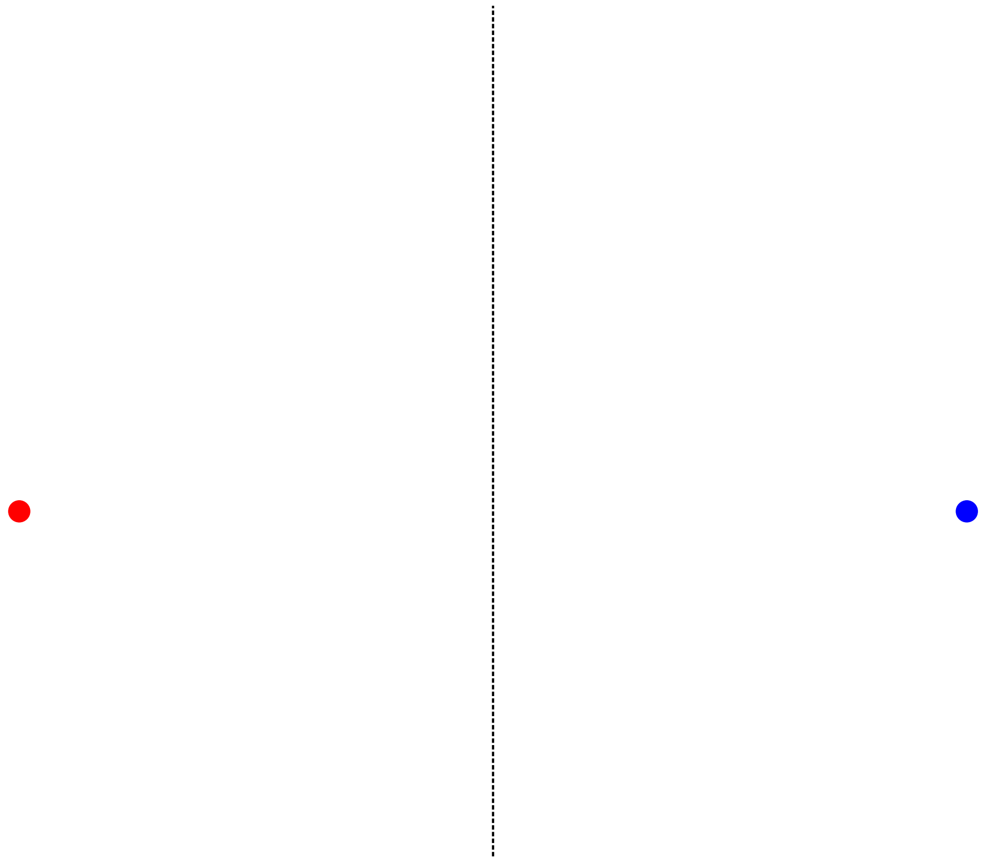
Semi-supervised: Goal is to use both the labeled and unlabeled data to get good performance with far fewer labels than required by fully-supervised learning.

Applications of semi-supervised learning

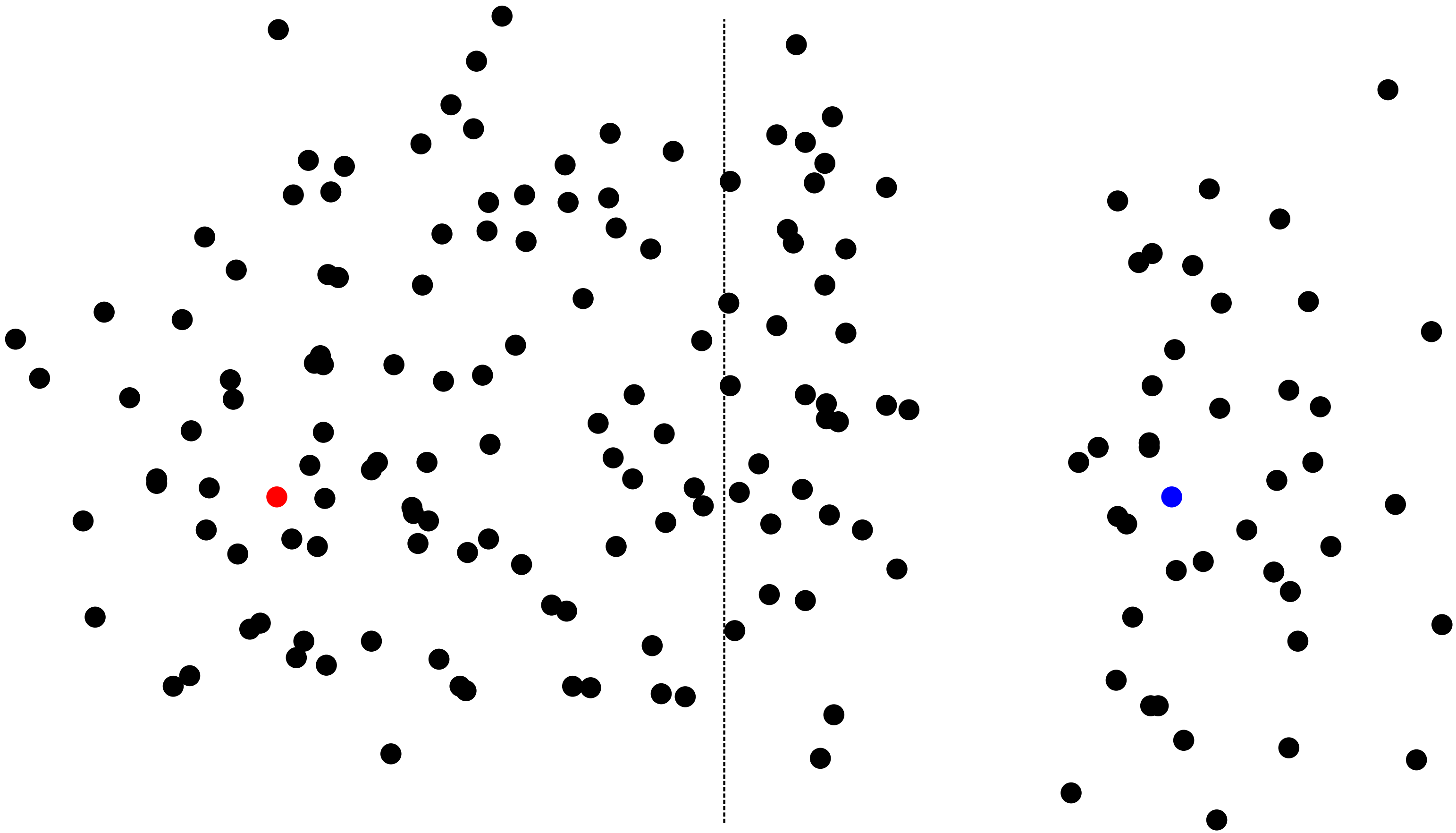
1. Speech recognition
2. Classification (images, video, website, etc.)
3. Inferring protein structure from sequencing

Why semi-supervised?

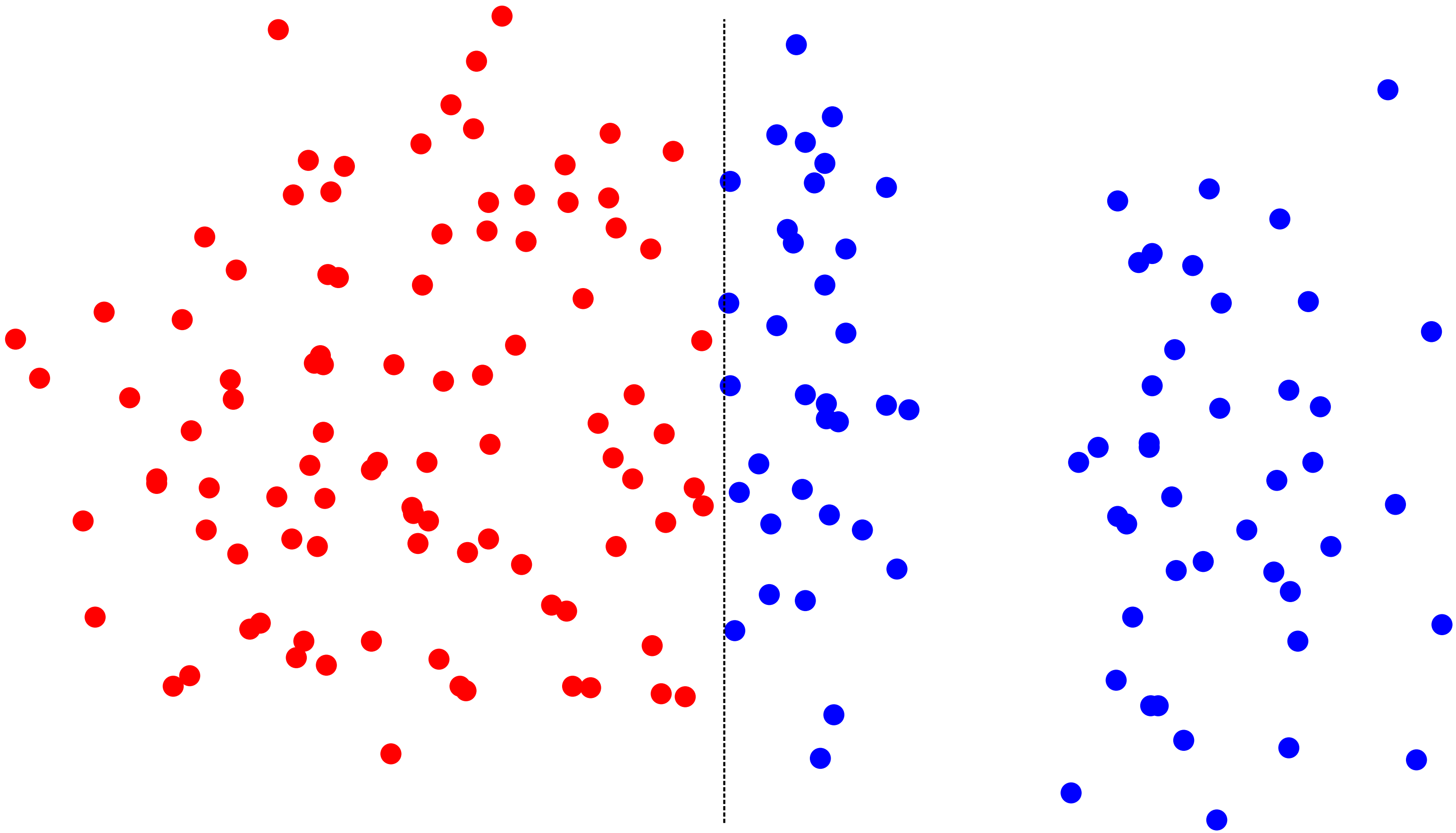
Why semi-supervised?



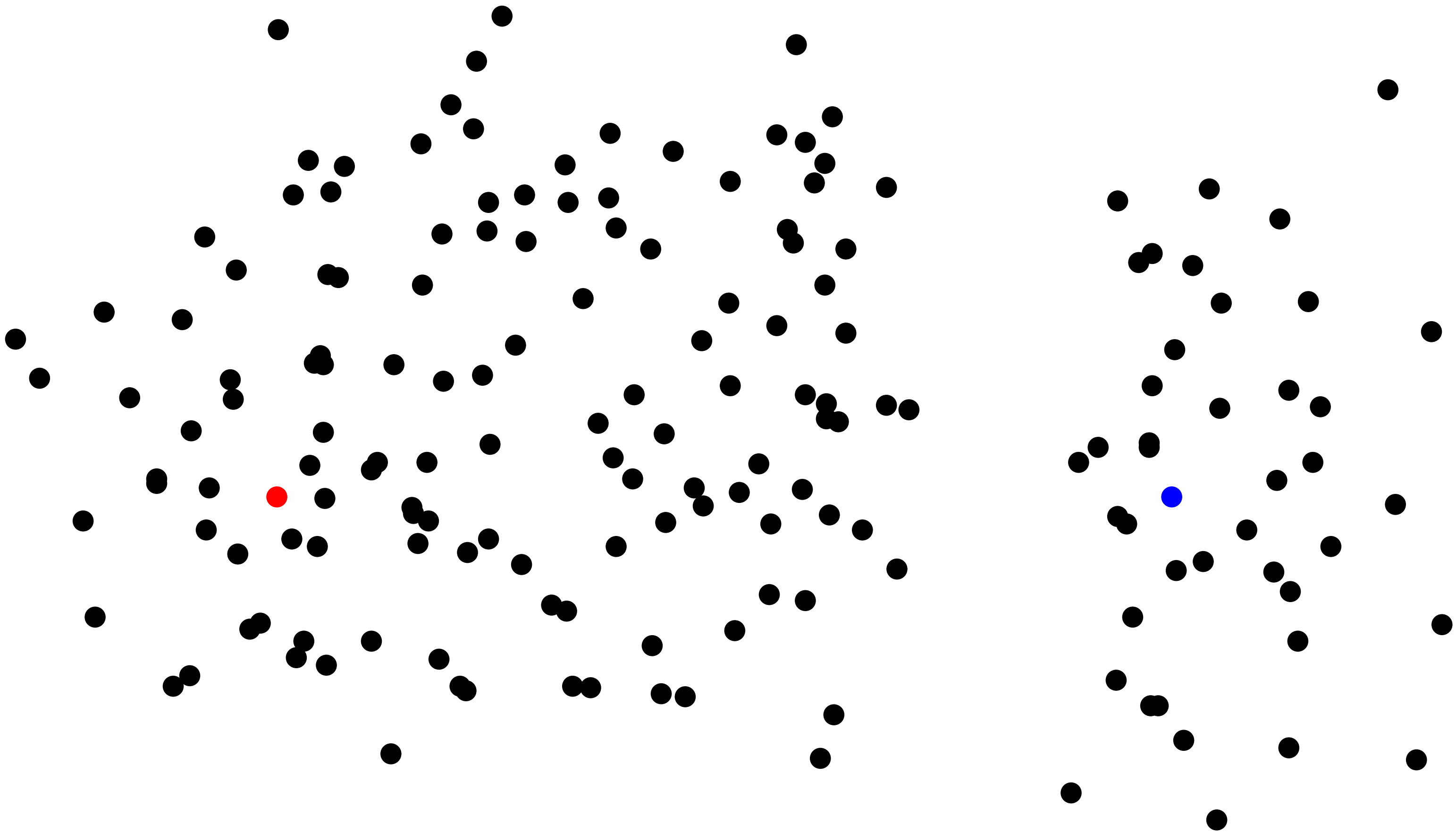
Why semi-supervised?



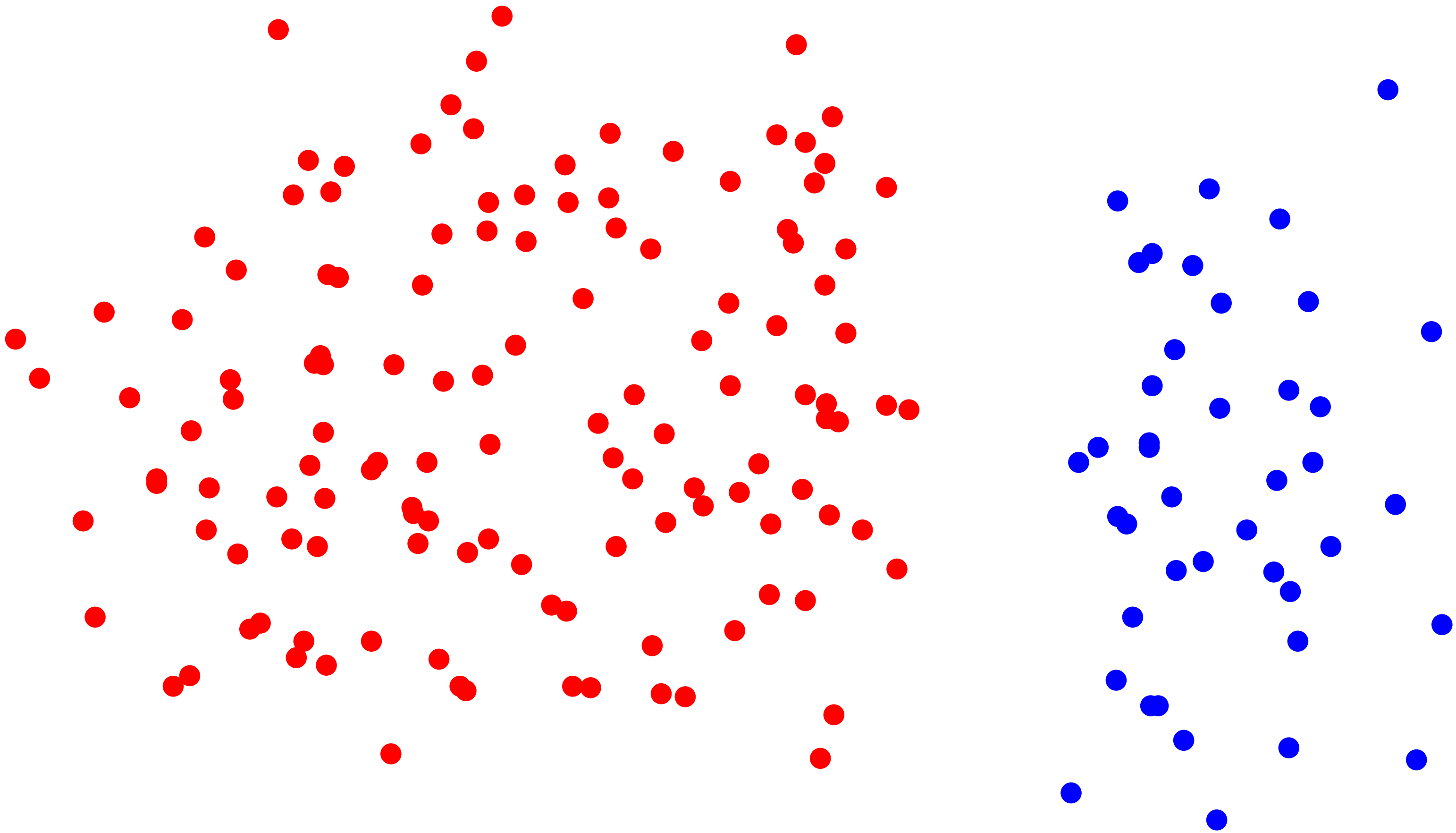
Why semi-supervised?



Why semi-supervised?



Why semi-supervised?



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

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

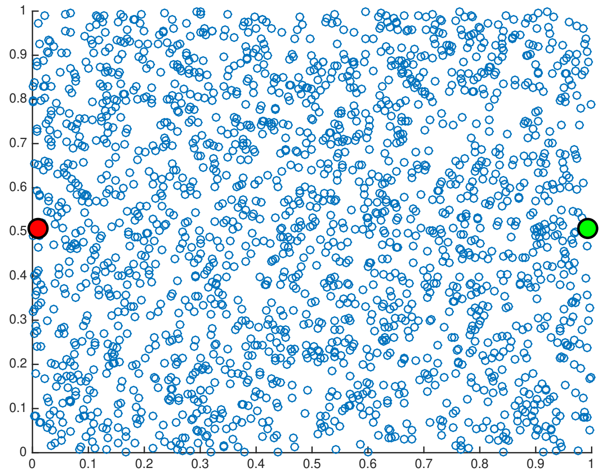
The label decision for vertex $x \in \mathcal{X}$ is determined by the largest component of $u(x)$

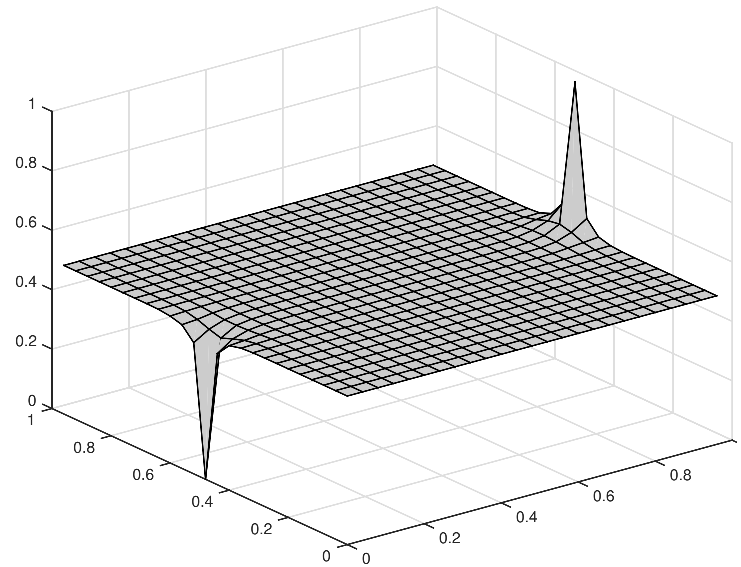
$$\ell(x) = \operatorname*{argmax}_{j \in \{1,\dots,k\}} \{u_j(x)\}.$$

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]

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]

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, ICML 2020]

Poisson learning

At low label rates one should replace Laplace learning

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

with Poisson learning

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

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

Poisson learning

At low label rates one should replace Laplace learning

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

with Poisson learning

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

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:

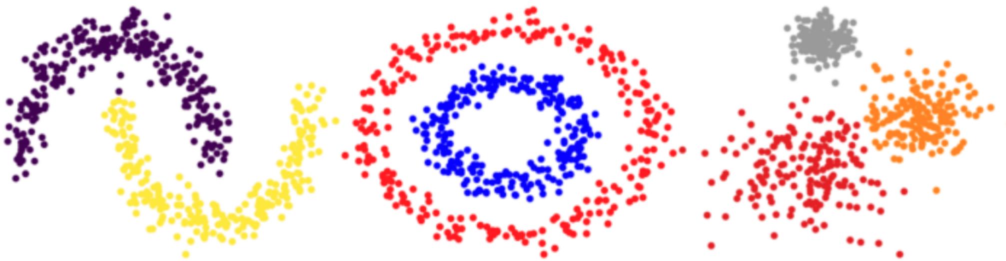
$$\ell(x) = \operatorname*{argmax}_{j \in \{1,\ldots,k\}} \{u_j(x)\}.$$

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.

Outline

GraphLearning Python Package (Click Here)

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

Outline

Pointwise consistency on random geometric graphs

Let $X_1, X_2, \ldots, X_n$ be a sequence of i.i.d random variables on $\Omega \subset \mathbb{R}^d$ with density $\rho \in C^2(\Omega)$, where Ω is open and bounded with a smooth boundary, and $\rho \geq \rho_{min} > 0$.

The random geometric graph Laplacian applied to $u : \Omega \to \mathbb{R}$ is

$$\mathcal{L}u(x) = \sum_{i=1}^{n} \eta\left(\frac{|X_i - x|}{\varepsilon}\right)(u(X_i) - u(x)),$$

where $\varepsilon > 0$ is the connectivity length scale (bandwidth) and $\eta : \mathbb{R} \to \mathbb{R}$ is smooth, nonnegative and has compact support in $[0, 1]$.

Pointwise consistency on random geometric graphs

Let $X_1, X_2, \ldots, X_n$ be a sequence of i.i.d random variables on $\Omega \subset \mathbb{R}^d$ with density $\rho \in C^2(\Omega)$, where Ω is open and bounded with a smooth boundary, and $\rho \geq \rho_{min} > 0$.

The random geometric graph Laplacian applied to $u : \Omega \to \mathbb{R}$ is

$$\mathcal{L}u(x) = \sum_{i=1}^{n} \eta\left(\frac{|X_i - x|}{\varepsilon}\right)(u(X_i) - u(x)),$$

where $\varepsilon > 0$ is the connectivity length scale (bandwidth) and $\eta : \mathbb{R} \to \mathbb{R}$ is smooth, nonnegative and has compact support in $[0, 1]$.

Today we'll prove that when u is C^3 we have

$$\frac{2}{\sigma_\eta n \varepsilon^{d+2}} \mathcal{L}u(x) = \rho^{-1} \mathrm{div}\left(\rho^2 \nabla u\right) + \underbrace{O\left(n^{-1/2}\varepsilon^{-1-d/2}\right)}_{\text{Variance}} + \underbrace{O(\varepsilon)}_{\text{Bias}}.$$

with high probability, provided $B(x, \varepsilon) \subset \Omega$.

Discrete to continuum convergence

Manifold assumption: Let $x_1, \ldots, x_n$ 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

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

Discrete to continuum convergence

Manifold assumption: Let $x_1, \ldots, x_n$ 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

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

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

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

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

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

Concentration of measure

Theorem (Bernstein's inequality)

Let $Y_1, \dots, Y_n$ 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 $|Y_i| \le M$ almost surely for all i. Then for any $t > 0$

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

Concentration of measure

Theorem (Bernstein's inequality)

Let $Y_1, \dots, Y_n$ 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 $t > 0$

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

Let $\delta > 0$ and choose $t > 0$ so that $\delta = 2\exp\left(-\frac{nt^2}{2\sigma^2+4Mt/3}\right)$. Then we get

$$\left|\frac{1}{n}\sum_{i=1}^n Y_i - \mu\right| \leq \sqrt{\frac{2\sigma^2|\log\frac{\delta}{2}|}{n}} + \frac{4M|\log\frac{\delta}{2}|}{3n}$$

with probability at least $1 - \delta$.

Concentration of measure

Theorem (Bernstein's inequality)

Let $Y_1, \dots, Y_n$ 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 $t > 0$

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

Let $\delta > 0$ and choose $t > 0$ so that $\delta = 2\exp\left(-\frac{nt^2}{2\sigma^2+4Mt/3}\right)$. Then we get

$$\left|\frac{1}{n}\sum_{i=1}^n Y_i - \mu\right| \leq \sqrt{\frac{2\sigma^2|\log\frac{\delta}{2}|}{n}} + \frac{4M|\log\frac{\delta}{2}|}{3n}$$

with probability at least $1-\delta$. Provided $M \leq C\sqrt{n}\sigma$ we can write

$$\frac{1}{n}\sum_{i=1}^n Y_i = \mu + O\left(\sqrt{\frac{\sigma^2}{n}}\right) \quad \text{w.h.p.}$$

Proof of Pointwise consistency

$$\mathcal{L}u(x) = \sum_{i=1}^{n} \underbrace{\eta\left(\frac{|x_i - x|}{\varepsilon}\right)\left(u(x_i) - u(x)\right)}_{Y_i}$$

$$|Y_i| \leq C\varepsilon = M.$$

$$\sigma^2 = \mathrm{Var}(Y_i) \sim \int_{B(x,\varepsilon)} \eta\left(\frac{|y-x|}{\varepsilon}\right)^2 \overbrace{\left(u(y) - u(x)\right)^2}^{O(\varepsilon^2)} \rho(x)^2 \, dx$$

$$\sim \varepsilon^{d+2}$$

Bernstein:

$$\frac{1}{n}\mathcal{L}u(x) = \overbrace{\int_{B(x,\varepsilon)} \eta\left(\frac{|x-y|}{\varepsilon}\right)(u(y)-u(x))\,\rho(y)\,dy}^{Au(x)} + \underbrace{O\left(\sqrt{\frac{\varepsilon^{d+2}}{n}}\right)}_{\text{variance}}.$$

Taylor expansions in $Au(x)$ after $z = \frac{x-y}{\varepsilon}$

$$Au(x) = \varepsilon^d \int_{B(0,1)} \eta(|z|)(u(x+\varepsilon z)-u(x))\,\rho(x+\varepsilon z)\,dz$$

$$\rho(x+\varepsilon z) = \rho(x) + \varepsilon \nabla\rho(x)\cdot z + O(\varepsilon^2)$$

$$u(x+\varepsilon z) - u(x) = \varepsilon \nabla u(x)\cdot z + \frac{\varepsilon^2}{2} z^T \nabla^2 u(x) z + O(\varepsilon^3)$$

① $\cancel{\varepsilon \nabla u(x)\cdot z \, \rho(x)}$ odd function over $B(0,1) \Longrightarrow 0$

② $\varepsilon^2 (\nabla\rho(x)\cdot z)(\nabla u(x)\cdot z)$

③ $\rho(x)\frac{\varepsilon^2}{2} z^T \nabla^2 u(x) z$

$$② \int_{B(0,1)} \eta(|z|)\,(\nabla \rho(x)\cdot z)\,(\nabla u(x)\cdot z)\,dz$$

$$= \sum_{i,j=1}^{n} \rho_{x_i}(x)\, u_{x_j}(x) \underbrace{\int_{B(0,1)} \eta(|z|)\, z_i z_j \, dz}_{\substack{= 0 \text{ if } i \neq j \\ =: \sigma_\eta \text{ if } i = j}}$$

$$= \sigma_\eta \nabla \rho(x) \cdot \nabla u(x)$$

(3)

$$\frac{1}{2}\rho(x)\int_{B(0,1)} \eta(|z|)\, z^T \nabla^2 u(x)\, z \, dz$$

$$= \frac{1}{2}\rho(x)\sum_{i,j=1}^{n} u_{x_i x_j}(x) \int_{B(0,1)} \eta(|z|)\, z_i z_j \, dz$$

$$= \frac{1}{2}\rho(x)\sum_{i=1}^{n} u_{x_i x_i}\sigma_\eta$$

$$= \frac{\sigma_\eta}{2}\rho(x)\,\Delta u(x).$$

Hence

$$Au(x) = \varepsilon^{d+2}\sigma_\eta\left(\frac{1}{2}\rho\Delta u + \nabla u\cdot\nabla\rho\right) + O(\varepsilon^{d+3})$$

$$= \frac{\sigma_\eta}{2}\varepsilon^{d+2}\rho^{-1}\left(\rho^2\Delta u + 2\rho\nabla\rho\cdot\nabla u\right) + O(\varepsilon^{d+3})$$

$$= \frac{\sigma_\eta}{2}\varepsilon^{d+2}\rho^{-1}\,\mathrm{div}(\rho^2\nabla u) + O(\varepsilon^{d+3})$$

Therefore

$$\frac{1}{n}\mathcal{L}u(x) = \frac{\sigma_\eta}{2}\varepsilon^{d+2}\rho^{-1}\operatorname{div}(\rho^2\nabla u) + O(\varepsilon^{d+3})$$

$$+ O\left(\sqrt{\frac{\varepsilon^{d+2}}{n}}\right)$$

and so

Bias

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

$$+ O\left(\sqrt{\frac{1}{n\varepsilon^{d+2}}}\right)$$

Variance.