Discrete to continuum convergence rates in graph-based learning at percolation length scales

Jeff Calder

School of Mathematics
University of Minnesota

AMS Special Session on Data Science at the Crossroads of Analysis, Geometry, and Topology II

Jan 5, 2023

Joint work with: Leon Bungert (Hausdorff Center for Mathematics, University of Bonn) and Tim Roith (Friedrich-Alexander-Universität Erlangen-Nürnberg)

Research supported by NSF-DMS grant 1713691

Graph-based learning

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

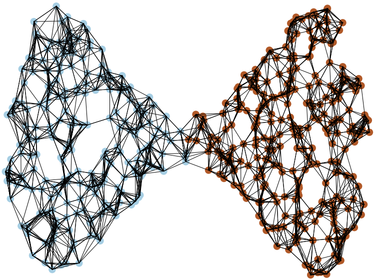
- Vertices $\mathcal{X} \subset \mathbb{R}^d$.
- Nonnegative edge weights $\mathcal{W} = (w_{xy})_{x,y \in \mathcal{X}}$.

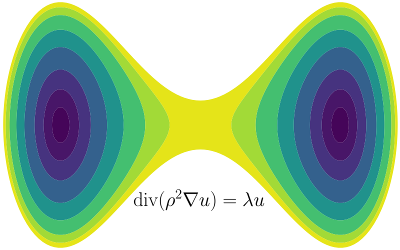
Some common graph-based learning tasks:

1. Clustering
2. Semi-supervised learning
3. Data Depth
4. Link prediction
5. Ranking

Applications of graph-based learning:

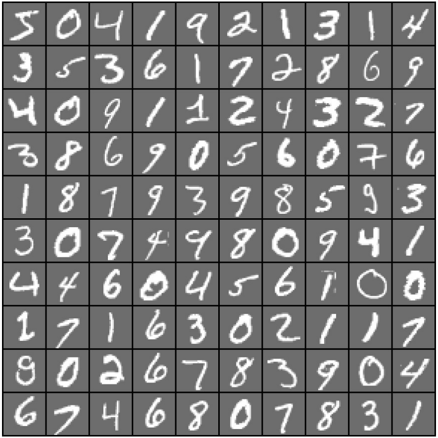
1. Image classification
2. Social media networks
3. Biological networks
4. Drug discovery
5. Wireless networks





Similarity graphs

Some MNIST Digits



- 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:

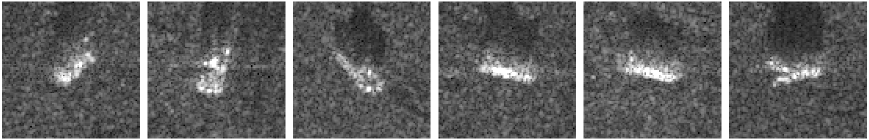
$$w_{xy} = \eta\left(\frac{|x-y|}{\varepsilon_k(x)}\right)$$

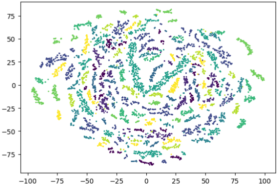
- Often $\eta(t) = e^{-t^2}$.

Similarity graphs via deep learning

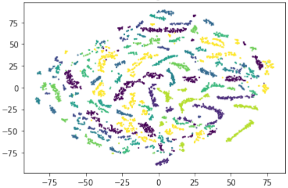
Set $w_{xy} = \eta\left(\frac{|\Psi(x)-\Psi(y)|}{\varepsilon}\right)$ where $\Psi : \mathbb{R}^d \to \mathbb{R}^N$ is learned.

Synthetic Aperture Radar (SAR) Images

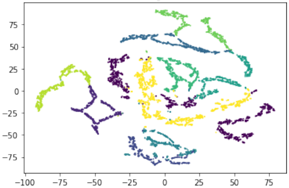




Raw Pixels



Autoencoder Embedding



Contrastive (SimCLR) Embedding

Calder, J., Cook, B., Thorpe, M., & Slepcev, D. (2020, November). **Poisson learning: Graph based semi-supervised learning at very low label rates**. In International Conference on Machine Learning (pp. 1306-1316). PMLR.

Miller, K., Mauro, J., Setiadi, J., Baca, X., Shi, Z., Calder, J., & Bertozzi, A. L. (2022, May). **Graph-based active learning for semi-supervised classification of SAR data**. In Algorithms for Synthetic Aperture Radar Imagery XXIX (Vol. 12095, pp. 126-139). SPIE.

Graph-based semi-supervised learning

Given: Graph $(\mathcal{X}, \mathcal{W})$, labeled nodes $\Gamma \subset \mathcal{X}$, and labels $g : \Gamma \to \mathbb{R}^k$.

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.

A common method is Laplacian regularized learning, which solves the 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

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

There are many other methods based on different graph PDEs or normalizations of the graph Laplacian.

Zhu, X., Ghahramani, Z., & 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) (pp. 912-919).

Spectral clustering

Spectral clustering: To cluster into k groups:

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

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

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

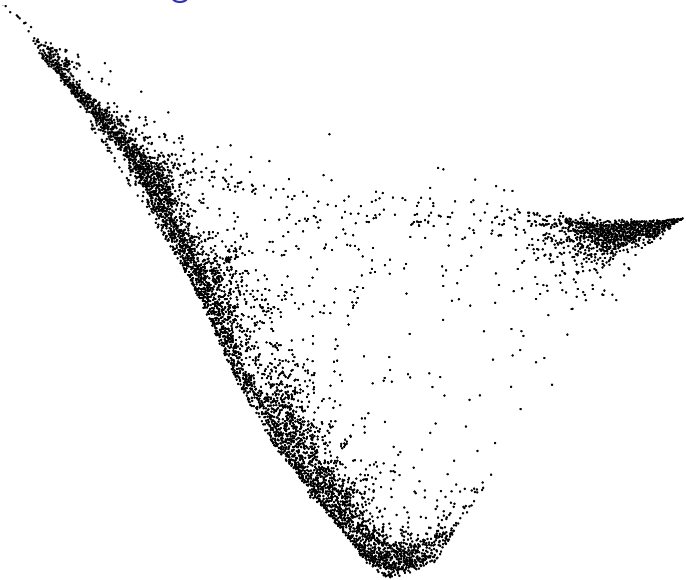
$$\Psi(x) = (u_1(x), u_2(x), \ldots, u_k(x)).$$

3. Cluster the point cloud $\mathcal{Y} = \Psi(\mathcal{X})$ with your favorite clustering algorithm.

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

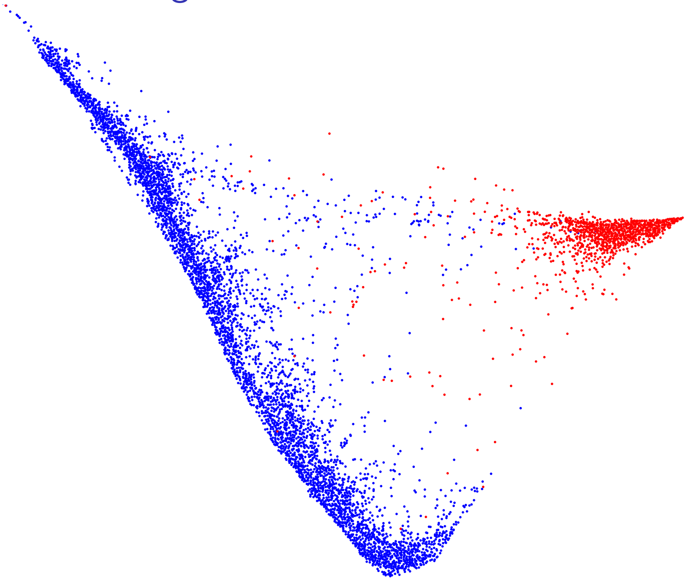
- Spectral clustering [Shi and Malik (2000)] [Ng, Jordan, and Weiss (2002)]
- Laplacian eigenmaps [Belkin and Niyogi (2003)]
- Diffusion maps [Coifman and Lafon (2006)]

Spectral embedding: MNIST



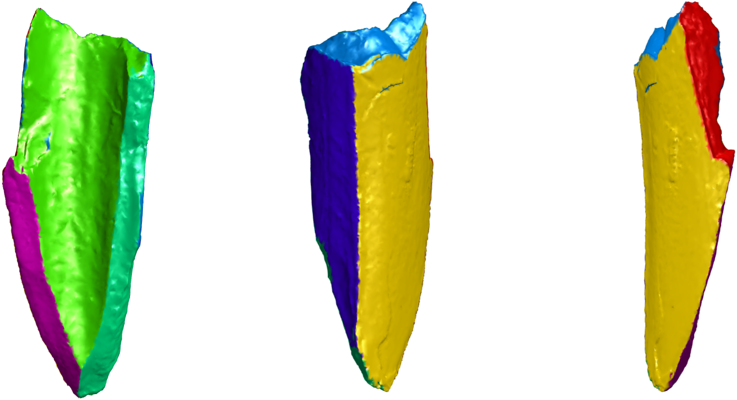
Digits 1 and 2 from MNIST visualized with spectral projection

Spectral embedding: MNIST



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

Application: Segmenting broken bone fragments



Spectral 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.

Discrete to continuum convergence

Let $\mathcal{X}_n = \{x_1, \ldots, x_n\}$ be an i.i.d. sample from a density ρ on a smooth manifold $\mathcal{M} \subset \mathbb{R}^D$ of dimension d. Define a graph with geometric weights of the form

$$w_{ij} = \eta\left(\varepsilon^{-1}|x_i - x_j|\right).$$

The spectrum of the graph-Laplacian $\mathcal{L}$ converges $(n \to \infty, \varepsilon \to 0)$ to the spectrum of the weighted Laplace-Beltrami operator

$$\Delta_{\mathcal{M}} u = -\rho^{-1}\mathrm{div}_{\mathcal{M}}(\rho^2 \nabla_{\mathcal{M}} u).$$

Sample of spectral convergence results

- Garcia Trillos, Gerlach, Hein, and Slepcev (2018):

$$\|u - u_n\|_{L^2(\mathcal{M})} \leq C\sqrt{\frac{\delta_n}{\varepsilon} + \varepsilon}, \quad \delta_n = \left(\frac{\log n}{n}\right)^{1/d}.$$

- Calder, Garcia Trillos (2022):

$$\|u - u_n\|_{L^2(\mathcal{M})} \leq C\varepsilon, \quad \text{provided } \varepsilon \geq \delta_n^{d/(d+4)}.$$

Problem: Prove quantitative rates at the more practically relevant scaling $\varepsilon \sim \delta_n$.

Loss of pointwise consistency

The graph Laplacian $\mathcal{L}$ is not consistent (nor convergent) when $\varepsilon \sim \delta_n$. At a high level:

$$\begin{aligned}
\mathcal{L}u(x) &= \frac{1}{n\varepsilon^{d+2}\sigma_\eta}\sum_{j=1}^{n}\eta\left(\varepsilon^{-1}|x-x_j|\right)(u(x_j)-u(x)) \\
&= \frac{1}{\varepsilon^{d+2}\sigma_\eta}\int_{B(x,\varepsilon)}\eta\left(\varepsilon^{-1}|x-y|\right)(u(y)-u(x))\rho(y)\,dy + O\left(\sqrt{\frac{\sigma^2}{n}}\right) \\
&= \Delta_\rho u(x) + O\left(\varepsilon + \sqrt{\frac{1}{n\varepsilon^{d+2}}}\right).
\end{aligned}$$

Since $\delta_n^d = \log(n)/n$ we can write the error term as (up to log factors)

$$\mathcal{L}u(x) = \Delta_\rho u(x) + O\left(\varepsilon + \sqrt{\frac{\delta_n^d}{\varepsilon^{d+2}}}\right).$$

To match the $O(\varepsilon)$ error term we need $\delta_n^d \leq \varepsilon^{d+4}$, or

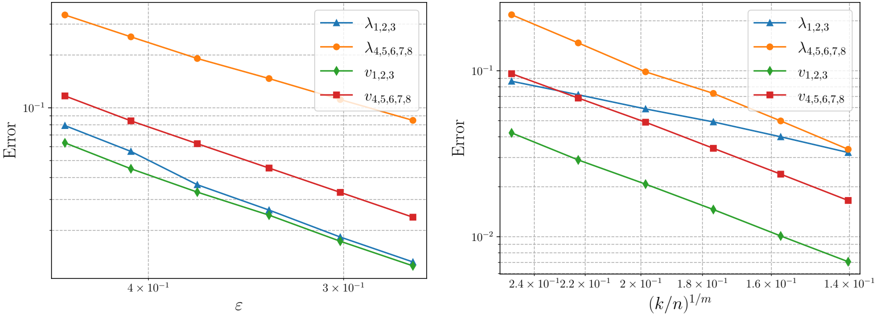
$$\varepsilon \geq \delta_n^{d/(d+4)}.$$

Numerical experiments

Rates of convergence for

$$\varepsilon = \delta_n^{d/(d+2)}$$

of the form $O(\varepsilon^b)$ (value of b is shown) for eigenvalues and eigenvectors of the graph Laplacian on the 2-sphere. Rates are all between $O(\varepsilon^2)$ and $O(\varepsilon^3)$.



(a) ε-graph (b) knn graph

We expect there is some kind of **homogenization** occurring at smaller length scales.

Lipschitz learning

Lipschitz learning performs semi-supervised learning by solving the ∞-Laplace equation

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

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

Rough consistency argument: Assume $w_{xy} = 1_{|x-y|\leq\varepsilon}$.

$$\begin{aligned} \mathcal{L}_\infty u(x) &= \left(\max_{y\in B(x,\varepsilon)} + \min_{y\in B(x,\varepsilon)}\right)(u(y)-u(x)) + O(\delta_n\varepsilon) \\ &= u\left(x+\varepsilon\frac{\nabla u}{|\nabla u|}\right) - 2u(x) + u\left(x-\varepsilon\frac{\nabla u}{|\nabla u|}\right) + O(\delta_n\varepsilon+\varepsilon^3) \\ &= \varepsilon^2\frac{\nabla u^T\nabla^2 u\nabla u}{|\nabla u|^2} + O(\delta_n\varepsilon+\varepsilon^3) = \varepsilon^2\Delta_\infty u(x) + O(\delta_n\varepsilon+\varepsilon^3). \end{aligned}$$

We require $\delta_n\varepsilon \ll \varepsilon^3$ or $\varepsilon \gg \delta_n^{1/2}$ for $O(\varepsilon)$ consistency.

Kyng, R., Rao, A., Sachdeva, S., & Spielman, D. A. (2015, June). Algorithms for Lipschitz learning on graphs. In Conference on Learning Theory (pp. 1190-1223). PMLR.

Discrete to continuum for ∞-Laplacian

Letting $x_1, \ldots, x_n$ be i.i.d. on $\Omega \subset \mathbb{R}^d$, the continuum version of the discrete problem

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

is the ∞-Laplace equation

(1)
$$\begin{cases} \Delta_\infty u = 0, & \text{in } \Omega \setminus \Gamma \\ u = g, & \text{on } \Gamma \\ \dfrac{\partial u}{\partial n} = 0, & \text{on } \partial\Omega \setminus \Gamma. \end{cases}$$

Discrete to continuum for ∞-Laplacian

- (Oberman 2005) On a uniform grid with we have $u_n \to u$ uniformly if $\varepsilon \gg \delta_n$.
- (Smart 2010) On a uniform grid

$$\|u_n - u\|_\infty \leq C \sqrt[3]{\frac{\delta_n}{\varepsilon^2}} \quad \text{for } \delta_n^{1/2} \leq \varepsilon \leq \delta_n^{1/5}.$$

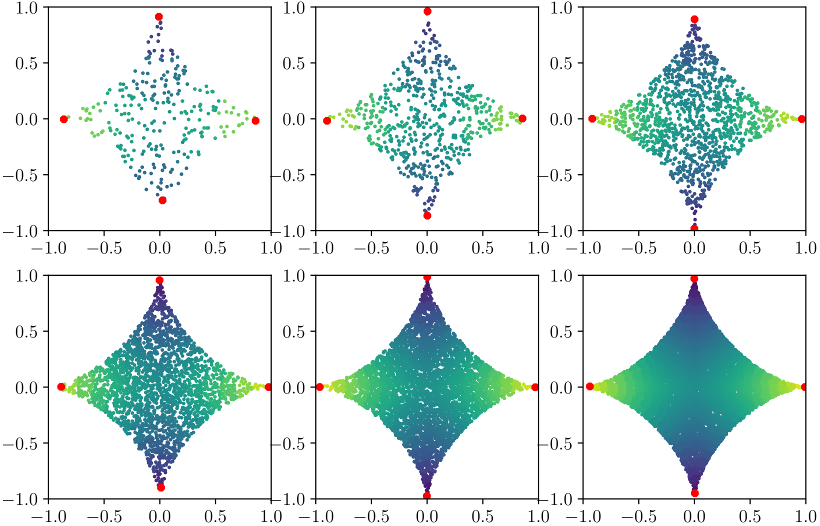
- (Calder 2019) On a random geometric graph (RGG) $w_{ij} = \eta(\varepsilon^{-1}|x_i - x_j|)$ on the Torus we have $u_n \to u$ provided $\varepsilon_n \gg \delta_n^{2/3}$.
- (Bungert & Roith 2022) Gamma convergence on RGG provided $\varepsilon_n \gg \delta_n$.
- (Bungert, Calder, & Roith, 2022a) On RGG we have

$$\|u_n - u\|_\infty \leq C \sqrt[4]{\frac{\delta_n}{\varepsilon}} \quad \text{for } \delta_n \ll \varepsilon \leq \delta_n^{5/9}.$$

- (Bungert, Calder, & Roith, 2022b) On uniform RGG with $\varepsilon \sim \delta_n$ we have

$$\|u_n - u\|_\infty \leq C\delta_n^{1/9}.$$

Numerical results



Numerical results

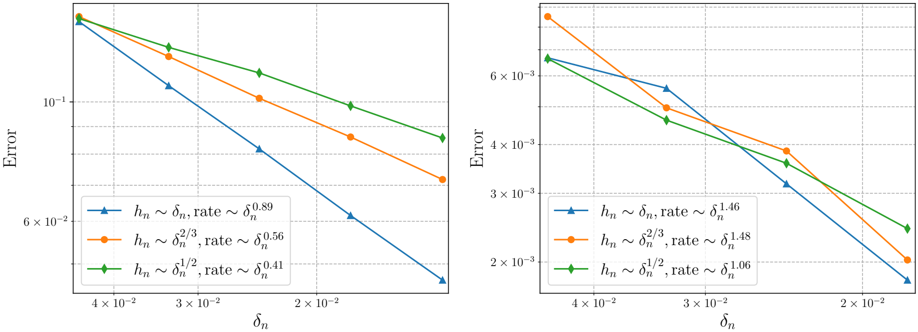


Figure: Empirical convergence rates for (left) unit weights and (right) singular weights.

Max Ball Theorem

For continuous $u : \mathbb{R}^d \to \mathbb{R}$ define

$$u^\varepsilon(x) = \max_{B(x,\varepsilon)} u \ \text{ and } \ u_\varepsilon(x) = \min_{B(x,\varepsilon)} u.$$

Define the nonlocal ∞-Laplacian

$$\Delta_\infty^\varepsilon u(x) = \left(\max_{B(x,\varepsilon)} + \min_{B(x,\varepsilon)} \right) u - 2u(x) = u^\varepsilon(x) + u_\varepsilon(x) - 2u(x).$$

Recall the ∞-Laplacian is defined as

$$\Delta_\infty u = \frac{\nabla u^T \nabla^2 u \nabla u}{|\nabla u|^2}.$$

Theorem (Smart 2010)

If $\Delta_\infty u = 0$ in the viscosity sense, then $\Delta_\infty^\varepsilon u_\varepsilon \leq 0$ and $\Delta_\infty^\varepsilon u^\varepsilon \geq 0$.

Max Ball Theorem

Theorem (Smart 2010)

If $\Delta_\infty u = 0$ *in the viscosity sense, then* $\Delta_\infty^\varepsilon u_\varepsilon \le 0$ *and* $\Delta_\infty^\varepsilon u^\varepsilon \ge 0$*.*

Proof.

1. Check that $\Delta_\infty |x| = 0$.
2. Use the comparison principle (comparison with cones) to obtain

$$u(y) \ge u(x) - \left(\frac{u(x) - u_{2\varepsilon}(x)}{2\varepsilon}\right)|y - x|, \quad y \in B(x, 2\varepsilon).$$

3. Minimize both sides over $y \in B(x,\varepsilon)$ (i.e., $|x-y| = \varepsilon$) to find that

$$u_\varepsilon \ge \frac{1}{2}(u + u_{2\varepsilon}).$$

4. Now compute

$$\Delta_\infty^\varepsilon u_\varepsilon(x) = \left(\max_{B(x,\varepsilon)} + \max_{B(x,\varepsilon)}\right) u_\varepsilon - 2u_\varepsilon(x) \le u(x) + u_{2\varepsilon}(x) - 2u_\varepsilon(x) \le 0.$$

□

Max Ball on Graph Functions

For $u_n : \mathcal{X}_n \to \mathbb{R}$ define

$$u_n^h(x) = \max_{\mathcal{X}_n \cap B(x,h)} u_n \quad \text{and} \quad u_{n,h}(x) = \min_{\mathcal{X}_n \cap B(x,h)} u_n.$$

Roughly speaking, we can show (using comparison against graph cones) that

$$u_n(y) \geq u_n(x) - \left(\frac{u_n(x) - u_{n,2h}(x)}{\min_{y \in \mathcal{X}_n \setminus B(x,2h)} d_n(x,y)} \right) d_n(x,y), \quad y \in \mathcal{X}_n \cap B(x,2h).$$

Minimize both sides over $y \in B(x,h)$ to obtain

$$u_{n,h}(x) \geq \frac{1}{2}(u_n(x) + u_{n,2h}) + s_n(x)(u_n - u_{n,2h}),$$

$$\text{where} \quad s_n(x) = \frac{1}{2} - \frac{\max_{x \in \mathcal{X}_n \cap B(x,h)} d_n(x,y)}{\min_{y \in \mathcal{X}_n \setminus B(x,2h)} d_n(x,y)}.$$

This yields

$$\boxed{\Delta_\infty^h u_{n,h} \leq C s_n h.}$$

Percolation Theory

First passage percolation theory studies asymptotics of distance functions on random irregular domains, like geometric graphs or lattices.

1. **Lattice Percolation:** Graph nodes are $\mathcal{X} = \varepsilon\mathbb{Z}^d$, edges between x and $x \pm \varepsilon e_i$ with i.i.d. random edge weights.
2. **Power Weighted Percolation:** Graph nodes are n i.i.d. random variables, and the graph is complete with edge weights

$$w_{xy} = |x-y|^\alpha \quad \text{for } \alpha > 1.$$

3. **Euclidean Percolation:** Graph nodes are n i.i.d. random variables, and edge weights are geometric

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

Auffinger, Antonio, Michael Damron, and Jack Hanson. 50 years of first-passage percolation. Vol. 68. American Mathematical Soc., 2017.

Ratio Convergence in Euclidean Percolation

Theorem (Bungert, Calder, Roith 2022)

Assume ρ *is uniform and* $\eta(t) = t^{-1}$. *Let* $x_0, x \in \mathbb{R}^d$ *and assume*

$$K\delta_n \leq \varepsilon \leq \frac{|x - x_0|}{2}.$$

Then there exist constants $C_1, C_2 > 0$ *which are independent of* x_0 *and* x *such that:*

1. **(Concentration)** *For all* $\lambda > 0$ *it holds that*

$$\mathbb{P}\left(|d_n(x_0, x) - \mathbb{E}\left[d_n(x_0, x)\right]| > \lambda K\delta_n \sqrt{\frac{|x - x_0|}{\varepsilon}}\right) \leq C_1 \exp(-C_2\lambda).$$

2. **(Ratio convergence in expectation)** *For* n *sufficiently large,* $x_0 = 0$, *and* $x \in \mathbb{R}^d$ *such that* $\varepsilon \leq |x|$ *it holds that*

$$\left|\frac{\mathbb{E}\left[d_n(0, x)\right]}{\mathbb{E}\left[d_n(0, 2x)\right]} - \frac{1}{2}\right| \leq C_1 \frac{\varepsilon}{|x|} + \frac{C_2 K\delta_n}{\sqrt{\varepsilon\,|x|}} \log(n^{1/d}\,|x|).$$

Ratio Convergence in Euclidean Percolation

Theorem (Bungert, Calder, Roith 2022)

Assume ρ is uniform and $\eta(t) = t^{-1}$. Let $x_0, x \in \mathbb{R}^d$ and assume $\varepsilon = K\delta_n$. Then up to log factors we have

1. **(Concentration)** *For all $\lambda > 0$ it holds that*

$$\mathbb{P}\left(\frac{|d_n(x_0,x) - \mathbb{E}\left[d_n(x_0,x)\right]|}{|x-x_0|} > \lambda K\sqrt{\frac{\delta_n}{|x-x_0|}}\right) \leq C_1 \exp(-C_2\lambda).$$

2. **(Ratio convergence in expectation)** *For n sufficiently large, $x_0 = 0$, and $x \in \mathbb{R}^d$ such that $K\delta_n \leq |x|$ it holds that*

$$\left|\frac{\mathbb{E}\left[d_n(0,x)\right]}{\mathbb{E}\left[d_n(0,2x)\right]} - \frac{1}{2}\right| \leq C_1 K\sqrt{\frac{\delta_n}{|x|}}.$$

Remark (Bungert, Calder, Roith 2022)

Compare this to the best known convergence rates to Euclidean distance

$$d_n(x,y) = |x-y| + O\left(\varepsilon + |x-y|\frac{\delta_n}{\varepsilon}\right).$$

Future work, papers, and code

Future Work:

1. Extension of percolation results to non-uniform point clouds.
2. Extension to general weights $\eta(\varepsilon^{-1}|x-y|)$.
3. Extension to other types of graph Laplacians (i.e., 2-Laplacian, or spectral convergence)

Papers:

Bungert, L., Calder, J., & Roith, T. (2022). **Uniform Convergence Rates for Lipschitz Learning on Graphs**. IMA Journal of Numerical Analysis.

Bungert, L., Calder, J., & Roith, T. (2022). **Ratio convergence rates for Euclidean first-passage percolation: Applications to the graph infinity Laplacian**. arXiv preprint arXiv:2210.09023.

Code:

- `https://github.com/jwcalder/LipschitzLearningRates`
- `https://github.com/TimRoith/PercolationConvergenceRates`