Graph-based Learning Beyond the Paradigm of Neural Networks

Binan Gu

Department of Mathematical Sciences, New Jersey Institute of Technology

New Jersey Institute of Technology
Fall 2020 Machine Learning Talk III

Motivation: Image Classification (Labelling)

Motivation: Image Classification (Labelling)

- 70, 000 grayscale 28×28 pixel handwritten digits $0 - 9$.

Motivation: Image Classification (Labelling)

- $70,000$ grayscale 28×28 pixel handwritten digits $0 - 9$.
- Construct k**-nearest neighbor graph** with weights of **Euclidean distance between images (an example).**

$$d_E^2(x, y) = \sum_{k=1}^{MN} (x_k - y_k)^2 , \quad x, y \in \mathbb{R}^{MN}$$

Motivation: Image Classification (Labelling)

- 70, 000 grayscale 28×28 pixel handwritten digits $0 - 9$.
- Construct k**-nearest neighbor graph** with weights of **Euclidean distance between images (an example).**

$$d_E^2(x, y) = \sum_{k=1}^{MN} (x_k - y_k)^2, \quad x, y \in \mathbb{R}^{MN}$$

- Minimize **graph energy** subject to constraints (training set data).

Motivation: Image Classification (Labelling)

- 70, 000 grayscale 28×28 pixel handwritten digits $0 - 9$.
- Construct k**-nearest neighbor graph** with weights of **Euclidean distance between images (an example).**

$$d_E^2(x, y) = \sum_{k=1}^{MN} (x_k - y_k)^2, \quad x, y \in \mathbb{R}^{MN}$$

- Minimize **graph energy** subject to constraints (training set data).

Conventional convolutional neural networks require Euclidean topology. The notion of distance in graphs can be abstract (manifolds).

Motivation: Image Classification (Labelling)

- $70,000$ grayscale 28×28 pixel handwritten digits $0 - 9$.
- Construct k**-nearest neighbor graph** with weights of **Euclidean distance between images (an example).**

$$d_E^2(x,y) = \sum_{k=1}^{MN} (x_k - y_k)^2, \quad x, y \in \mathbb{R}^{MN}$$

- Minimize **graph energy** (?) subject to constraints (training set data).

Conventional convolutional neural networks require Euclidean topology. The notion of distance in graphs can be abstract (manifolds).

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$.

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$.
$\mathcal{X}$: data, lives in $\mathbb{R}^d$.
$\mathcal{Y}$: labels (class), lives in $\mathbb{R}^k$, e.g., in the image classification problem, the label for an image showing digit 7 is the integer 7.

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$.
$\mathcal{X}$: data, lives in $\mathbb{R}^d$.
$\mathcal{Y}$: labels (class), lives in $\mathbb{R}^k$, e.g., in the image classification problem, the label for an image showing digit 7 is the integer 7.

Problem

Learn a labelling function $u : \mathcal{X} \to \mathcal{Y}$ given

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$.
$\mathcal{X}$: data, lives in $\mathbb{R}^d$.
$\mathcal{Y}$: labels (class), lives in $\mathbb{R}^k$, e.g., in the image classification problem, the label for an image showing digit 7 is the integer 7.

Problem

Learn a labelling function $u : \mathcal{X} \to \mathcal{Y}$ given

- all of the labels $\mathcal{Y}$: **fully-supervised**, i.e. least square regression. But labels are hard to obtain (sparseness).

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^{n} \in \mathcal{X} \times \mathcal{Y}$.
$\mathcal{X}$: data, lives in $\mathbb{R}^d$.
$\mathcal{Y}$: labels (class), lives in $\mathbb{R}^k$, e.g., in the image classification problem, the label for an image showing digit 7 is the integer 7.

Problem

Learn a labelling function $u : \mathcal{X} \to \mathcal{Y}$ given

- all of the labels $\mathcal{Y}$: **fully-supervised**, i.e. least square regression. But labels are hard to obtain (sparseness).
- none of the labels $\mathcal{Y}$: **unsupervised**.

Semi-Supervised Learning

Consider ordered pairs $\{(x_i, y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$.
$\mathcal{X}$: data, lives in $\mathbb{R}^d$.
$\mathcal{Y}$: labels (class), lives in $\mathbb{R}^k$, e.g., in the image classification problem, the label for an image showing digit 7 is the integer 7.

Problem

Learn a labelling function $u : \mathcal{X} \to \mathcal{Y}$ given

- all of the labels $\mathcal{Y}$: **fully-supervised**, i.e. least square regression. But labels are hard to obtain (sparseness).
- none of the labels $\mathcal{Y}$: **unsupervised**.

Ground in-between

Learn u given only labeled data $(x_1, y_1), \ldots, (x_m, y_m)$ where $m \ll n$: **semi-supervised learning.**
Various sample size limits to the two extreme modes.

Graph Energy Minimization

Smoothness assumption for semi-supervised learning.

Graph Energy Minimization

Smoothness assumption for semi-supervised learning.

Graph construction

Recall $\mathcal{X}$ is data space. Construct $G = (\mathcal{X}, \mathcal{W})$ with weight $W = \left(w_{xy}\right)_{x,y\in\mathcal{X}}$ encoding *similarity* between data.

Graph Energy Minimization

Smoothness assumption for semi-supervised learning.

Graph construction

Recall $\mathcal{X}$ is data space. Construct $G = (\mathcal{X}, \mathcal{W})$ with weight $W = \left(w_{xy}\right)_{x,y\in\mathcal{X}}$ encoding *similarity* between data.

Recast as an optimization problem

For energy $\mathcal{E}(u)$ and a labeling function $u(x) = \left(u_i(x)\right)_{i=1}^{k}$

$$\begin{cases} \text{Minimise } \mathcal{E}(u) \text{ over } u : \mathcal{X} \to \mathbb{R}^k & \text{smoothness of } u \\ \text{subject to } u = g : \Gamma \subset \mathcal{X} \to \mathcal{Y} \text{ on } \Gamma & \text{given labeled data} \end{cases}$$

Graph Energy Minimization

Smoothness assumption for semi-supervised learning.

Graph construction

Recall $\mathcal{X}$ is data space. Construct $G = (\mathcal{X}, \mathcal{W})$ with weight $W = \left(w_{xy}\right)_{x,y \in \mathcal{X}}$ encoding *similarity* between data.

Recast as an optimization problem

For energy $\mathcal{E}(u)$ and a labeling function $u(x) = \left(u_i(x)\right)_{i=1}^{k}$

$$\begin{cases} \text{Minimise } \mathcal{E}(u) \text{ over } u : \mathcal{X} \to \mathbb{R}^k & \text{smoothness of } u \\ \text{subject to } u = g : \Gamma \subset \mathcal{X} \to \mathcal{Y} \text{ on } \Gamma & \text{given labeled data} \end{cases}$$

where examples of graph energy $\mathcal{E}(u)$ are Dirichlet energy

$$\mathcal{E}(u) = \frac{1}{2} \sum_{x,y \in \mathcal{X}} w_{xy} \left| u(y) - u(x) \right|^2 = \frac{1}{2} u^T \underbrace{L_w}_{\text{graph Laplacian}} u$$

Calculus on Graphs

We need proper notions of

- inner product

$$(u, v)_{l^2(\mathcal{X})} = \sum_{x \in \mathcal{X}} u(x)\, v(x)\,;$$

Calculus on Graphs

We need proper notions of

- inner product

$$(u, v)_{l^2(\mathcal{X})} = \sum_{x \in \mathcal{X}} u(x)\, v(x)\, ;$$

- derivatives/gradient,

$$\nabla u(x, y) = u(x) - u(y)\, ;$$

Calculus on Graphs

We need proper notions of

- inner product

$$(u, v)_{l^2(\mathcal{X})} = \sum_{x \in \mathcal{X}} u(x)\, v(x)\,;$$

- derivatives/gradient,

$$\nabla u(x, y) = u(x) - u(y)\,;$$

- vector fields (on edges), antisymmetric

$$V : \mathcal{X}^2 \to \mathbb{R}, \quad V(x, y) = -V(y, x)\,;$$

Calculus on Graphs

We need proper notions of

- inner product

$$(u, v)_{l^2(\mathcal{X})} = \sum_{x \in \mathcal{X}} u(x)\, v(x)\,;$$

- derivatives/gradient,

$$\nabla u(x, y) = u(x) - u(y)\,;$$

- vector fields (on edges), antisymmetric

$$V : \mathcal{X}^2 \to \mathbb{R}, \quad V(x, y) = -V(y, x)\,;$$

- divergence (to satisfy discrete divergence theorem)

$$\operatorname{div} V(x) = \sum_{y \in \mathcal{X}} w_{xy}\, V(x, y)\,,$$

and classical theoretical tools

- maximum principles for Laplacian regularized minimization;

Calculus on Graphs

We need proper notions of

- inner product

$$(u, v)_{l^2(\mathcal{X})} = \sum_{x \in \mathcal{X}} u(x)\, v(x)\,;$$

- derivatives/gradient,

$$\nabla u(x, y) = u(x) - u(y)\,;$$

- vector fields (on edges), antisymmetric

$$V : \mathcal{X}^2 \to \mathbb{R}, \quad V(x, y) = -V(y, x)\,;$$

- divergence (to satisfy discrete divergence theorem)

$$\mathrm{div}\, V(x) = \sum_{y \in \mathcal{X}} w_{xy} V(x, y)\,,$$

and classical theoretical tools

- maximum principles for Laplacian regularized minimization;
- you name it ...

More on Random Graphs: Concentration of Measure

It is wise to learn how "well-behaved" a random graph generated by sampled data is.

More on Random Graphs: Concentration of Measure

It is wise to learn how “well-behaved” a random graph generated by sampled data is.

More on Random Graphs: Concentration of Measure

It is wise to learn how "well-behaved" a random graph generated by sampled data is.

Chebyshev's overkilling condition and bad tails

For i.i.d random variable X_i with finite common mean μ and variance σ^2,

$$S_n = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \mathbb{P}\left(|S_n - \mu_X| \geq t\right) \leq \frac{\sigma^2}{nt^2}.$$

More on Random Graphs: Concentration of Measure

It is wise to learn how "well-behaved" a random graph generated by sampled data is.

Chebyshev's overkilling condition and bad tails

For i.i.d random variable X_i with finite common mean μ and variance σ^2,

$$S_n = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad \mathbb{P}\left(|S_n - \mu_X| \geq t\right) \leq \frac{\sigma^2}{nt^2}.$$

Hoeffding Inequality

For i.i.d random variables X_i with finite mean μ such that $|X - \mu| \leq b$ for some positive b,

$$\mathbb{P}\left(|S_n - \mu_X| \geq t\right) \leq 2\exp\left(-\frac{nt^2}{2b^2}\right).$$

Passing to the (mean-field) limit

From discrete to continuous

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u$$

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u \stackrel{?}{\to}$$

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u \xrightarrow{?} \frac{1}{2} \int_{\mathcal{X}} |\nabla u(x)|^2 \rho^2(x)\, dx = \mathcal{E}(u)$$

where $C_{\epsilon,n,w}$ is a proper normalizing constant.

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u \overset{?}{\to} \frac{1}{2}\int_{\mathcal{X}} |\nabla u(x)|^2 \rho^2(x)\, dx = \mathcal{E}(u)$$

where $C_{\epsilon,n,w}$ is a proper normalizing constant.

- Solving discrete graph energy minimisation is expensive when data set becomes uncontrollably large.

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u \xrightarrow{?} \frac{1}{2}\int_{\mathcal{X}} |\nabla u(x)|^2 \rho^2(x)\, dx = \mathcal{E}(u)$$

where $C_{\epsilon,n,w}$ is a proper normalizing constant.

- Solving discrete graph energy minimisation is expensive when data set becomes uncontrollably large.
- With probabilistic tools, we can make almost sure statements about the convergence of the discrete energies to continuous (nonlocal) energies.

Passing to the (mean-field) limit

From discrete to continuous

$$\mathcal{E}_{\epsilon,n,w}(u) = \frac{C_{\epsilon,n,w}}{2} u^T L_w u \overset{?}{\to} \frac{1}{2} \int_{\mathcal{X}} |\nabla u(x)|^2 \rho^2(x)\, dx = \mathcal{E}(u)$$

where $C_{\epsilon,n,w}$ is a proper normalizing constant.

- Solving discrete graph energy minimisation is expensive when data set becomes uncontrollably large.
- With probabilistic tools, we can make almost sure statements about the convergence of the discrete energies to continuous (nonlocal) energies.
- “It is an interesting, and somewhat open, problem to determine the fewest number of labeled points for which discrete to continuum convergence holds.” - Jeff Calder [1]

A Transportation Point of View

Map from empirical distribution (discreteness of data), via partition of space and an extension operator, to a continuum integral counterpart.

A Transportation Point of View

Map from empirical distribution (discreteness of data), via partition of space and an extension operator, to a continuum integral counterpart.

Extension operator

$X_1, \ldots, X_n$ i.i.d with density ρ on U. There exists a partition (a.s.) $\{U_i\}$ (a cover) of those data, a corresponding density ρ_δ such that $\rho_\delta(U_i) = \frac{1}{n}$ and an extension operator E_δ such that

$$E_\delta u(x) = \sum_{i=1}^{n} u(X_i) \mathbf{1}_{U_i}(x),$$

A Transportation Point of View

Map from empirical distribution (discreteness of data), via partition of space and an extension operator, to a continuum integral counterpart.

Extension operator

$X_1, \ldots, X_n$ i.i.d with density ρ on U. There exists a partition (a.s.) $\{U_i\}$ (a cover) of those data, a corresponding density ρ_δ such that $\rho_\delta(U_i) = \frac{1}{n}$ and an extension operator E_δ such that

$$E_\delta u(x) = \sum_{i=1}^{n} u(X_i) \mathbf{1}_{U_i}(x),$$

Transportation Map T_δ

Define $T_\delta(x) = X_i$ iff $x \in U_i$. Then $E_\delta u = u \circ T_\delta$. If one considers an empirical measure μ_n on $A \subset U$, then T_δ pushes forward ρ_δ to μ_n.

Take-aways

- Learn PDE
- Learn probability theory
- Learn calculus of variations if you want to prove convergence of new methods
- Graph-based methods don't restrict on underlying topology.
- Graph-based semi-supervised learning is just a mask of all of the above combined.

References

- J. Calder. The Calculus of Variations. *Ch.5.* University of Minnesota 2020.
- O. Chapelle, B. Schölkopf, A. Zien. Semi-Supervised Learning. The MIT Press 2010.
- L. Wang, Y. Zhang, J. Feng. On the Euclidean Distance of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1334-1339, 2005.
- A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Comput. Surv. 31, 3, 264–323, 1999.
- Y. Lecun, L. Bottou, Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.