Mathematics of Image and Data Analysis
Math 5467

Lecture 20: Graph-based semi-supervised learning

Instructor: Jeff Calder
Email: jcalder@umn.edu

http://www-users.math.umn.edu/~jwcalder/5467S21

Last time

- Intro to Machine Learning

Today

- Graph-based semi-supervised learning.

Graph

Recall: Semi-supervised learning uses both labeled and unlabeled data to learn.

One way to use the unlabeled data is to build a graph, which is encoded into a weight matrix W.

- $W(i,j)$ is the similarity between i and j ($W(i,j) \geq 0$).
- We assume W is symmetric $W = W^T$.
- Often can choose Gaussian weights (recall spectral clustering)

$$W(i,j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}.$$

Some datasets already have graph structure (citation databases, network problems, etc.).

Example graph

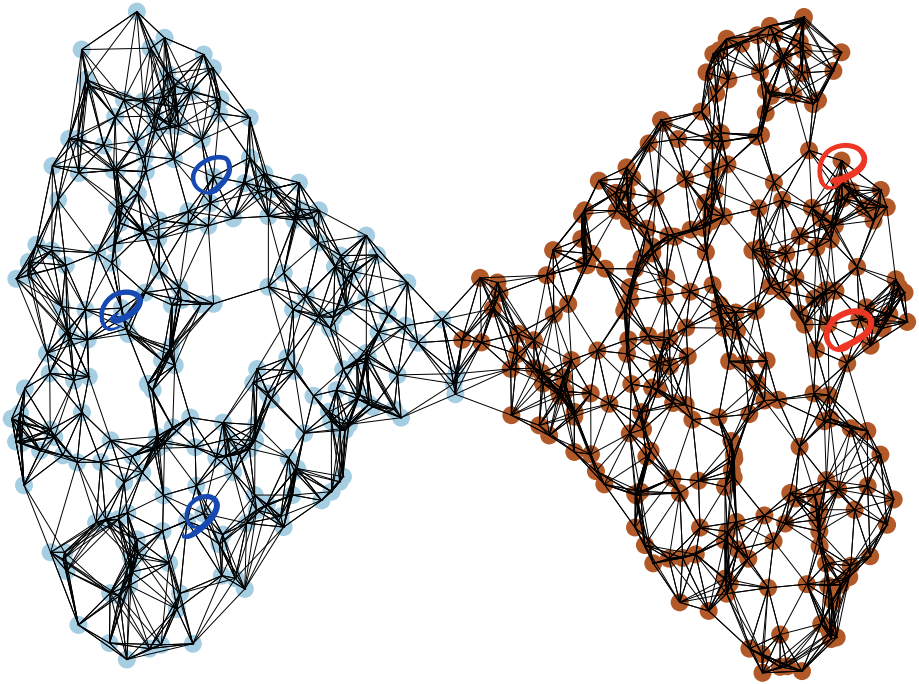


Figure 1: An example of a k-nearest neighbor graph.

Graph-based semi-supervised learning

Let $I_m = \{1, 2, \ldots, m\}$ denote the indices of all our datapoints.

We asssume there is a subset of the nodes $\Gamma \subset I_m$ that are asssigned label vectors from the one-hot vectors

$$S_k = \{e_1, e_2, \ldots, e_k\}.$$

i^{th} class $\longrightarrow$ $e_i = (0, 0, \cdots, 0, 1, 0, \cdots, 0)$ (i^{th} spot.)

We can treat the labels as a function $g : \Gamma \to S_k$, where $g(i)$ is the label of node $i \in \Gamma$.

Task: Extend labels from the subset Γ to the rest of the graph is a meaningful way.

Laplacian regularization

It is common in practice to take the *semi-supervised smoothness assumption*, which stipulates that the learned labels should vary as smoothly as possible, and in particular, should not change rapidly within high density regions of the graph, which are likely to be clusters with the same label.

Laplacian regularized learning imposes the semi-supervised smoothness assumption by minimizing the function

Graph cut

$$E(u) = \frac{1}{4}\sum_{i=1}^{m}\sum_{j=1}^{m} W(i,j)\|u(i) - u(j)\|^2 \tag{1}$$

over labeling functions $u : I_m \to \mathbb{R}^k$, subject to $u = g$ on Γ, that is, that the known labels are correct.

$$\text{Label}(i) = \underset{1 \le j \le k}{\arg\max} \{u(i) \cdot e_j\}$$

Gradient descent

To minimize E we use gradient descent. Define the inner product for $u, v : I_m \to \mathbb{R}^k$ by

$$\langle u, v \rangle_H = \sum_{i=1}^{m} d(i) u(i)^T v(i), \tag{2}$$

where $d : I_m \to \mathbb{R}$ are the degrees, given by $d(i) = \sum_{j=1}^{m} W(i,j)$. The induced norm is

$$\|u\|_H^2 = \langle u, u \rangle = \sum_{i=1}^{m} d(i) \|u(i)\|^2.$$

Euclidean norm

We claim that $\nabla E(u) = d^{-1} L u$.

$$Lu(i) = \sum_{j=1}^{m} w(i,j)(u(i) - u(j))$$

Gateaux Derivative

$$\frac{d}{dt}\Big|_{t=0} E(u + tv) = \langle \nabla E(u), v \rangle_H$$

Def of ∇E

$$E(u+tv) = \frac{1}{4}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,\|u(i)-u(j)+t(v(i)-v(j))\|^2$$

$\|a+b\|^2 = \|a\|^2 + \|b\|^2 + 2a^Tb$

$$= \cancel{\frac{1}{4}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,\|u(i)-u(j)\|^2} \quad \frac{d}{dt}\Big|_{t=0} = 0$$

$$+ \cancel{\frac{t^2}{4}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,\|v(i)-v(j)\|^2}$$

$$+ \frac{t}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T(v(i)-v(j))$$

$$\frac{d}{dt}\Big|_{t=0} E(u+tv) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T(v(i)-v(j))$$

$$= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T v(i)$$

$$- \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T v(j)$$

swap i and j

$$= -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T v(i)$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} w(i,j)\,(u(i)-u(j))^T v(i)$$

$$= \sum_{i=1}^{m} (Lu(i))^T v(i)$$

$$= \sum_{i=1}^{m} d(i) \left(d(i)^{-1} L u(i) \right)^T v(i)$$

$$= \langle d^{-1} L u, v \rangle_H$$

$$\Longrightarrow \boxed{\nabla E(u) = d^{-1} L u.}$$

Gradient descent

Using gradient descent to minimize E amounts to the iteration

$$u_{k+1} = u_k - dt \nabla E(u_k). \tag{3}$$

$u_{k+1} = 3$ on T

$\nabla E = d^{-1} L u.$

Lemma 1. *If $0 < dt \leq 1$ then for all $k \geq 1$ and $1 \leq i \leq m$ we have*

$$\|u_k(i)\| \leq \max_{1 \leq i \leq m} \|u_0(i)\|. \tag{4}$$

Proof: $u_{k+1}(i) = u_k(i) - dt\, d(i)^{-1} L u_k(i)$

$d(i) = \sum_{j=1}^{m} w(i,j)$

$$u_{k+1}(i) = u_k(i) - dt\, d(i)^{-1} \sum_{j=1}^{m} w(i,j)\,(u_k(i) - u_k(j))$$

$$= u_k(i) - dt\, d(i)^{-1} \sum_{j=1}^{m} w(i,j)\, u_k(i) + dt\, d(i)^{-1} \sum_{j=1}^{m} w(i,j)\, u_k(j)$$

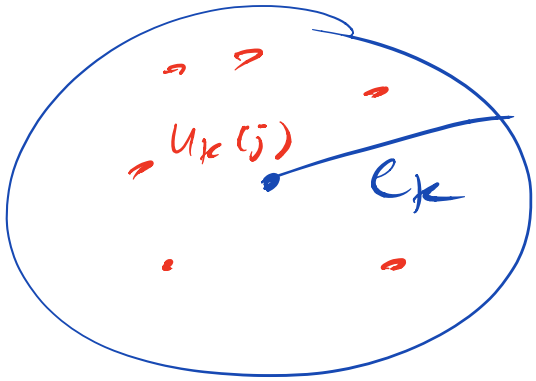
$$= \left(1 - dt\right) u_k(i) + \sum_{j=1}^{m} \left(\frac{dt\, w(i,j)}{d(i)}\right) u_k(j)$$

↑ sum to dt

If $dt \le 1$, then

$u_{k+1}(i)$ is a weighted average of $u_k(j)$ for $j = 1, \ldots, m$

Let $C_k = \max_{1 \le j \le m} \|u_k(j)\|$



Ball convex

$\Rightarrow \|u_{k+1}(i)\| \le C_k$

or $C_{k+1} \le C_k \le C_{k-1} \cdots \le C_0$.

Convergence?

We may wish to go beyond stability and instead prove convergence of the iterations as $k \to \infty$ to a solution of the equation $\nabla E = 0$, that is

$$\begin{cases} Lu(i) = 0, & \text{if } i \in I_m \setminus \Gamma \\ u(i) = g(i), & \text{otherwise.} \end{cases} \tag{5}$$

- Depends on whether the graph is connected.
- If the graph is connected, (5) has a unique solution and gradient descent converges.
- If the graph is not connected, then (5) can have multiple solutions, and gradient descent will converge to one, but which one is dependent on initialization.

Clasification of MNIST digits

We use a k-nearest neighbor graph with $k = 10$ and weights given by

$$W(i,j) = \exp\left(-\frac{4\|x_i - x_j\|^2}{d_k(x_i)^2}\right),$$

where $d_k(x_i)$ is the distance to the k^{th} nearest neighbor. The matrix is then symmetrized $W = W + W^T$.

10	**20**	**40**	**80**	**160**
85.4 (4.4)	91.7 (1.2)	93.4 (0.5)	94.3 (0.3)	94.8 (0.1)

Table 1: Laplace learning on MNIST with 10, 20, 40, 80, and 160 labels per class. The average (standard deviation) classification accuracy over 100 trials is shown.

GraphLearning (.ipynb)