Label Complexity of Graph-Based Semi-Supervised Learning

Salman Avestimehr

In collaboration with Aamir Anis, Aly El Gamal, and Antonio Ortega

ITA 2/2/15
Semi-Supervised Learning

- \( n \) sample points from feature space (\( \mathbb{R}^d \)) according to \( p(x) \)
- The samples belong to two (or more) classes
- \( \ell \) sampled points are labeled (expert feedback)
  - Labels determine class membership
  - Typically \( \ell/n \) is small
- Objective: use both geometry of sample points and labels knowledge to construct a classifier for the rest of data
Graph Representation for SSL

- Representing the geometry in feature space with a graph
  - Vertices represent the sample points
  - Weighted edges represent the geometry or similarity between samples (e.g., distance based Gaussian Kernel)
  - Indicator function, $f = 1_S$, represents the classifier (assuming two classes)
- Note: labels are less likely to change across heavy edges, so the classifier $f = 1_S$ is expected to have “low bandwidth”
- Objective: use the graph and labels knowledge to find a “low bandwidth” classifier $f = 1_S$

```
\delta_S
```

```
f = 1_S
```

```
w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{||X_i - X_j||^2}{2\sigma^2}\right)
```
In this talk …

(Q1) How many labels should we get (label complexity)?
(Q2) Which vertices should we choose for labeling?
(Q3) How to construct the classifier, \( f = 1_S \), to label other vertices?

\[
W_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp \left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right)
\]
Problem Formulation

- \( n \) sample points from feature space \((\mathbb{R}^d)\) according to \( p(x) \)
- Sample points belong to either class 1 or class 2, which are separated by a smooth hyper surface \( \delta_S \) (with radius of curvature > \( \tau \))
- The graph is constructed according to distance based Gaussian Kernel
- The classifier corresponding to \( \delta_S \) is represented by \( f=1_S \)

(Q1) How many labels are needed to construct \( f=1_S \) (label complexity)?
(Q2) Which vertices should we choose for labeling?
(Q3) How to construct the classifier, \( f=1_S \), to label other vertices?

\[
\delta_S
\]

\[
f = 1_S
\]

Distance-Based Gaussian Kernel:
\[
w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp \left( -\frac{\|X_i - X_j\|^2}{2\sigma^2} \right)
\]
Main Result

- The fraction of labels needed for constructing the classifier $f=1_S$ is “asymptotically” given by
  
  $$\int p(x) < \sup_{s \in \partial S} p(s) \, dx$$

**Graph Representation for SSL: Geometry Information**

- Vertices represent the sampled points
- Edges represent similarity between the samples

**Distance-Based Gaussian Kernel**

$$w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp \left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right)$$

*Kernel Bandwidth* determines volume of neighborhood
Main Result

- The fraction of labels needed for constructing the classifier $f=1_S$ is “asymptotically” given by

$$\int p(x) < \sup_{s \in \partial S} p(s)$$

- The result demonstrates how label complexity depends on $p(x)$ and $\delta_S$ (e.g., low density boundaries are easier to learn)
Main Result

- The fraction of labels needed for constructing the classifier $f=1_S$ is “asymptotically” given by

$$\int p(x) dx < \sup_{s \in \partial S} p(s)$$

- The result demonstrates how label complexity depends on $p(x)$ and $\delta_S$ (e.g., low density boundaries are easier to learn).
- We also demonstrate how to achieve the bound (i.e., which vertices to choose for labeling and how to construct $f=1_S$).
How to prove the result?

- How many labels are needed to construct \( f = 1_S \)?

- Use graph sampling theory

\[
\delta_S
\]

Distance-Based Gaussian Kernel:

\[
w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right)
\]
Graph Sampling Theory

- Consider a weighted undirected graph with signal $f$

- Graph Laplacian:

  $L = \frac{1}{n} (D - W)$

  where $D$ is a diagonal degree matrix and $W$ is the adjacency matrix.

- Bandwidth of graph signal:

  $\omega(f) = \max \lambda_i : u_i^T f \neq 0$

  where $\lambda_i$ is the $i$-th smallest eigenvalue of $L$ with eigenvector $u_i$.

- Theorem (Pesenson, 2008):

  # labels needed to construct $f = \max i : \lambda_i \leq w(f)$

  # Laplacian eigenvalues below $w(f)$
Two challenges …

- Using the graph sampling theorem we have

\[
\text{# labels needed to construct } f = \max_i : \lambda_i \leq w(f)
\]

- To characterize label complexity we need to
  1. Find the bandwidth of \( f=1_S \) (corresponding to the underlying classification boundary \( \delta_S \))?
  2. Find the eigenvalue distribution of graph Laplacian?

\[
\delta_S
\]
How to find the bandwidth of classifier?

- **Bandwidth Approximation Theorem (Anis-Gadde-Ortega, 2014):**

  \[
  w(f) = \lim_{m \to \infty} w_m(f) = \lim_{m \to \infty} \sqrt{m \frac{f^T L^m f}{f^T f}}
  \]

- So, we need to analyze

  \[
  w_m(1_S) = \sqrt{m \frac{1^T_S L^m 1_S}{1^T_S 1_S}}
  \]
Special case (m=1)

Theorem [Maier-Luxburg-Hein]

Under the conditions $\sigma \to 0$ and $n\sigma^{d+1} \to \infty$,

$$\frac{\sqrt{2\pi}}{n\sigma} L_1^T L_1 S \xrightarrow{p.} \int_{\partial S} p^2(s) ds,$$

where $ds$ ranges over all $(d-1)$-dimensional volume elements tangent to the hyperplane $\partial S$.

- Note that: $L_1^T L_1 S = \sum_{i \in S, j \in S^c} w_{ij} = \text{cut}(S, S^c)$

$\delta_S$ minimum cut graph clustering results in a low density separation

$$w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{||X_i - X_j||^2}{2\sigma^2}\right)$$
Generalization to $m>1$

**Theorem**

If conditions 1–4 hold, then

$$m \sqrt{\frac{1^T S L^m 1_S}{1^T S 1_S}} = \omega_m(1_S) \xrightarrow{p.} \sup_{s \in \partial S} p(s),$$

Further, almost sure convergence holds if

$$\frac{n \sigma^{md+1}}{m C^m \log n} \to \infty.$$

1. Large sample size: $n \to \infty$,
2. Shrinking neighborhood volume: $\sigma \to 0$,
3. Large bandwidth estimate: $m \to \infty$,
4. Scaling laws: $m/n \to 0$, $m \sigma^2 \to 0$, $(1/\sigma)^{1/m} \to 1$, $(n \sigma^{md+1})/(m C^m) \to \infty$, where $C = 2/(2\pi)^{d/2}$.
Generalization to $m > 1$

**Theorem**

*If conditions 1–4 hold, then*

\[
\sqrt{\frac{1_T}{1_S} \frac{L^m 1_S}{1_T 1_S}} = \omega_m(1_S) \xrightarrow{p} \sup_{s \in \partial S} p(s),
\]

*Further, almost sure convergence holds if*

\[
\frac{n \sigma^{md+1}}{m C m \log n} \to \infty.
\]

1. Large sample size: $n \to \infty$,
2. Shrinking neighborhood volume: $\sigma \to 0$,
3. Large bandwidth estimate: $m \to \infty$,
4. Scaling laws: $m/n \to 0$, $m \sigma^2 \to 0$, $(1/\sigma)^{1/m} \to 1$, $(n \sigma^{md+1})/(m C^m) \to \infty$,
   where $C = 2/(2\pi)^{d/2}$.

**Proof based on**

a) Variance-bias decomposition of $w_m(1_S)$

b) Use results on concentration of U statistics ("Probability inequalities for sums of bounded random variables", by W. Hoeffding)

c) Establish convergence of the bias term
Two challenges …

- Using the graph sampling theorem we have

$$\text{# labels needed to reconstruct } f = \max i : \lambda_i \leq w(f)$$

- To characterize label complexity we need to

  ① Find the bandwidth of $f=1_S$ (corresponding to the underlying classification boundary $\delta_S$)

  ② Find the eigenvalue distribution of graph Laplacian?

---

**Semi-Supervised Learning**

- $n$ sampled points from feature space $\mathbb{R}^d$ according to $p(x)$

- $l$ sampled points are labelled (expert feedback), $l \ll n$

- Labels determine class membership

**Goal:** Find smooth classification boundary $f = 1_S$
Convergence of Eigenvalue Distribution

\( q_n(x) \): Density of eigenvalues of \( L \) equal to \( x \)
\( Q_n(t) = \int_{-\infty}^{t} q(x) \, dx \): Fraction of eigenvalues below \( t \)

**Theorem**

*In the large sample and shrinking kernel bandwidth regime* \( n \to \infty, \sigma \to 0, \)
\[
\mathbb{E} \{ Q_n(t) \} \to \int_{p(x)<t} p(x) \, dx
\]

Proof similar to the proof of Wigner’s semi circle law.

\[
w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right)
\]
Main Result

$F_{m,S}$: fraction of labels required to reconstruct any function with bandwidth $\omega_m(1_S)$

Theorem

If conditions 1–4 hold, then

$$
\lim \mathbb{E} \{F_{m,S}\} \leq \int_{p(x) < \sup_{s \in \partial S} p(s)} p(x) dx,
$$

1. Large sample size: $n \to \infty$,
2. Shrinking neighborhood volume: $\sigma \to 0$,
3. Large bandwidth estimate: $m \to \infty$,
4. Scaling laws: $m/n \to 0$, $m\sigma^2 \to 0$, $(1/\sigma)^{1/m} \to 1$, $(n\sigma^{md+1})/(mC^m) \to \infty$, where $C = 2/(2\pi)^{d/2}$. 
**Problem Formulation**

- \( n \) sample points from feature space \((\mathbb{R}^d)\) according to \( p(x) \)
- Sample points belong to either class 1 or class 2, which are separated by a smooth hyper surface \( \delta_S \) (with radius of curvature \( > \tau \))
- The classifier corresponding to \( \delta_S \) is represented by \( f=1_S \)

(Q1) How many labels are needed to construct \( f=1_S \) (label complexity)?
(Q2) Which vertices should we choose for labeling?
(Q3) How to construct the classifier, \( f=1_S \), to label other vertices?

\[
w_{ij} = K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp \left( -\frac{\|X_i - X_j\|^2}{2\sigma^2} \right)
\]
Which vertices to choose for labeling?

- From label complexity, we know the average number of vertices that should be chosen to construct \( f=1_S \).
- But, which vertices should we choose for labeling?
  - For each subset of nodes, there exists a cut-off frequency that represents the maximum bandwidth of graph signals that can be reconstructed from them.
  - We should choose the subset of vertices whose cut-off frequency is above the bandwidth of \( f=1_S \).
- How to find those vertices efficiently?

\[
\delta_S
\]
Problem Formulation

- \( n \) sample points from feature space \((\mathbb{R}^d)\) according to \( p(x) \)
- Sample points belong to either class 1 or class 2, which are separated by a smooth hyper surface \( \delta_S \) (with radius of curvature > \( \tau \))
- The classifier corresponding to \( \delta_S \) is represented by \( f = 1_S \)

(Q1) How many labels are needed to construct \( f = 1_S \) (label complexity)?
(Q2) Which vertices should we choose for labeling?
(Q3) How to construct the classifier, \( f = 1_S \), to label other vertices?

\[
K_{\sigma^2}(X_i, X_j) = \frac{1}{(2\pi\sigma^2)^d/2} \exp \left( -\frac{||X_i - X_j||^2}{2\sigma^2} \right)
\]
How to construct the classifier?

- We can simply find the graph signal with **lowest bandwidth** that matches the labeled samples
  - If the number of labeled samples is above label complexity, this algorithm is guaranteed to exactly construct \( f = 1_S \)
  - Recall that \( w_m(1_S) \xrightarrow{p} \sup_{s \in \delta S} p(s) \), so this algorithm essentially results in a low density separation
  - In general, this algorithm minimizes signal variations across heavy edges with cost term \( m \sqrt{f^T L_m f} \). (connected to other works in the literature, e.g. [*], where Laplacian is used for regularization)

\[ \delta S \]

\[ f = 1_S \]

[*] M. Belkin, P. Niyogi, V. Sindhwani “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples”.
Summary and Concluding Remarks

• We characterized “label complexity” for graph based semi-supervised learning, by
  a) assuming there is an underlying smooth partition for data
  b) determining the # labeled samples needed to construct that classifier

• We also proposed a new approach for
  a) selecting samples for labeling (choosing vertices with highest cut-off frequency)
  b) classifying unlabeled data (finding the classifier with lowest bandwidth that matches the labeled data)

• A future direction:
  ❖ In general, there is no smooth boundary that perfectly classifies data. How to characterize the tradeoff between classification error and label complexity?
Questions?

References: