Semi-supervised Learning on Riemannian Manifolds

Sumit Shekhar

ENEE698: Clustering Seminar

Outline

- Introduction
- Review
  - Data Manifold Representation
  - Graph Equivalence
- Classification
- Proposed Algorithm
- Experiments
- Further usage
  - Clustering

Introduction

Semi-supervised Learning

- Learning to classify using labeled and unlabeled data
- For two class problem, C_1 vs C_2 on input space $\mathcal{X}$, learn
  - probability density, $p(x)$ on $\mathcal{X}$
  - class densities $\{p(C_1|x) \in \mathcal{X}\}$ and $\{p(C_2|x) \in \mathcal{X}\}$

Introduction

Semi-supervised Learning

- Learning to classify using labeled and unlabeled data
- For two class problem, C_1 vs C_2 on input space $\mathcal{X}$, learn
  - probability density, $p(x)$ on $\mathcal{X}$
  - class densities $\{p(C_1|x) \in \mathcal{X}\}$ and $\{p(C_2|x) \in \mathcal{X}\}$

Unlabeled data learns $p(x)$, labeled data to learn class-conditionals.

Introduction

Problem

- Consider case where data lies on a low-dimensional manifold
  - $p(x)$ puts all its measure on a compact manifold in $\mathbb{R}^n$
- Unlabeled data for learning manifold
- Labeled data to specify the classifier

Introduction

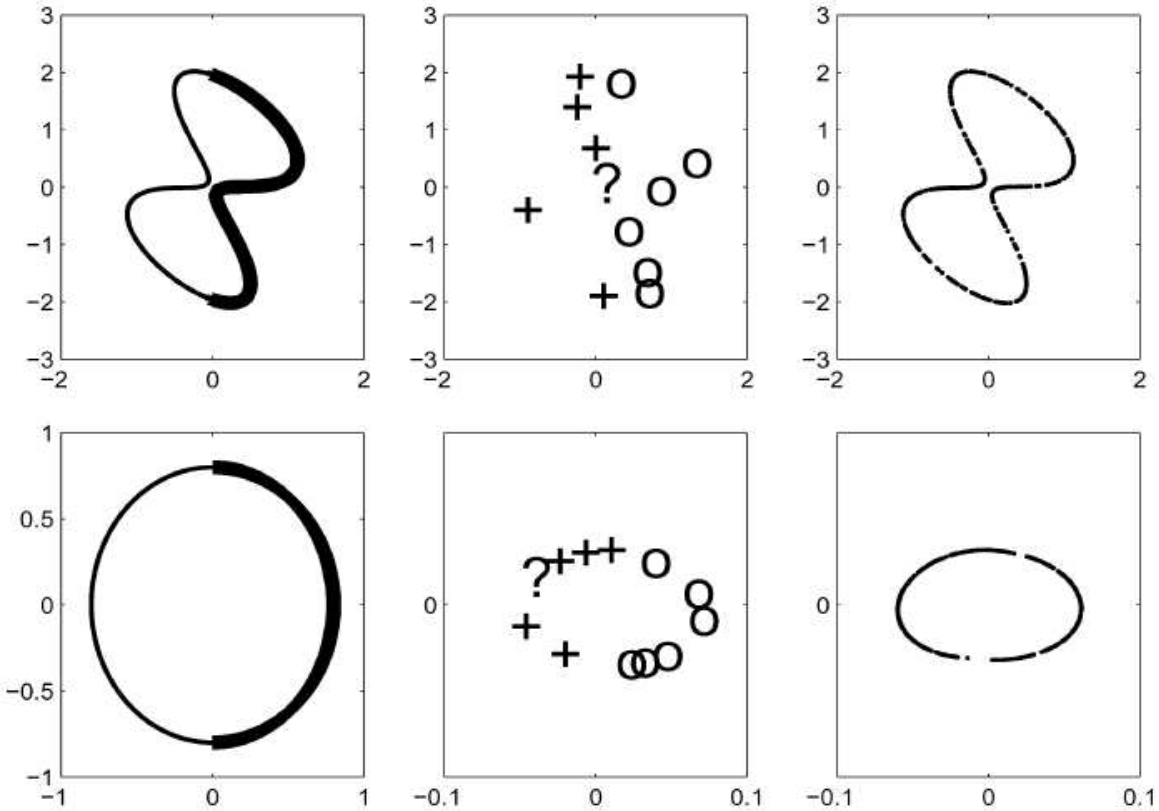


Figure 1. Top row: Panel 1. Two classes on a plane curve. Panel 2. Labeled examples. "?" is a point to be classified. Panel 3. 500 random unlabeled examples. Bottom row: Panel 4. Ideal representation of the curve. Panel 5. Positions of labeled points and "?" after applying eigenfunctions of the Laplacian. Panel 6. Positions of all examples.

Introduction

Requirements of representation

- Both labeled and unlabeled data required for classification.
- Out of possible representations of manifold, the one with slowest possible varying coordinates preferred.
- Classifier better learnt using geodesic distances.
- To avoid "curse of dimensionality", learn classifier as $f : \mathcal{M} \rightarrow Y$ on the low-dimensional manifold.

Introduction

Requirements of representation

- Both labeled and unlabeled data required for classification.
- Out of possible representations of manifold, the one with slowest possible varying coordinates preferred.
- Classifier better learnt using geodesic distances.
- To avoid "curse of dimensionality", learn classifier as $f : \mathcal{M} \rightarrow Y$ on the low-dimensional manifold.

Proposed Solution:

Use Laplacian Eigenmaps

Review

Laplace-Beltrami Operator

- Let $\mathcal{M}$ be a smooth Riemannian manifold. Then Laplace-Beltrami operator on twice differentiable function $f : \mathcal{M} \to \mathbb{R}$ is:

$$\triangle f \stackrel{\text{def}}{=} -\text{div } \nabla(f)$$

Review

Laplace-Beltrami Operator

- Let $\mathcal{M}$ be a smooth Riemannian manifold. Then Laplace-Beltrami operator on twice differentiable function $f : \mathcal{M} \to \mathbb{R}$ is:

$$\triangle f \stackrel{\text{def}}{=} -\text{div } \nabla(f)$$

- Eigenfunctions correspond to functions minimizing the "smoothness" criteria:

$$\arg \min_{||f||_{\mathcal{L}^2(\mathcal{M})=1}} \int_{\mathcal{M}} ||\nabla f(x)||^2$$

Review

- For compact manifold, discrete spectrum and any function $f \in \mathcal{L}^2(\mathcal{M})$ can be written as:

$$f(\mathbf{x}) = \sum_{i=0}^{\infty} a_i e_i(\mathbf{x})$$

where, e_i are eigenfunctions.

Review

Graph Equivalent: Given a weighted graph G = (V,E) with data points $\mathbf{x_i}$ as nodes, an optimal embedding $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ to $\mathbb{R}$ obtained by minimizing:

$$\sum_{ij} (y_i - y_j)^2 W_{ij}$$

where, weights, W_{ij} incur heavy penalty for dissimilar $\mathbf{x_i}$ and $\mathbf{x_j}$.
Optimal embedding:

$$L\mathbf{y} = \lambda D\mathbf{y}$$

where, $L = D - W$ is Laplace operator on graph and D diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

Review

Graph Equivalent: Given a weighted graph G = (V,E) with data points $\mathbf{x_i}$ as nodes, an optimal embedding $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ to $\mathbb{R}$ obtained by minimizing:

$$\sum_{ij} (y_i - y_j)^2 W_{ij}$$

where, weights, W_{ij} incur heavy penalty for dissimilar $\mathbf{x_i}$ and $\mathbf{x_j}$.
Optimal embedding:

$$L\mathbf{y} = \lambda D\mathbf{y}$$

where, $L = D - W$ is Laplace operator on graph and D diagonal matrix with $D_{ii} = \sum_j W_{ij}$.
Equivalent to Laplace-Beltrami operator on manifold.

Classification

- Consider a square-integrable function $m : \mathcal{M} \to \{-1, +1\}$ or any measurable set S_1 and S_2.
- Classification can be interpreted as problem of approximating m on manifold.
- Use labeled data to fit the coefficients of Laplacian to $m(\mathbf{x})$:

$$m(\mathbf{x}) \approx \sum_{0}^{N} a_i e_i(\mathbf{x})$$

.

- Eigenfunctions of Laplacian provide maximally smooth approximation.

Comparison: Can be compared with geodesic-NN (not very stable though)

Algorithm

Given k points $\mathbf{x_1}, \cdots, \mathbf{x_k} \in \mathbb{R}^l$, assume $s < k$ points have label $c_i \in \{-1, +1\}$ and rest unlabeled.
Goal: Label unlabeled data points.

Step 1: Construct adjacency graph with nodes i and j corresponding to $\mathbf{x_i}$ and $\mathbf{x_j}$ connected if either of them are within n nearest neighbors of each other. Take $w_{ij} = 1$ for connected nodes, else $w_{ij} = 0$.

Step 2: Compute p eigenvectors corresponding to the smallest eigenvalues for the problem:

$$L\mathbf{e} = \lambda D\mathbf{e}$$

Algorithm

Step 2: Create a matrix $\mathbf{E}$ taking the p eigenvectors as its rows:

$$\mathbf{E} = \begin{pmatrix} e_{11} & e_{12} & \cdots & e_{1k} \\ e_{21} & e_{22} & \cdots & e_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{p1} & e_{p2} & \cdots & e_{pk} \end{pmatrix}$$

Step 3: To approximate class, mimize error function,

$$Err(\mathbf{a}) = \sum_{i=1}^{s} \left(c_i - \sum_{j=1}^{p} a_j e_{ji} \right)^2$$

where the sum is taken over the labeled points. The solution is given as:

$$\mathbf{a} = (\mathbf{E}_{lab}^T \mathbf{E}_{lab})^{-1} \mathbf{E}_{lab}^T \mathbf{c}$$

Algorithm

Step 3: where, $\mathbf{c} = (c_1, \cdots, c_s)$ and $\mathbf{E_{lab}}$ is the matrix of values of eigenfunctions on the labeled points.

Step 4: If $\mathbf{x}_i$ is an unlabeled point, we put:

$$c_i = \begin{cases} 1, & \text{if } \sum_{j=1}^{p} e_{ij} a_j \geq 0 \\ -1, & \text{if } \sum_{j=1}^{p} e_{ij} a_j < 0 \end{cases}$$

Experiments

1. *Text Classification* - Newsgroup dataset

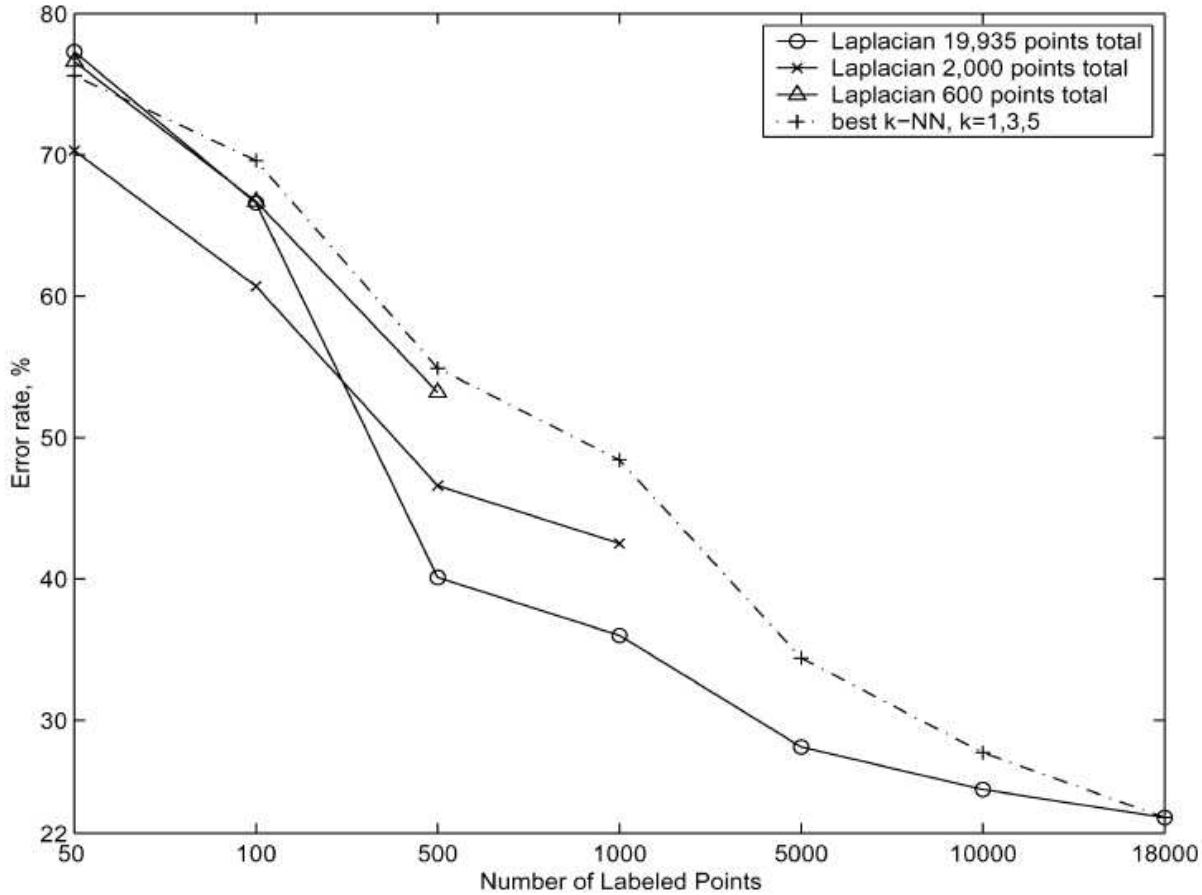


Figure 3. 20 Newsgroups data set. Error rates for different numbers of labeled and unlabeled points compared to best k-NN baseline.

Experiments

1. *Text Classification* - Newsgroup dataset

Table 2. Percentage error rates for various numbers of labeled points and eigenvectors. The total number of points is 19935. The error is calculated on the unlabeled part of the dataset.

	Number of eigenvectors									
Labeled points	5	10	20	50	100	200	500	1000	2000	Best k-NN
50	83.4	77.3	72.1							75.6
100	81.7	74.3	66.6	60.2						69.6
500	83.1	75.8	65.5	46.4	40.1	42.4				54.9
1000	84.6	77.6	67.1	47.0	37.7	36.0	42.3			48.4
5000	85.2	79.7	72.9	49.3	36.7	32.3	28.5	28.1	30.4	34.4
10000	83.8	79.8	73.8	49.8	36.9	31.9	27.9	25.9	25.1	27.7
18000	82.9	79.8	73.8	50.1	36.9	31.9	27.5	25.5	23.1	23.1

Experiments

2. *Comparison with geodesic NN* - MNIST digit dataset

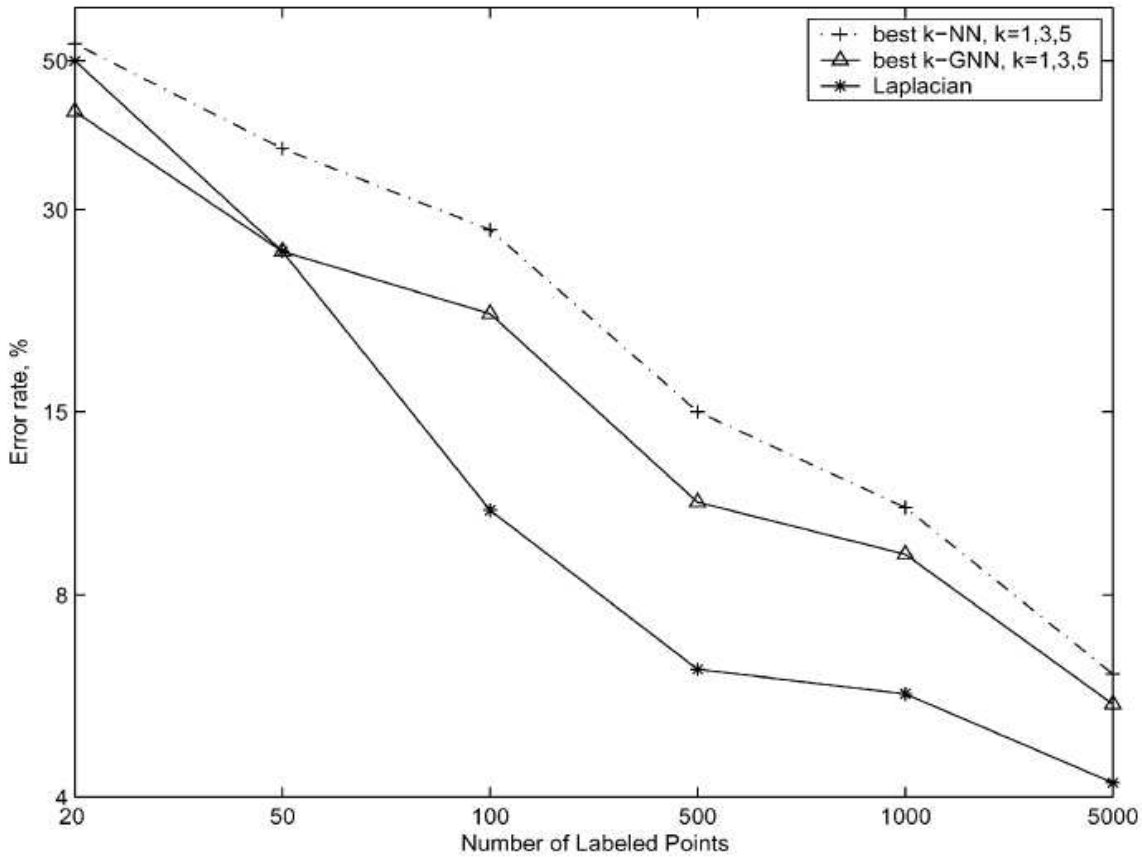
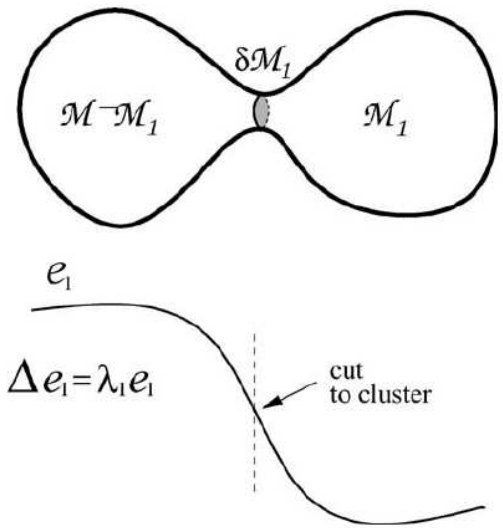


Figure 5. MNIST dataset. Error rates for different numbers of labeled and unlabeled points compared to best k-NN and best k-GNN. The total number of points is 10000.

Further Usage - Clustering

Consider partitioning the following manifold, $\mathcal{M}$ into two parts $\mathcal{M}_1$ and $\mathcal{M} - \mathcal{M}_1$ by a membrane $\delta\mathcal{M}_1$.



Cheeger's Constant

$$h_{\mathcal{M}} = \inf_{\mathcal{B}=\delta\mathcal{M}_1} \frac{\text{vol}^{n-1}\mathcal{B}}{\min(\text{vol}^n(\mathcal{M}_1), \text{vol}^n(\mathcal{M} - \mathcal{M}_1))}$$

Clustering contd.

Cheeger's Observation: If clustering is nice, then a function $\tilde{f}$ can be constructed as:

$$\tilde{f}(x) = \begin{cases} \frac{1}{\text{vol}(\mathcal{M}_1)} & x \in \mathcal{M}_1 \\ \frac{1}{\text{vol}(\mathcal{M}-\mathcal{M}_1)} & x \in \mathcal{M} - \mathcal{M}_1 \end{cases}$$

such that $\frac{\int_{\mathcal{M}} ||\nabla \tilde{f}||^2}{\int_{\mathcal{M}} ||\tilde{f}||^2}$ is closely related to Cheeger's contant (1970).

Clustering contd.

Cheeger's Observation: If clustering is nice, then a function $\tilde{f}$ can be constructed as:

$$\tilde{f}(x) = \begin{cases} \frac{1}{\text{vol}(\mathcal{M}_1)} & x \in \mathcal{M}_1 \\ \frac{1}{\text{vol}(\mathcal{M}-\mathcal{M}_1)} & x \in \mathcal{M} - \mathcal{M}_1 \end{cases}$$

such that $\frac{\int_{\mathcal{M}} ||\nabla \tilde{f}||^2}{\int_{\mathcal{M}} ||\tilde{f}||^2}$ is closely related to Cheeger's contant (1970). The first non-trivial eigenfunction of $\triangle$, e_1 is given as:

$$\arg \min_{f \perp const} \frac{\int_{\mathcal{M}} ||\nabla f||^2}{\int_{\mathcal{M}} ||f||^2}$$

which is close to $\tilde{f}$.

Clustering contd.

Thus, an approximate to optimal clustering is given as:

$$\mathcal{M}_1 = \{x | e_1(x) > 0\}$$

$$\mathcal{M} - \mathcal{M}_1 = \{x | e_1(x) \leq 0\}$$

Graph Equivalence: Equivalent relationship on graph reduces to normalized cut, thus relating it to Cheeger's constant.

References


M. Belkin, and P. Niyogi.
Semi-Supervised Learning on Riemannian Manifolds.
Machine Learning, 56:209–239, 2004.

Questions