Local Learning Algorithms

for Transductive Classification, Clustering and Data Projection

Mingrui Wu

Max Planck Institute for Biological Cybernetics

July 5th, 2007

Outline

Supervised Local Learning Algorithms

Supervised learning problem, training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

- Global Learning Algorithms
  - A model $\mathcal{M}$ is built with all the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.
  - $\mathcal{M}$ is used to to predict the labels of any unseen test data.
- Local Learning Algorithms
  - For a given test point $\mathbf{x}$, build a model $\mathcal{M}_x$ only using $\{(\mathbf{x}_i, y_i)\}_{\mathbf{x}_i \in \mathcal{N}_x}$.
  - Different models may be used for different test points.

Supervised Local Learning Algorithms

Supervised learning problem, training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$

- Global Learning Algorithms
  - A model $\mathcal{M}$ is built with all the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$.
  - $\mathcal{M}$ is used to to predict the labels of any unseen test data.
- Local Learning Algorithms
  - For a given test point $\mathbf{x}$, build a model $\mathcal{M}_x$ only using $\{(\mathbf{x}_i, y_i)\}_{\mathbf{x}_i \in \mathcal{N}_x}$.
  - Different models may be used for different test points.

Local Learning Is Good

Local learning algorithms often outperform global ones since local models are trained only with the points that are related to the particular test data.

The good performance of local learning methods indicates:
The label of a point can be well estimated based on its neighbors.

Transductive Classification via Local Learning Regularization

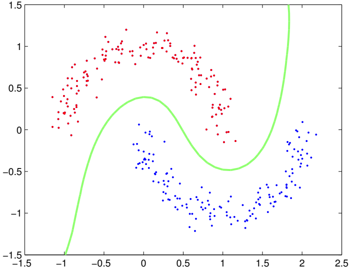
Supervised and Transductive Classification

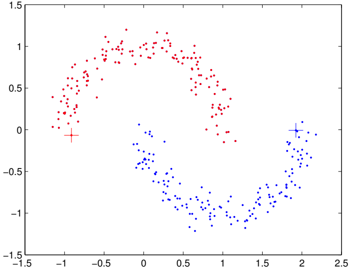
- Binary Supervised Classification
  - Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathcal{R}^d$, $y_i \in \{-1, 1\}$.
  - Goal: Classification function $f(\mathbf{x})$.
  - Learning only from labeled data.
- Binary Transductive Classification (TC)
  - Given:
    - Labeled data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathcal{R}^d$, $y_i \in \{-1, 1\}$.
    - Unlabeled data: $\{(\mathbf{x}_i)\}_{i=l+1}^{l+u}$, typically $u >> l$.
  - Goal: Predict the class labels of the given unlabeled points.
  - Learning from both labeled and unlabeled data.

Supervised and Transductive Classification

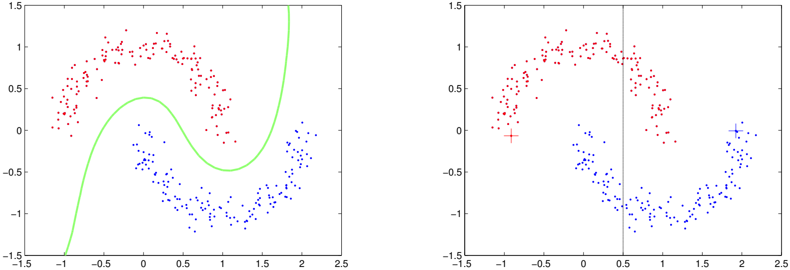
- Binary Supervised Classification
  - Given: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathcal{R}^d$, $y_i \in \{-1, 1\}$.
  - Goal: Classification function $f(\mathbf{x})$.
  - Learning only from labeled data.
- Binary Transductive Classification (TC)
  - Given:
    - Labeled data: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathcal{R}^d$, $y_i \in \{-1, 1\}$.
    - Unlabeled data: $\{(\mathbf{x}_i)\}_{i=l+1}^{l+u}$, typically $u >> l$.
  - Goal: Predict the class labels of the given unlabeled points.
  - Learning from both labeled and unlabeled data.

A Toy Example





Can We Ignore the Unlabeled Data?



Motivation

- Labeled data are expensive or difficult to obtain.
- Unlabeled data are much easier to get.
- Example: Web page classification.

Prior Assumption Is Important

- In many TC algorithms [Zhu et al., 2003, Belkin et al., 2005], each $\mathbf{x}_i$ is assigned a real value f_i

$$y_i = \text{sign}(f_i) \qquad l+1 \leq i \leq l+u$$

- Main part: computing f_i of each $\mathbf{x}_i$.
- Key: the prior assumption about the properties that f_i should have over the data points.
- Cluster assumption: If two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are on the same cluster, then the values of f_i and f_j should be similar to each other.

A Quadratic Objective Function for TC

A typical formulation for TC [Zhu et al., 2003, Zhou et al., 2004]

$$\min_{\mathbf{f}\in\mathbb{R}^n} \mathbf{f}^\top \mathbf{R}\mathbf{f} + (\mathbf{f}-\mathbf{y})^\top \mathbf{C}(\mathbf{f}-\mathbf{y})$$

where

- $\mathbf{f} = [f_1, \ldots, f_n]^\top \in \mathbb{R}^n$.
- $\mathbf{R} \in \mathbb{R}^{n\times n}$: regularization matrix.
- $\mathbf{y} = [y_1, \ldots, y_l, 0, \ldots, 0]^\top \in \mathbb{R}^n$.
- $\mathbf{C}$: a diagonal matrix. $c_i = C_l > 0$ for $1 \le i \le l$, and $c_i = C_u \ge 0$ for $l+1 \le i \le n$.

Laplacian Regularizer

- Graph Laplacian [Zhu et al., 2003], $\mathbf{R} = \mathbf{L}$.

$$\mathbf{f}^\top \mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i - f_j)^2$$

w_{ij}, similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$

$$w_{ij} = \exp(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2)$$

- Smoothness constraint: The labels should be similar among nearby points.

TC via Local Learning

Local Learning Problem (LL-Problem)

For a data point $\mathbf{x}_i$, given the values of f_j at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i at $\mathbf{x}_i$?

- $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i} \to f_i$, a learning problem.
- Local Learning Regularizer:

$$\sum_{i=1}^{n} (f_i - o_i(\mathbf{x}_i))^2$$

$o_i(\cdot)$, trained with $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$

- Idea: f_i should be well estimated locally based on the neighboring points of $\mathbf{x}_i$, using supervised learning algorithms.

TC via Local Learning

Local Learning Problem (LL-Problem)

For a data point $\mathbf{x}_i$, given the values of f_j at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i at $\mathbf{x}_i$?

- $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i} \rightarrow f_i$, a learning problem.
- Local Learning Regularizer:

$$\sum_{i=1}^{n} (f_i - o_i(\mathbf{x}_i))^2$$

  $o_i(\cdot)$, trained with $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$
- Idea: f_i should be well estimated locally based on the neighboring points of $\mathbf{x}_i$, using supervised learning algorithms.

TC via Local Learning

Local Learning Problem (LL-Problem)

For a data point $\mathbf{x}_i$, given the values of f_j at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i at $\mathbf{x}_i$?

- $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i} \to f_i$, a learning problem.
- Local Learning Regularizer:

$$\sum_{i=1}^{n} (f_i - o_i(\mathbf{x}_i))^2$$

$o_i(\cdot)$, trained with $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$

- Idea: f_i should be well estimated locally based on the neighboring points of $\mathbf{x}_i$, using supervised learning algorithms.

TC via Local Learning

Local Learning Problem (LL-Problem)

For a data point $\mathbf{x}_i$, given the values of f_j at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i at $\mathbf{x}_i$?

- $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i} \rightarrow f_i$, a learning problem.
- Local Learning Regularizer:

$$\sum_{i=1}^{n} (f_i - o_i(\mathbf{x}_i))^2$$

  $o_i(\cdot)$, trained with $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$
- Idea: f_i should be well estimated locally based on the neighboring points of $\mathbf{x}_i$, using supervised learning algorithms.

Computing $o_i(\mathbf{x}_i)$ (1/2)

Following [Bottou & Vapnik, 1992],

- Linear model

$$o_i(\mathbf{x}) = \mathbf{w}_i^\top (\mathbf{x} - \mathbf{x}_i) + b_i, \quad \forall \mathbf{x} \in \mathbb{R}^d$$

- Training data, $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$. Training problem:

$$\min_{\mathbf{w}_i \in \mathbb{R}^d, b_i \in \mathbb{R}} \lambda \|\mathbf{w}_i\|^2 + \sum_{\mathbf{x}_j \in \mathcal{N}_i} (o_i(\mathbf{x}_j) - f_j)^2$$

- Solution

$$o_i(\mathbf{x}_i) = \boldsymbol{\alpha}_i^\top \mathbf{f}_i$$

$\mathbf{f}_i \in \mathbb{R}^{n_i}$, the vector $[f_j]^\top$ for $\mathbf{x}_j \in \mathcal{N}_i$.
$o_i(\mathbf{x}_i)$ can be computed analytically, even if we do not know the values of $\{f_j\}_{\mathbf{x}_j \in \mathcal{N}_i}$.

Computing $o_i(\mathbf{x}_i)$ (2/2)

- Matrix form of $o_i(\mathbf{x}_i)$

$$\mathbf{o} = \mathbf{Af}$$

$$\mathbf{o} = [o_1(\mathbf{x}_1), \ldots, o_n(\mathbf{x}_n)]^\top, \mathbf{f} = [f_1, \ldots, f_n]^\top$$

- An example:
$o_1(\mathbf{x}_1) = a \times f_2 + b \times f_3, \quad o_2(\mathbf{x}_2) = c \times f_1 + d \times f_4, \ldots.$
then

$$\begin{pmatrix} o_1(\mathbf{x}_1) \\ o_2(\mathbf{x}_2) \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 & a & b & 0 & \cdots \\ c & 0 & 0 & d & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \end{pmatrix}$$

Local Learning Regularizer

- Local Learning Regularizer:

$$\sum_{i=1}^{n}(f_i - o_i(\mathbf{x}_i))^2 = \|\mathbf{f} - \mathbf{o}\|^2 = \|\mathbf{f} - \mathbf{A}\mathbf{f}\|^2 = \mathbf{f}^\top \mathbf{R}_L \mathbf{f}$$

$$\mathbf{R}_L = (\mathbf{I} - \mathbf{A})^\top (\mathbf{I} - \mathbf{A})$$

- Quadratic objective:

$$\min_{\mathbf{f} \in \mathbb{R}^n} \mathbf{f}^\top \mathbf{R}_L \mathbf{f} + (\mathbf{f} - \mathbf{y})^\top \mathbf{C} (\mathbf{f} - \mathbf{y})$$

Comparison with Laplacian Regularizer

- Laplacian regularizer:

$$\mathbf{f}^\top \mathbf{L}\mathbf{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i - f_j)^2$$

- Current explanations: smoothness constraints, manifold regularization, random walk.
- Implicit answer to LL-problem: Setting $\frac{\partial}{\partial \mathbf{f}}\mathbf{f}^\top \mathbf{L}\mathbf{f}$ to $\mathbf{0}$, we have,

$$f_i = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} f_j}{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}}$$

Leave-One-Out (LOO) Error

- Quadratic objective

$$\min_{\mathbf{f}\in\mathbb{R}^n} \mathbf{f}^\top \mathbf{R}\mathbf{f} + (\mathbf{f}-\mathbf{y})^\top \mathbf{C}(\mathbf{f}-\mathbf{y})$$

Solution: $\mathbf{f} = (\mathbf{R}+\mathbf{C})^{-1}\mathbf{C}\mathbf{y}$, let $\mathbf{M} = (\mathbf{R}+\mathbf{C})^{-1}$.

- LOO procedure for TC: In the i-th iteration ($1 \leq i \leq l$), $(\mathbf{x}_i, y_i)$ –> $\mathbf{x}_i$, solution $\mathbf{f}^{(i)}$.
- To compute LOO error: We only need to know $f_i^{(i)}$.

Computing LOO Error Efficiently

$$f_i^{(i)} = \frac{f_i - C_l y_i m_{ii}}{1 - (C_l - C_u)m_{ii}} \qquad 1 \leq i \leq l$$

Experimental Results

Dataset	Lap-Reg	NLap-Reg	LLE-Reg	LL-Reg
g241c	39.00±2.23	45.00±3.92	41.46±4.51	**21.36±3.67**
g241d	36.12±1.50	43.31±3.30	40.15±4.05	**22.51±1.79**
Digit1	**3.02±0.84**	**2.91±0.59**	**2.54±0.72**	**2.63±0.66**
USPS	7.09±3.37	4.60±2.04	4.70±1.86	**3.67±1.24**
COIL	**11.11±2.72**	**10.71±3.29**	13.61±4.01	12.04±2.30
BCI	47.60±2.29	47.22±2.31	44.23±3.74	**31.15±5.02**
Text	29.29±2.36	**23.55±3.55**	50.11±0.37	**24.23±3.28**
Banana	14.26±1.69	14.15±1.96	17.09±2.48	**12.75±1.70**
Diabetis	32.80±2.54	31.93±2.71	33.31±2.51	**27.63±2.40**
German	31.76±2.51	**30.82±1.40**	33.69±2.95	**29.95±2.49**
Image	14.39±1.63	14.28±1.88	18.67±2.44	**12.08±2.07**
Ringnorm	19.06±2.17	**9.66±0.86**	11.85±1.48	10.28±0.38
Splice	37.48±3.75	36.01±8.50	39.36±2.31	**27.27±5.05**
Twonorm	**4.17±1.30**	**4.05±1.29**	7.54±1.33	**3.35±1.02**
Waveform	17.09±3.27	16.88±3.28	19.02±1.80	**13.50±1.94**

Remarks and Questions

Remarks

- Local learning regularization for TC.
- A flexible framework, adapting various learning algorithms for TC.
- Examining some current regularizers under this framework.
- An efficient way to compute the LOO error.

Questions

- No labeled points at all, unsupervised learning?
- Nonlinear local models?

Remarks and Questions

Remarks

- Local learning regularization for TC.
- A flexible framework, adapting various learning algorithms for TC.
- Examining some current regularizers under this framework.
- An efficient way to compute the LOO error.

Questions

- No labeled points at all, unsupervised learning?
- Nonlinear local models?

A Local Learning Approach for Clustering

The clustering problem

Clustering Problem

- n data points, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$.
- $c > 0$.
- goal: partition the given data into c clusters.
- current methods: k-means, single-link, spectral clustering.

Representation of Clustering Results

- Partition Matrix: $\mathbf{P} \in \{0, 1\}^{n \times c}$
- Scaled Partition Matrix: $\mathbf{F} = \mathbf{P}(\mathbf{P}^\top \mathbf{P})^{-\frac{1}{2}}$

$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \end{pmatrix}$$

- Two useful properties:

$$\mathbf{F}^\top \mathbf{F} = (\mathbf{P}^\top \mathbf{P})^{-\frac{1}{2}} \mathbf{P}^\top \mathbf{P} (\mathbf{P}^\top \mathbf{P})^{-\frac{1}{2}} = \mathbf{I}$$

$$\mathbf{P} = P(\mathbf{F}) = Diag(\mathbf{F}\mathbf{F}^\top)^{-\frac{1}{2}} \mathbf{F}$$

Basic Idea

LL-Problem, $\mathbf{F} \in \mathbb{R}^{n \times c}$, f_i^l

For a data point $\mathbf{x}_i$ and a cluster $\mathcal{C}_l$, given the values of f_j^l at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i^l at $\mathbf{x}_i$?

Objective function:

$$\min_{\mathbf{F} \in \mathbb{R}^{n \times c}} \quad \sum_{l=1}^{c} \sum_{i=1}^{n} (f_i^l - o_i^l(\mathbf{x}_i))^2 \tag{1}$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix} \tag{2}$$

$o_i^l(\cdot)$, trained with $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j \in \mathcal{N}_i}$.

Basic Idea

LL-Problem, $\mathbf{F} \in \mathbb{R}^{n \times c}$, f_i^l

For a data point $\mathbf{x}_i$ and a cluster $\mathcal{C}_l$, given the values of f_j^l at $\mathbf{x}_j \in \mathcal{N}_i$, what should be the proper value of f_i^l at $\mathbf{x}_i$?

Objective function:

$$\min_{\mathbf{F} \in \mathbb{R}^{n \times c}} \quad \sum_{l=1}^{c} \sum_{i=1}^{n} (f_i^l - o_i^l(\mathbf{x}_i))^2 \tag{1}$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix} \tag{2}$$

$o_i^l(\cdot)$, trained with $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j \in \mathcal{N}_i}$.

Computing $o_i^l(\mathbf{x}_i)$

- Training data: $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j \in \mathcal{N}_i}$. Kernel learning algorithms: $o_i^l(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathcal{N}_i} \beta_{ij}^l K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}_i^\top \beta_i^l$
- kernel ridge regression:

$$\min_{\beta_i^l \in \mathbb{R}^{n_i}} \lambda (\beta_i^l)^\top \mathbf{K}_i \beta_i^l + \left\| \mathbf{K}_i \beta_i^l - \mathbf{f}_i^l \right\|^2$$

$\mathbf{K}_i = [K(\mathbf{x}_u, \mathbf{x}_v)] \in \mathbb{R}^{n_i \times n_i}$, for $\mathbf{x}_u, \mathbf{x}_v \in \mathcal{N}_i$.

- solution of kernel ridge regression: $\beta_i^l = (\mathbf{K}_i + \lambda \mathbf{I})^{-1} \mathbf{f}_i^l$
- $o_i^l(\mathbf{x}_i) = \mathbf{k}_i^\top (\mathbf{K}_i + \lambda \mathbf{I})^{-1} \mathbf{f}_i^l = \alpha_i^\top \mathbf{f}_i^l$, $\alpha_i^\top = \mathbf{k}_i^\top (\mathbf{K}_i + \lambda \mathbf{I})^{-1}$

Computing $o_i^l(\mathbf{x}_i)$

- Training data: $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j \in \mathcal{N}_i}$. Kernel learning algorithms: $o_i^l(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathcal{N}_i} \beta_{ij}^l K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{k}_i^\top \boldsymbol{\beta}_i^l$
- kernel ridge regression:

$$\min_{\boldsymbol{\beta}_i^l \in \mathbb{R}^{n_i}} \lambda(\boldsymbol{\beta}_i^l)^\top \mathbf{K}_i \boldsymbol{\beta}_i^l + \left\| \mathbf{K}_i \boldsymbol{\beta}_i^l - \mathbf{f}_i^l \right\|^2$$

$\mathbf{K}_i = [K(\mathbf{x}_u, \mathbf{x}_v)] \in \mathbb{R}^{n_i \times n_i}$, for $\mathbf{x}_u, \mathbf{x}_v \in \mathcal{N}_i$.

- solution of kernel ridge regression: $\boldsymbol{\beta}_i^l = (\mathbf{K}_i + \lambda \mathbf{I})^{-1} \mathbf{f}_i^l$
- $o_i^l(\mathbf{x}_i) = \mathbf{k}_i^\top (\mathbf{K}_i + \lambda \mathbf{I})^{-1} \mathbf{f}_i^l = \boldsymbol{\alpha}_i^\top \mathbf{f}_i^l$,
  $\boldsymbol{\alpha}_i^\top = \mathbf{k}_i^\top (\mathbf{K}_i + \lambda \mathbf{I})^{-1}$

Quadratic Objective Function

- Matrix form: $\mathbf{o}^l = \mathbf{A}\mathbf{f}^l$
  $\mathbf{o}^l = [o_1^l(\mathbf{x}_1), \ldots, o_n^l(\mathbf{x}_n)]^\top$, $\mathbf{f}^l = [f_1^l, \ldots, f_n^l]^\top$
- Objective function:

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \quad \sum_{l=1}^{c}\sum_{i=1}^{n}(f_i^l - o_i^l(\mathbf{x}_i))^2 = \sum_{l=1}^{c}\left\|\mathbf{f}^l - \mathbf{o}^l\right\|^2 \qquad (3)$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix} \qquad (4)$$

- Quadratic objective function:

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \quad \sum_{l=1}^{c}\left\|\mathbf{f}^l - \mathbf{A}\mathbf{f}^l\right\|^2 = \sum_{l=1}^{c}(\mathbf{f}^l)^\top \mathbf{T}\mathbf{f}^l = trace(\mathbf{F}^\top\mathbf{T}\mathbf{F})$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix}$$

$$\mathbf{T} = (\mathbf{I}-\mathbf{A})^\top(\mathbf{I}-\mathbf{A})$$

Quadratic Objective Function

- Matrix form: $\mathbf{o}^l = \mathbf{A}\mathbf{f}^l$
  $\mathbf{o}^l = [o_1^l(\mathbf{x}_1), \ldots, o_n^l(\mathbf{x}_n)]^\top$, $\mathbf{f}^l = [f_1^l, \ldots, f_n^l]^\top$
- Objective function:

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \quad \sum_{l=1}^{c}\sum_{i=1}^{n}(f_i^l - o_i^l(\mathbf{x}_i))^2 = \sum_{l=1}^{c}\left\|\mathbf{f}^l - \mathbf{o}^l\right\|^2 \tag{3}$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix} \tag{4}$$

- Quadratic objective function:

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \quad \sum_{l=1}^{c}\left\|\mathbf{f}^l - \mathbf{A}\mathbf{f}^l\right\|^2 = \sum_{l=1}^{c}(\mathbf{f}^l)^\top\mathbf{T}\mathbf{f}^l = trace(\mathbf{F}^\top\mathbf{T}\mathbf{F})$$

$$\text{s.t.} \quad \mathbf{F} \text{ is a scaled partition matrix}$$

$$\mathbf{T} = (\mathbf{I}-\mathbf{A})^\top(\mathbf{I}-\mathbf{A})$$

Relaxation

Relax **F** to continuous domain

$$\min_{\mathbf{F}\in\mathbb{R}^{n\times c}} \quad \mathit{trace}(\mathbf{F}^\top \mathbf{T}\mathbf{F})$$
$$\text{s.t.} \quad \mathbf{F}^\top \mathbf{F} = \mathbf{I}$$

Solution:

$$\{\mathbf{F}^\star\mathbf{R} : \mathbf{R} \in \mathbb{R}^{c\times c}, \quad \mathbf{R}^\top\mathbf{R} = \mathbf{I}\}$$

where $\mathbf{F}^\star \in \mathbb{R}^{n\times c}$, consists of the c smallest eigenvectors of **T**

Discretization: Obtaining the Final Clustering Result

- Get real valued partition matrix: $\mathbf{P}^\star = P(\mathbf{F}^\star)$
- A property: $P(\mathbf{F}^\star\mathbf{R}) = \mathbf{P}^\star\mathbf{R} \quad \forall \mathbf{R}^\top\mathbf{R} = \mathbf{I}$.
- $\mathbf{F}^\star\mathbf{R}$ close to the true SPM, $\mathbf{P}^\star\mathbf{R}$ close to the corresponding discrete PM.
- Compute $\mathbf{R}$ and discrete PM $\mathbf{P}$ [Yu & Shi, 2003]:

$$\min_{\mathbf{P}\in\mathbb{R}^{n\times c},\mathbf{R}\in\mathbb{R}^{c\times c}} \quad \|\mathbf{P} - \mathbf{P}^\star\mathbf{R}\|^2 \tag{5}$$

$$\text{subject to} \quad \mathbf{P} \in \{0,1\}^{n\times c}, \quad \mathbf{P}\mathbf{1}_c = \mathbf{1}_n \tag{6}$$

$$\mathbf{R}^\top\mathbf{R} = \mathbf{I} \tag{7}$$

Numerical Results

		U3568	U49	UMist	UMist5	News4a	News4b
NMI, cosine	Spec-Clst	*0.6575*	*0.3608*	0.7483	*0.8810*	*0.6468*	*0.5765*
	LLCA1	**0.8720**	**0.6241**	**0.8003**	**1**	**0.7587**	**0.7125**
	LLCA2	**0.8720**	**0.6241**	*0.7889*	**1**	**0.7587**	**0.7125**
	k-means	0.5202	0.2352	0.6479	0.7193	0.0800	0.0380
NMI, Gaussian	Spec-Clst	0.8245	0.4319	*0.8099*	*0.8773*	**0.4039**	**0.1861**
	LLCA1	**0.8493**	**0.5980**	**0.8377**	**1**	*0.2642*	*0.1776*
	LLCA2	*0.8467*	*0.5493*	**0.8377**	**1**	0.0296	0.0322
	k-means	0.5202	0.2352	0.6479	0.7193	0.0800	0.0380
Error (%), cosine	Spec-Clst	32.93	*16.56*	46.26	*9.29*	*28.26*	*21.73*
	LLCA1	**3.57**	**8.01**	**36.00**	**0**	**7.99**	**9.65**
	LLCA2	**3.57**	**8.01**	*38.43*	**0**	**7.99**	**9.65**
	k-means	*22.16*	22.30	56.35	36.43	70.62	74.08
Error (%), Gaussian	Spec-Clst	5.68	13.51	41.74	*10.00*	**42.34**	*64.71*
	LLCA1	**4.61**	**8.43**	**33.91**	**0**	*47.24*	**53.25**
	LLCA2	*4.70*	*9.80*	*37.22*	**0**	74.38	72.97
	k-means	22.16	22.30	56.35	36.43	70.62	74.08

Remarks and Questions

Remarks

- Adapting the local learning idea for the clustering problem.
- Cost function: the cluster label of each data point can be well estimated based on its neighbors.
- Nonlinear local models.
- Easy implementation, encouraging results.

Questions

- The LL approach is effective to explore the relationship among neighboring points. Any other applications?

Remarks and Questions

Remarks

- Adapting the local learning idea for the clustering problem.
- Cost function: the cluster label of each data point can be well estimated based on its neighbors.
- Nonlinear local models.
- Easy implementation, encouraging results.

Questions

- The LL approach is effective to explore the relationship among neighboring points. Any other applications?

Local Learning Projections

Linear Projection

- Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, $\mathbf{x}_i \in \mathcal{X} \in \mathbb{R}^d$, $y_i \in \{1, 2, \ldots, c\}$.
- Goal: Find a low dimensional subspace of $\mathcal{X}$, which retains the discriminating information for classification.
- Projection matrix $\mathbf{P} \in \mathbb{R}^{d \times p}$, $p < d$, $\mathbf{x} \to \mathbf{P}^\top \mathbf{x}$.
- Reducing noise, removing redundant information irrelevant to the classification task.

Some Linear Projection Methods

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Locality Preserving Projection (LPP) [He & Niyogi, 2004]

Local Learning Projections (LLP)

Idea

The projection of a point can be well estimated based on its neighbors in the same class.

Objective function:

$$\min_{\mathbf{P}\in\mathbb{R}^{d\times p},\mathbf{F}\in\mathbb{R}^{p\times n}} \sum_{l=1}^{p}\sum_{i=1}^{n}(f_i^l - o_i^l(\mathbf{x}_i))^2 = \sum_{l=1}^{p}\left\|\mathbf{f}^l - \mathbf{o}^l\right\|^2 \quad (8)$$

$$\text{s.t.} \quad \mathbf{F} = \mathbf{P}^\top\mathbf{X} \quad (9)$$

$$\mathbf{P}^\top\mathbf{P} = \mathbf{I} \quad (10)$$

$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $o_i^l(\cdot)$, trained with $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j\in\mathcal{N}_i}$

Local Learning Projections (LLP)

Idea

The projection of a point can be well estimated based on its neighbors in the same class.

Objective function:

$$\min_{\mathbf{P}\in\mathbb{R}^{d\times p},\mathbf{F}\in\mathbb{R}^{p\times n}} \quad \sum_{l=1}^{p}\sum_{i=1}^{n}(f_i^l - o_i^l(\mathbf{x}_i))^2 = \sum_{l=1}^{p}\left\|\mathbf{f}^l - \mathbf{o}^l\right\|^2 \tag{8}$$

$$\text{s.t.} \quad \mathbf{F} = \mathbf{P}^\top \mathbf{X} \tag{9}$$

$$\mathbf{P}^\top \mathbf{P} = \mathbf{I} \tag{10}$$

$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, $o_i^l(\cdot)$, trained with $\{(\mathbf{x}_j, f_j^l)\}_{\mathbf{x}_j\in\mathcal{N}_i}$

Locality Preserving Projections (LPP)

Consider the case $p = 1$, $f_i = \mathbf{p}^\top \mathbf{x}_i$

- Basic idea of LPP: $E_{LPP}(\mathbf{f}) = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 w_{ij}$
- Setting $\frac{\partial}{\partial \mathbf{f}} E_{LPP}(\mathbf{f})$ to $\mathbf{0}$, optimal $\mathbf{f}$: $f_i = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} f_j}{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}}$
- LLP explicitly requires that f_i can be well estimated based on the neighbors of $\mathbf{x}_i$, while LPP specifies this implicitly.
- In LLP, f_i is estimated by $o_i(\mathbf{x}_i)$, which is trained with well established regression approaches, while in LPP, f_i is estimated with the local average.

Locality Preserving Projections (LPP)

Consider the case $p = 1$, $f_i = \mathbf{p}^\top \mathbf{x}_i$

- Basic idea of LPP: $E_{LPP}(\mathbf{f}) = \frac{1}{2}\sum_{i,j}(f_i - f_j)^2 w_{ij}$
- Setting $\frac{\partial}{\partial \mathbf{f}} E_{LPP}(\mathbf{f})$ to $\mathbf{0}$, optimal $\mathbf{f}$: $f_i = \frac{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij} f_j}{\sum_{\mathbf{x}_j \in \mathcal{N}_i} w_{ij}}$
- LLP explicitly requires that f_i can be well estimated based on the neighbors of $\mathbf{x}_i$, while LPP specifies this implicitly.
- In LLP, f_i is estimated by $o_i(\mathbf{x}_i)$, which is trained with well established regression approaches, while in LPP, f_i is estimated with the local average.

Comparison with PCA (1/2)

Global estimation error:

Input data $\{\mathbf{x}_i\}_{i=1}^n$, a vector $\mathbf{f} = [f_1, \ldots, f_n]^\top \in \mathbb{R}^n$, define

$$E_{global}(\mathbf{f}) = \sum_{i=1}^{n} (f_i - o_{all}(\mathbf{x}_i))^2$$

$o_{all}(\cdot)$: trained with $\{(\mathbf{x}_i, f_i)\}_{i=1}^n$, using kernel ridge regression.

Proposition

Let $\bar{\mathbf{f}} = [\bar{f}_1, \ldots, \bar{f}_n]^\top \in \mathbb{R}^n$, where $\bar{f}_i$ denotes the projection value of $\mathbf{x}_i$ given by KPCA algorithm. Then among all the unit length vectors, $\bar{\mathbf{f}} / \|\bar{\mathbf{f}}\|$ is the one with the minimal global estimation error. Namely,

$$\bar{\mathbf{f}} / \|\bar{\mathbf{f}}\| = arg \min_{\mathbf{f}^\top \mathbf{f}=1} E_{global}(\mathbf{f})$$

Comparison with PCA (1/2)

Global estimation error:

Input data $\{\mathbf{x}_i\}_{i=1}^n$, a vector $\mathbf{f} = [f_1, \ldots, f_n]^\top \in \mathbb{R}^n$, define

$$E_{global}(\mathbf{f}) = \sum_{i=1}^{n} (f_i - o_{all}(\mathbf{x}_i))^2$$

$o_{all}(\cdot)$: trained with $\{(\mathbf{x}_i, f_i)\}_{i=1}^n$, using kernel ridge regression.

Proposition

Let $\bar{\mathbf{f}} = [\bar{f}_1, \ldots, \bar{f}_n]^\top \in \mathbb{R}^n$, where $\bar{f}_i$ denotes the projection value of $\mathbf{x}_i$ given by KPCA algorithm. Then among all the unit length vectors, $\bar{\mathbf{f}} / \|\bar{\mathbf{f}}\|$ is the one with the minimal global estimation error. Namely,

$$\bar{\mathbf{f}} / \|\bar{\mathbf{f}}\| = arg \min_{\mathbf{f}^\top \mathbf{f} = 1} E_{global}(\mathbf{f})$$

Comparison with PCA (2/2)

- PCA minimizes global estimation error:

$$E_{global}(\mathbf{f}) = \sum_{i=1}^{n} (f_i - o_{all}(\mathbf{x}_i))^2$$

$o_{all}(\cdot)$: trained with $\{(\mathbf{x}_i, f_i)\}_{i=1}^{n}$. Not using class labels.

- LLP minimizes local estimation error:

$$E_{local}(\mathbf{f}) = \sum_{i=1}^{n} (f_i - o_i(\mathbf{x}_i))^2$$

$o_i(\cdot)$: trained with $\{(\mathbf{x}_j, f_j)\}_{\mathbf{x}_j \in \mathcal{N}_i}$. Using class labels.

Experimental Results

Dataset	m	1-NN	PCA	LDA	LPP	LLP
Yale	5	42.1±4.0	42.1±4.0	24.3±2.7	22.4±3.6	**19.8±3.5**
	6	40.3±4.1	40.3±4.1	21.5±3.8	20.2±4.7	**16.9±3.5**
	7	38.8±4.3	38.8±4.3	19.8±4.1	19.7±4.4	**14.8±3.7**
ORL	5	11.9±1.2	11.9±1.2	5.7±1.0	6.7±1.1	**3.1±1.2**
	6	9.1±2.0	9.1±2.0	4.3±1.2	4.5±1.7	**2.6±1.1**
	7	6.9±2.4	6.9±2.4	3.5±1.2	3.8±1.3	**2.0±1.0**
YaleB	10	55.2±1.0	55.2±1.0	21.6±1.1	19.6±3.1	**16.6±1.0**
	20	41.7±0.8	41.7±0.8	13.8±0.9	17.6±2.9	**11.2±1.5**
	30	34.5±1.3	34.5±1.3	12.8±1.1	13.6±1.1	**8.7±1.4**
PIE	10	65.0±0.5	65.0±0.5	29.7±1.3	28.8±1.4	**18.7±0.9**
	20	48.8±0.7	48.8±0.7	21.1±0.7	20.7±1.1	**16.6±0.6**
	30	37.9±0.8	37.9±0.8	10.8±0.6	**9.7±0.8**	14.3±0.7
UMist	5	15.2±3.8	15.2±3.8	10.2±2.8	14.5±3.7	**6.2±2.7**
	6	12.0±3.1	12.0±3.1	7.2±2.3	12.0±2.3	**4.5±2.3**
	7	9.7±2.3	9.7±2.3	5.9±2.2	9.8±2.4	**3.7±1.8**

Remarks

- Adapting the local learning idea for data projection.
- LLP can keep local relationship among neighboring points.
- A new explanation of PCA, based on which we can see the advantages of LLP over PCA.

Summary

Summary

- Adapting the local learning idea for TC, clustering and data projection.
- Investigating existing methods from the local learning point of view.
- Asking the relevant question explicitly.

Future Works

- Other applications, image segmentation, image matting, ranking, etc.
- Multi-view clustering and transductive learning.
- Local learning on graphs.

Thank you very much for your attention!

Belkin, M., Niyogi, P., & Sindhwani, V. (2005).
On manifold regularization.
AISTATS05

Bottou, L., & Vapnik, V. (1992).
Local learning algorithms.
Neural Computation, *4*, 888–900.

He, X., & Niyogi, P. (2004).
Locality preserving projections.
In S. Thrun, L. Saul and B. Schölkopf (Eds.), *Advances in neural information processing systems 16*.

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H. (2005).
Face recognition using laplacianfaces.
IEEE Transactions on Pattern Analysis and machine Intelligence, *27*, 328–340.

Yu, S. X., & Shi, J. (2003).
Multiclass spectral clustering.

International Conference on Computer Vision.
ACM.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004).
Learning with local and global consistency.
Advances in Neural Information Processing Systems 16.

Zhu, X., Ghaharmani, Z., & Lafferty, J. (2003).
Semi-supervised learning using gaussian fields and harmonic functions.
Proc. 20th International Conference on Machine Learning.