Adapting Kernel-Based Methods: Multi-class SVM and Semi-Supervised Spectral Analysis

Candidature Presentation for Mphil Degree

Wu Zhili
Supervisors: C.H.Li, Y.Y.Tang

CS Dept, HKBU
Overview

- Introduction to Learning
  - Machine Learning
  - \(\{\Phi|Un|\text{Semi}\}\)-supervised Learning

- Support Vector Machine (SVM)
  - Multi-class SVM

- Spectral Method
  - Semi-supervised Spectral Method

- Prospect and Conclusion
Machine Learning

Machine Learning: the study of computer algorithms that improve automatically through experience - Tom Mitchell 1997

Applications:

Data Mining to learn general rules
Bioinformatics
Information Extraction/Filtering

"... it(machine learning) will lead to appropriate, partial automation of every element of scientific method, from hypothesis generation to model construction to decisive experimentation."

Eric Mjolsness and Dennis DeCoste, Science 2001
Learning

- **Supervised Learning** - train labeled dataset, predict the class of unknown pattern
  - Regression - infinite types of labels
  - Classification (Pattern Recognition) - finite

- **Unsupervised Learning** - no labeled data
  - Clustering - group data into several 'clusters'

- **Semi-supervised Learning** - labeled data scarcely available, unlabeled abundantly present
  - Application - Webpage categorization, Gene expression analysis
Kernel Matrix

- Given a dataset $X = \{(x_i, y_i)\}_{i=1}^n$
  - $x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1, \text{unknown}\}$

- Construct a symmetric $n \times n$ matrix $K$, $K_{ij}$ measures pairwise 'relation' between $x_i, x_j$
  - e.g. $K_{ij} = f(x_i, x_j) = \exp \left( -\frac{||x_i - x_j||^2}{2\sigma^2} \right)$

- If $K$ (Semi)Positive Definite $\rightarrow$ Kernel Matrix
  - $f(x_i, x_j) = \phi(x_i)^T \phi(x_j) = < \phi(x_i), \phi(x_j) >$
  where $<, >$ is the inner product, $\phi(x_i) \in \mathbb{R}^q (q \geq d)$
  is a nonlinear mapping of $x_i$
### IQ Quiz - Which is the odd one out?

<table>
<thead>
<tr>
<th></th>
<th>Symmetry</th>
<th>Curvature</th>
<th>Aesthetic appealing</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>2</td>
<td>3</td>
<td>9.9</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>2</td>
<td>6.5</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
<td>4.4</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
<td>9.5</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
<td>9.2</td>
</tr>
</tbody>
</table>
Using \( K_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \), \( 2\sigma^2 = 1 \)

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>1.0000</td>
<td>0.000</td>
<td>0.0000</td>
<td>0.3135</td>
<td>0.2254</td>
</tr>
<tr>
<td>D</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0016</td>
<td>0.0000</td>
<td>0.0003</td>
</tr>
<tr>
<td>C</td>
<td>0.0000</td>
<td>0.0016</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>B</td>
<td>0.3135</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.9139</td>
</tr>
<tr>
<td>A</td>
<td>0.2254</td>
<td>0.0003</td>
<td>0.0000</td>
<td>0.9139</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
SVM-Large Margin

- a separating plane $x^T w + b = 0$
  
  s.t \[
  \begin{align*}
  x_i^T w + b &\geq +1 \text{ if } y_i = +1 \\
  x_i^T w + b &\leq -1 \text{ if } y_i = -1
  \end{align*}
  \]
  $\rightarrow y_i(x_i^T w + b) \geq 1$

- maximize the margin $= \frac{2}{\sqrt{w^T w}} = \frac{2}{\|w\|_2}$
Separable Case: \( y_i (x_i^T w + b) - 1 \geq 0, \quad \forall i \)
\[ \rightarrow \min \frac{w^T w}{2} \]

Nonseparable: permitting \( y_i (x_i^T w + b) - 1 < 0 \)
\[ \rightarrow \min \{ \frac{w^T w}{2} + C \sum \max \{0, 1 - y_i (x_i^T w + b)\} \}, \quad C \geq 0 \]

Obtain the following Quadratic Program by Lagrangian Optimization
\[ \rightarrow \max \{-\frac{1}{2} \sum_{i,j} y_i \alpha_i < x_i, x_j > \alpha_j y_j + \sum_i \alpha_i\} \]
\[ \text{s.t. } \sum_i \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad C \rightarrow +\infty \text{ if separable} \]

Solved by standard QP routines, or faster algorithms utilizing the special linear constraints: working set selection, steepest feasible descent, shrink, SMO ...
plug-in \( < \phi(x_i), \phi(x_j) > \)
\[
\max\{ -\frac{1}{2} \sum_{i,j} y_i \alpha_i x_i^T x_j + \alpha_i y_i + \sum_i \alpha_i \}
\]

What’s the benefit?
- Hyperplane in hyperspace
- \( \phi(x_i) \) more separable, sparsely scattering
- Less relying on Penalty Constant \( C \)

succinctly denoting:
\[
\max\{ -\frac{1}{2} \beta^T K \beta + \alpha^T e \}
\]
\[
\text{s.t. } \beta^T e = 0, \beta_i = \alpha_i y_i, \alpha_i \in [0, C] \text{ where } e \text{ is a vector of 1s} \]
SVM - Solution & Decision Function

Solution of $\alpha$

$$\begin{cases} 
\text{if } y_i(x_i^T w + b) - 1 > 0, \alpha_i = 0 \\
\text{if } y_i(x_i^T w + b) - 1 = 0, \alpha_i \in [0, C] \\
\text{if } y_i(x_i^T w + b) - 1 < 0, \alpha_i = C
\end{cases}$$

Support Vectors

Decision Function

$$h(x) = x^T w + b = \sum_{i=0}^{\text{num of SVs}} \alpha_i y_i < x_i, x > + b$$

$$y(x) = \text{sign}(h(x))$$
Gaussian Kernel $\exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$

A nonseparable case
Multi-class SVM by Output Coding

Output Code (Outcode)

<table>
<thead>
<tr>
<th>1001</th>
<th>0001</th>
<th>0101</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0000</td>
<td>0100</td>
</tr>
<tr>
<td>1010</td>
<td>0010</td>
<td>0110</td>
</tr>
</tbody>
</table>

ECOC (Error Correcting Output Code)

- many \((\leq 2^n)\) subspaces possible
- each subspace is encoded by an n-bit code
- each class is a union of several subspaces

(decoding-dependent!)

- Error Correcting - flipping several bits does not flop the overall result
Given an 3-class learning task

Representing each class by an 3-bit code, thus form an $3 \times 3$ code matrix $M$

Train 3 SVMs, the $i$–th SVM splits the data w.r.t the $i$–th column of $M$
- One-Per-Class (OPC) Code (one against others)
  * e.g. one column of OPC $[1, 0, 0]^T$
- Pairwise Coupling (PWC) Code (one against one)
  * e.g. one column of PWC $[1, 0, \Phi]^T$

SVMs assign an 3-bit code to each new data, classify it to the class of the 'closest' code
- Unbalancedly train $m$ SVMs for a $m$-class task
- Weak Error Correcting ability (e.g. Region 110, 000, 101, 011)
- Evenly train $\frac{m(m-1)}{2}$ SVMs for an $m$-class task
- Correct 110, 101, 011 to the right class
Hadamard Code


- Exclude the first column, \( m - 1 \) SVMs are trained.
- Strong Error Correcting ability: large row separation, columns pairwise orthogonal
Table 1: Experimental results using different output codes based on the SVMs with Gaussian RBF kernels

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Hadamard</th>
<th>One-per-Class</th>
<th>Random Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R_{emp}$</td>
<td>$R_{gen}$</td>
<td>$R_{emp}$</td>
</tr>
<tr>
<td>dermatology</td>
<td>0.0000</td>
<td><strong>0.0121</strong></td>
<td>0.0037</td>
</tr>
<tr>
<td>glass</td>
<td>0.0498</td>
<td><strong>0.1184</strong></td>
<td>0.0265</td>
</tr>
<tr>
<td>ecoli</td>
<td>0.0179</td>
<td><strong>0.0159</strong></td>
<td>0.0365</td>
</tr>
<tr>
<td>vowel</td>
<td>0.0000</td>
<td><strong>0.0013</strong></td>
<td>0.0000</td>
</tr>
<tr>
<td>yeast</td>
<td>0.0745</td>
<td><strong>0.0685</strong></td>
<td>0.2652</td>
</tr>
<tr>
<td>letter</td>
<td>0.0000</td>
<td><strong>0.0171</strong></td>
<td>0.0089</td>
</tr>
</tbody>
</table>
Summary of ECOC SVM

- Code is important
  - Hadamard Code: Economic number of SVMs, balanced split, but hard to train
  - PWC: Train many SVMs, balanced split, easy to train a single SVM
  - OPC: Unbalanced Split, hard to train, easy to understand

- Decoding Method is Crucial!
  - Hamming Decoding (Majority Vote) Poisons OPC
  - Weighted Decoding Scheme may help
    
e.g. class probability $\propto |h(x)|^{-1} = |x^Tw + b|^{-1}$
Given a dataset $X = \{(x_i, y_i)\}_{i=1}^n$
- $x_i \in \mathbb{R}^d$, $y_i$ is unknown if unsupervised

Construct a symmetric $n \times n$ matrix $K$, $K_{ij}$ measures pairwise 'relation' (similarity) between $x_i, x_j$
- e.g $K_{ij} = f(x_i, x_j) = \exp\left(-\frac{|x_i-x_j|^2}{2\sigma^2}\right)$

Similarity Matrix
- from Kernel Matrix
- from user-defined metric/nonmetric
- from input space / feature space
A Similarity Matrix $K \iff$ an undirected weighted graph $G = (V, E)$

- $V$: vertices represent data points
- $E$: edges weighted by the pairwise measurement $k_{ij}

bipartition the graph into two disjointed node sets $A, B$, assigning labels $y_i = +1$ if $x_i \in A$

$y_i = -1$ if $x_i \in B$

w.r.t the following minimization criteria:

$$\min C(A, B) = \sum_{i \neq j} K_{ij}$$

$$\Rightarrow \min \frac{1}{4} \sum (y_i - y_j)^2 K_{ij}$$
\[
\min \frac{1}{4} \sum (y_i - y_j)^2 K_{ij}
\]

- \( y = \pm e \) → trivial minimum 0
- evenly split → \( y^T e = 0 \) - but NP-hard!
  
  \[
  \rightarrow \min \frac{1}{2} (\sum_{i,j} K_{ij} - y^T K y) = \min \frac{1}{2} y^T L y, \quad L = D - K 
  \]

  \[D = \text{diag}(d_1, \ldots, d_n) \text{ with } d_i = \sum_{j=1}^n K_{ij}, \quad L \text{ is SPD}\]

- Still NP-hard! Approximated by Spectral
- Relaxing \( y \in \mathbb{R}^n \), restricting \( y^T y = n \)
  
  \[y^T L y \propto \frac{y^T L y}{y^T y} \geq \lambda_2 = \frac{v_2^T L v_2}{v_2^T v_2} > \lambda_1 = 0 = e^T L e\]

  \( \lambda_2 \) is the smallest nonzero eigenvalue of \( L \), \( v_2 \) is the corresponding eigenvector → the \( y \) we want

  \( \lambda_2 \) is the approximated minimum
Spectral Clustering

- threshold = median($v_2$) → a balanced split
  Any $p$-percentile cut works well if a split with ratio $p$ required

- ($v_{2i}, \ldots, v_{ni}$) → the new embedding of $x_i$
  Any clustering algorithm can work on the new embedding

- More about Spectral ...
  application: molecular physics, computer vision, high
  performance computing, data clustering
  siblings: Average cut, Normalized cut, Ratio Cut, Min-max
  Cut, Recursive cut ...
  relatives: PCA, Kernel PCA, LLE, MDS, Random Walk ...
Spectral Clustering - Examples

Some Challenge Clustering Problems
Some Semi-supervised Approaches

- **Co-training**
  Data consist of two feature views -> find latent label from unlabeled data [Blum(1998)]

- **Semi-supervised SVM**
  SVM variant by LP optimization upon the whole dataset [Bennett(1998)]
  Transductive SVM (TSVM): iteratively recruit new data, update SVM [Joachim(1999)]

- **Adapting Kernel Matrix**
  kernel alignment: Modify the metric of SVM kernel by Spectrum [Cristianini(2002)]
Given two labeled sets: positive $s$ and negative $t$

Add a supersource $S$ and a supersink $T$ with $w = +\infty \rightarrow$ S-T cut problem
Via Max-Flow: Augmenting Path, Preflow Push ... [Blum(2001)]

Advantages:
- Many algorithms and implementing variants available
- Efficient, bounded by $O(|V||E|)$

Disadvantages:
- Sensitive to outliers
- Hard to get a balanced cut
- Discrete output ($y_i = \pm 1$), not a probability measure
Via Semi-supervised Spectral Methods

Advantages:
- Systematically decomposing a \((n + 2) \times (n + 2)\) matrix
- A balanced cut, outliers insensitive
- Continuous Output, \(v_2\) indicating probability/ordering
- Supersource/sink coefficient \(w\) can be adapted to the different confidence of labeled data
- Eig-Solver Complexity: \(O(|V||E|)\). Fastest Lanczos method \(O(|V|+|E|)\)

Disadvantages:
- Numerically instability of eigen-decomposition
- Fast eigen-decomposition routine needed for large \(L\)
Spectral v.s. Max Flow (cont.)

(Spectral and SVM, TSVM)
Experiment on Ionosphere data from the UCI Repository

(Spectral and max-flow algo)

Adapting Kernel-Based Methods: Multi-class SVM and Semi-Supervised Spectral Analysis – p.29/31
Conclusion and Future Work

Kernel-based Methods
- SVM and Spectral Clustering – is there any relation?
- ECOC SVM – looking for a better code

Semi-supervised Learning
- Extend spectral methods – building an application
- Many novel approaches come out – more survey ...
Acknowledgement

- Special Thanks to Supervisors!
- Thanks to Thesis Committee
- Thanks to Friends and Audience.
- Please Do not hesitate to give me feedback or raise your questions!