Semi-Supervised Learning Based on Manifold in BCI

Ji-Ying Zhong, Xu Lei, and De-Zhong Yao

Abstract—A Laplacian support vector machine (LapSVM) algorithm, a semi-supervised learning based on manifold, is introduced to brain-computer interface (BCI) to raise the classification precision and reduce the subjects’ training complexity. The data are collected from three subjects in a three-task mental imagery experiment. LapSVM and transductive SVM (TSVM) are trained with a few labeled samples and a large number of unlabeled samples. The results confirm that LapSVM has a much better classification than TSVM.

Index Terms—Brain-computer interface, manifold learning, semi-supervised learning, support vector machine.

1. Introduction

Brain-computer interface (BCI) provides a channel to translate human’s thinking activity to external control signal. It is closely linked with many disciplines including brain science and cognition nerve science, psychology, signal processing, and so on. A BCI system includes several modules: input module, signal processing module, and signal transform module. The input module generates and detects the brain signal which represents a kind of thinking activity, and describes the feature signal with feature parameter. For making sure of corresponding thinking activity, and intended behavior. References [8] and [9] applied a self-training semi-supervised support vector machine algorithm in BCI. The algorithm converges fast and has low computational burden. But like many other semi-supervised learning, it easily finds local minimum, and often suffers from an inability to deal with a large number of unlabeled examples.

The signal processing module is an important part of a BCI system. It usually consists of feature estimation and pattern classification. The effectiveness of classification concerns not only the accuracy and training effort, but also the ability of real time of a BCI system. At present, main classification algorithms can be categorized as follows.

Supervised learning: it only uses labeled samples to train the classifier and ignores unlabeled samples. References [1] and [2] used Fisher linear classifier, SVM, and neural network to classify the samples in BCI. But usually most samples are unlabeled, so it is a huge project to label samples.

Non-supervised learning: the non-supervised learning is also called clustering. All the samples are unlabeled; the samples need to be labeled by the learning arithmetic automatic. The reported methods include k-NN[1], Bayes classification[4], and HMM[3]. This kind of learning arithmetic affects the classification accuracy in some extent for ignoring the available prior information.

Semi-supervised learning: it exploits a great deal of unlabeled samples with the available labeled samples to train the classifier, such as the transductive SVM which was utilized to classify the data in BCI competition with satisfied result[6]. Reference [7] applied reinforcement learning to find the functional mapping between neural activity and intended behavior. References [8] and [9] applied transductive SVM to classify the data in BCI competition with satisfied result[6]. Reference [7] applied reinforcement learning to find the functional mapping between neural activity and intended behavior. References [8] and [9] applied reinforcement learning to find the functional mapping between neural activity and intended behavior. References [8] and [9] applied reinforcement learning to find the functional mapping between neural activity and intended behavior. References [8] and [9] applied reinforcement learning to find the functional mapping between neural activity and intended behavior.

2. Methods

2.1 Semi-Supervised Learning Framework

Before detailing LapSVM, we introduce the semi-
supervised learning framework.

Suppose there is a data set of \( l \) labeled samples \( \{x_i, y_i\}_{i=1}^{l} \) and a set of \( u \) unlabeled samples \( \{x_i\}_{i=l+1}^{l+u} \), where \( x_i \in \mathbb{R}^N \) and \( y_i \in \{-1, 1\} \). Combined with unlabeled examples, we have the following regularization framework:

\[
 f^*(x) = \arg \min_{f \in H} \int_{x \in Y} V(y, f(x))dP(x, y) + \gamma_A \|f\|_K^2 + \gamma_y \|f\|_Y^2 \tag{1}
\]

where

\[
 \int_{x \in Y} V(y, f(x))dP(x, y)
\]

is an expected risk, \( V(y, f(x)) \) is a loss function, \( P(x, y) \) is the joint distribution; the function \( f^*(x) \) lies in a bounded convex subset of RKHS (reproducing kernel Hilbert space) \( H_K \) defined by a positive definite kernel function \( K \), which satisfies the Mercer condition; \( \|f\|_K^2 \) is the traditional Tikhonov regularization term in RHKS, and \( \|f\|_Y^2 \) is the regularization term based on manifold\(^{[10]} \); \( \gamma_A \) and \( \gamma_y \) are the parameters which control the balance of these two terms.

2.2 Graph Theory for Regularization Based on Manifold

To use the manifold regularization term, a graph based on labeled and unlabeled data point is built, and a weighted neighborhood graph \( G=(V, E) \) is formed. The vertex set \( V \) of the graph can be defined on the training set, including both labeled and unlabeled examples; and the edge set \( E \) contains the pairs of neighboring vertices \((x_i, y_j)\). After this, the relation between the data and the neighborhood graph can be established and a sparse neighboring matrix\(^{[12]} \) is generated as follows.

Step 1: use \( k \) nearest neighbors \((k-NN)\) to construct a neighborhood graph including both labeled and unlabeled examples.

Step 2: define weight value. If \( x_i \) is among \( k \) nearest neighbors of \( x_j \), the weight value is

\[
 W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{4t}\right)
\]

else \( W_{ij} = 0 \). This step forms the adjacency matrix \( W \) of the graph.

Step 3: build Laplacian matrix \( L = D - W \), where the diagonal matrix \( D \) satisfies \( D_{ii} = d_i \), and

\[
 d_i = \sum_{j=1}^{l+u} W_{ij}
\]

is the degree of vertex \( x_i \).

After defining the weighted neighborhood graph, we can formulate the following regularization term which includes the geometric information of both labeled and unlabeled data:

\[
 ||f||_Y^2 = \sum_{i=1}^{l+u} \frac{1}{W_{ii}} \|f(x_i)\|^2 = \mathbf{f}^T \mathbf{L} \mathbf{f}
\]

where \( f = (f(x_1), f(x_2), \ldots, f(x_{l+u})) \).

The intrinsic geometry of the data and the local structure of graph or manifold can be obtained by minimizing (2). The smoothness criterion makes the learned function do not change too much from nearby samples. If \( x_i \) and \( x_j \) have small distance, the learned function \( f(x_i) \) and \( f(x_j) \) should be approximate equal due to the large graph weight \( W_{ij} \).

Ultimately, we can generate the optimization function:

\[
 f^*(x) = \arg \min_{f \in H} \int_{x \in Y} V(y, f(x))dP(x, y) + \gamma_A \|f\|_K^2 + \gamma_y \|f\|_Y^2 \tag{3}
\]

By the representer theorem, (3) can be given by

\[
 f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x) \tag{4}
\]

2.3 Laplacian Support Vector Machine

Laplacian support vector machine (LapSVM) solves the optimization problem in (3) with the soft margin loss function defined as:

\[
 V(y, f(x)) = \max(0, 1 - y_i f(x_i)) \quad y_i \in \{-1, 1\}
\]

By introducing slack variables and using standard Lagrange multiplier techniques used for deriving SVM, we arrive at the following quadratic program in \( l \) dual variables \( \beta = [\beta_1, \beta_2, \ldots, \beta_l] \):

\[
 \beta^* = \max_{\beta \in \mathbb{R}^l} \sum_{i=1}^{l} \beta_i - \frac{1}{2} \beta^T Q \beta \quad \text{subject to} \quad \sum_{i=1}^{l} y_i \beta_i = 0, \quad 0 \leq \beta \leq \frac{1}{l}, \quad i = 1, 2, \ldots, l \tag{5}
\]

In (5),

\[
 Q = Y J K \left( 2\gamma_A I + 2\frac{\gamma_y}{(u+l)^2} L \right)^{-1} J^T Y
\]

where \( Y \) is the diagonal matrix \( Y_{ii} = y_i \), \( K \) is the Gram matrix over both the labeled and the unlabeled data; \( L \) is the data adjacency graph Laplacian; \( I \) is unit matrix ; and \( J \) is a \( l \times (l+u) \) matrix given by \( J_{ii} = 1 \) if \( i=j \), and \( J_{ij} = 0 \) otherwise. To obtain the optimal expansion coefficient vector \( \alpha^* \in \mathbb{R}^{l+u} \), one has to solve the following linear system after solving the quadratic program above:

\[
 \alpha^* = \left( 2\gamma_A J + 2\frac{\gamma_y}{(u+l)^2} L \right)^{-1} J^T Y \beta^* \tag{6}
\]

Detailed deduction of the algorithm can be found in \([10],[11]\), and \([13]\). In our work, cross validation is adopted to select the parameters.

2.4 LapSVM Algorithm Procedure

Suppose there is a data set of \( l \) labeled samples
\{x_i, y_i\}_{i=1}^{l+u} \text{ and a data set of } u \text{ unlabeled samples } \{x_i\}_{i=l+1}^{l+u}. \text{ The LapSVM algorithm is implemented in following procedures.}

Step 1: construct data adjacency graph with \((l+u)\) nodes using \(k\) nearest neighbors, choose edge weights \(W_{ij}\) with heat kernel weights:

\[ W_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{4\tau}\right). \]

Step 2: choose a kernel function \( K(x, y) \), compute the Gram matrix \( K_{ij} = K(x_i, y_j) \).

Step 3: compute graph Laplacian matrix: \( \mathbf{L} = \mathbf{D} - \mathbf{W} \).

Step 4: choose \( \gamma_A \) and \( \gamma_B \).

Step 5: compute \( \alpha^* \) by using (7).

Step 6: output function \( f^*(x) = \sum_{i=1}^{l+u} \alpha^*_i K(x_i, x) \).

In this work, LapSVM is evaluated in comparison with TSVM for the classification of mental imagery tasks in BCI system. The details of TSVM can be found in [6].

3. Experiments Results

3.1 BCI Experimental Data

The EEG data sets used in this work come from the IDIAP Research Institute in BCI Competition III. (http://ida.first.fhg.de/projects/bci/competition_iii/desc_V.htm). EEG signals were recorded with a Biosemi system using a cap with 32 integrated electrodes located at standard positions of the International 10-20 System. The sampling rate was 512 Hz. This dataset contains data from 3 normal subjects during 4 non-feedback sessions. In experiment, the subjects sat in a normal chair, relaxed arms resting on their legs, and performed 3 tasks: imagination of repetitive self-paced left hand movements, imagination of repetitive self-paced right hand movements, and generation of words beginning with the same random letter. All 4 sessions of a given subject were acquired on the same day, each lasting 4 minutes with 5 minutes to 10 minutes breaks between them. The subject performed a given task for about 15 seconds and then switched randomly to another task at the operator’s request. EEG data was not split into trials since the subjects were continuously performing any of the mental tasks.

The raw EEG potentials are first spatially filtered by means of a surface Laplacian. Then, the power spectral density (PSD) of every 62.5 ms in the band 8 Hz to 30 Hz was estimated over the last second of data with a frequency resolution of 2 Hz for the 8 centro-parietal channels C3, Cz, C4, CP1, CP2, P3, Pz, and P4. As a result, an EEG sample is a 96-dimensional vector (8 channels time 12 frequency components). The data sets are listed in Table 1.

In order to analyze the data between sessions, each sample of the EEG data is normalized to an interval of [0,1].

Since the brain-computer interface system needs a response in every 0.5 s, the data over eight consecutive samples are averaged. Finally, we achieve the training sets and testing sets listed in Table 2[6].

3.2 Data Processing

A. Data Processing in a Session

For each subject, Table 3 lists the number of labeled examples for 3, 5, 8 and 10 fold cross validation. To \( n \) fold, the data are divided into \( n \) equal chunks for each session: one chunk forms the labeled examples and the remaining chunks form the unlabeled examples. Then, ten adjacency graphs are constructed and the results of the algorithms, LapSVM and TSVM, are compared. The mean performance of LapSVM and TSVM in predicting the labels of unlabeled examples according to 3, 5, 8 and 10 fold cross validation are listed in Table 4 to 7.

Table 1: EEG Data sets for classification in a session

<table>
<thead>
<tr>
<th>Subject</th>
<th>Classes</th>
<th>Dimension</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject A</td>
<td>3</td>
<td>96</td>
<td>348</td>
<td>347</td>
<td>356</td>
</tr>
<tr>
<td>Subject B</td>
<td>3</td>
<td>96</td>
<td>342</td>
<td>432</td>
<td>446</td>
</tr>
<tr>
<td>Subject C</td>
<td>4</td>
<td>102</td>
<td>432</td>
<td>432</td>
<td>432</td>
</tr>
</tbody>
</table>

Table 2: EEG Data sets for classification between sessions

<table>
<thead>
<tr>
<th>Subject</th>
<th>Classes</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject A</td>
<td>4</td>
<td>436</td>
<td>432</td>
<td>446</td>
</tr>
<tr>
<td>Subject B</td>
<td>3</td>
<td>434</td>
<td>432</td>
<td>434</td>
</tr>
<tr>
<td>Subject C</td>
<td>4</td>
<td>432</td>
<td>428</td>
<td>430</td>
</tr>
</tbody>
</table>

Table 3: The number of labeled examples \((n\text{-fold cross validation})\)

<table>
<thead>
<tr>
<th>Subject</th>
<th>n-fold cross</th>
<th>The number of labeled data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject A</td>
<td>n=10</td>
<td>348</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
<td>436</td>
</tr>
<tr>
<td></td>
<td>n=5</td>
<td>697</td>
</tr>
<tr>
<td></td>
<td>n=3</td>
<td>1162</td>
</tr>
<tr>
<td>Subject B</td>
<td>n=10</td>
<td>347</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
<td>434</td>
</tr>
<tr>
<td></td>
<td>n=5</td>
<td>694</td>
</tr>
<tr>
<td></td>
<td>n=3</td>
<td>1157</td>
</tr>
<tr>
<td>Subject C</td>
<td>n=10</td>
<td>342</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
<td>428</td>
</tr>
<tr>
<td></td>
<td>n=5</td>
<td>684</td>
</tr>
<tr>
<td></td>
<td>n=3</td>
<td>1141</td>
</tr>
</tbody>
</table>

Table 4: Classification accuracy in a session (10-fold cross validation)

<table>
<thead>
<tr>
<th>Method</th>
<th>Subject</th>
<th>Test precision in a session (%)</th>
<th>Average Channel c</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Session 1</td>
<td>Session 2</td>
</tr>
<tr>
<td>TSVM</td>
<td>Subject A</td>
<td>71.90</td>
<td>72.41</td>
</tr>
<tr>
<td>LapSVM</td>
<td>Subject B</td>
<td>60.27</td>
<td>64.58</td>
</tr>
<tr>
<td>TSVM</td>
<td>Subject C</td>
<td>52.87</td>
<td>59.50</td>
</tr>
</tbody>
</table>
Comparing the tables above, we can draw the conclusions as follows:

TSVM and LapSVM can perform better along with the increase of training data.

No matter the ratio of training set to the testing set is about 1:9, 1:7, 1:4 or 1:2, LapSVM performs better than TSVM. It means that no matter how many the labeled samples are, the accuracy of LapSVM is higher than TSVM's.

For LapSVM, when the training set is small, the test classification accuracy has a big difference for subject A, B, and C, but with increasing the size of the training set, the difference is becoming smaller and smaller. But for TSVM there is no such regularity.

The results in Table 6 for three subjects show the effect of increasing the size of the labeled examples. From Table 6, we find that when the labeled set size is increased nearly double of the original labeled set size, the mean performances of subjects A, B, and C can be increased by 5.91%, 7.89%, 9.01%, respectively.

### B. Data Processing between Sessions

Due to the non-stationary characteristics of EEG signals, the data distribution between sessions has great discrepancy. Hence if we train a classification with one session data and classify the other session data, the classification results will not be ideal.

In our work, Laplacian SVM is used to deal the problem mentioned above. The data sets are shown in Table 3. The training sets are regarded as labeled examples, and testing sets as unlabeled examples. The features are first extracted and several of them whose fisher ratios are bigger than those of the new data's features are selected. Then the data are classified with LapSVM algorithm. Table 8 shows the comparison results between LapSVM and TSVM.

As we can see from Table 8, the LapSVM can perform better than TSVM does, and the result reflects the fact that LapSVM can reduce the effect of EEG signal's non-stationary characteristics.

### 4. Conclusions

In this work, a semi-supervised learning framework is introduced based on manifold for pattern classification in BCI. This framework provides a basis for several algorithms including unsupervised, semi-supervised, and fully supervised learning. By using this framework, we have solved the following problems. 1) The training complexity of a BCI system is reduced through reducing the number of labeled examples. 2) Both labeled and unlabeled examples are used to train a classification. In this way, we can solve the classification problems and improve the classification accuracy by utilizing unlabeled data of EEG. 3) The local optimum problem with non-convex objective function such as TSVM is avoided since our objective function based on Laplacian is a convex function. 4) The effect of non-stationary of EEG signals can be reduced as seen from Table 8.

### References


Ji-Ying Zhong was born in Sichuan Province, China, in 1981. She received the B.S. degree from the Southwest University of Science and Technology, Mianyang, Sichuan, in 2006. She is now pursuing the M.S. degree with University of Electronic Science and Technology of China (UESTC), Chengdu. Her research interests include brain computer interface and pattern recognition.

Xu Lei was born in Chongqing city, China, in 1982. He received B.S. degree in Information and computational science from UESTC in 2005. He is now pursuing the Ph.D. degree in biomedical engineering with UESTC. His research interests include EEG classification, EEG inverse problem, and EEG/fMRI fusion.

De-Zhong Yao was born in Chongqing, China, 1965. He received the Ph.D. degree in applied geophysics from the Chengdu University of Technology, Chengdu, China, in 1991, and completed his postdoctoral fellowship in electromagnetic field with UESTC in 1993. He has been a faculty member since 1993, a professor since 1995, and the Dean of the School of Life Science and Technology, UESTC since 2001, the director of the Key Laboratory for NeuroInformation of Ministry of Education, since 2009. He was a visiting scholar with the University of Illinois at Chicago, USA, from September 1997 to August 1998, and a visiting professor with the McMaster University, Canada, from November 2000 to May 2001 and with the Aalborg University, Denmark, from November 2003 to February 2004. He has published more than 80 peer reviewed papers in international journals and conferences. His current research interests include EEG and fMRI with their applications in cognitive science and neurological problems.