Classifier Learning with A New Locality Regularization Method

Hui Xue[1] Songcan Chen[1*] Xiaoqin Zeng[2]

[1] *Computer Science & Engineering College, Nanjing University of Aeronautics & Astronautics, 210016*

[2] *Department of Computer Science & Engineering, HoHai University, 210098*

Nanjing, P.R. China

Abstract: It is well known that the generalization capability is one of the most important criterions to develop and evaluate a classifier for a given pattern classification problem. The localized generalization error model (R_{SM}) [2, 12] recently proposed by Ng et al. provides a more intuitive look at the generalization error. Although R_{SM} gives a brand-new method to promote the generalization performance, it is in nature equivalent to another type of regularization. In this paper, we first prove the essential relationship between R_{SM} and regularization, and demonstrate that the stochastic sensitivity measure in R_{SM} exactly corresponds to a regularizing term. Then, we develop a new generalization error bound from the regulation viewpoint, which is inspired by the proved relationship between R_{SM} and regularization. Moreover, we derive a new regularization method, called as locality regularization (LR), from the bound. Different from the existing regularization methods which *artificially and externally* append the regularizing term in order to smooth the solution, LR is *naturally and internally* deduced from the defined expected risk functional and calculated by employing locality information. Through combining with spectral graph theory, LR introduces the local structure information of the samples into the regularizing term and further improves the generalization capability. In contrast with R_{SM}, which is relatively

[*] Corresponding author: Tel: +86-25-84896481 Ext. 12106; Fax:+86-25-84498069; E-mail: s.chen@nuaa.edu.cn (S. Chen) xuehui@nuaa.edu.cn (H. Xue), and xzeng@hhu.edu.cn(X. Zeng)

sensitive to the different sampling of the samples, LR uses the discrete *k*-neighborhood rather than the common continuous *Q*-neighborhood in R_{SM} to differentiate the relative position of different training samples automatically and avoid the complex computation of *Q* for various classifiers. Furthermore, LR uses the regularization parameter to control the trade-off between the training accuracy and the classifier stability. Experimental results on artificial and real world problems show that LR yields better generalization capability than both R_{SM} and some traditional regularization methods.

Keywords: Localized generalization error model; Stochastic sensitivity measure; Locality regularization (LR); Classifier Learning; Pattern classification.

1. Introduction

A classifier design method is usually an algorithm that develops a classifier *f* to approximate an unknown input-output mapping function *F* from finitely available data, i.e. training samples. Once such a classifier has been elaborately designed, it can be used to predict the class labels corresponding to unseen samples. Hence, the goal of developing a good classifier is to ensure the high prediction accuracy, i.e. generalization capability, for future unseen data [1].

Specifically, for a given pattern classification problem, the training samples, i.e. a set of input-output pairs $T = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, are generated according to a fixed but unknown probability distribution $P(\boldsymbol{x})$, where y_i is the class label of the input $\boldsymbol{x}_i$. The classifier *f* can be developed by minimizing the empirical risk on the training samples

$$R_{emp} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i))^2 \quad (1)$$

The quality of *f* produced by a specific design method is measured by the discrepancy between the true output produced by the mapping function *F* and the estimated value produced by *f* for unseen samples. The expected value of the discrepancy is defined as the generalization error [2]

$$R_{gen} = \int_{\Pi \setminus T}(F(\boldsymbol{x}) - f(\boldsymbol{x}))^2 dP(\boldsymbol{x}) \quad (2)$$

where Π denotes the entire input space.

Many techniques have been proposed to improve generalization capability for the classifier design. Regularization method was presented originally by Tikhonov [3, 4] for solving ill-posed problem. The basic idea of regularization is to stabilize and smooth the solution by means of some auxiliary nonnegative functional that embeds prior information about the solution [5]. The quantity to be minimized in regularization method is the *Tikhonov* functional, including both the empirical risk functional and the regularizing term, which are connected with a regularization parameter. Through optimizing the parameter, a satisfactory balance can be achieved between the training accuracy (bias) and the classifier complexity (variance) to gain good generalization capability [6]. An alternative method to promote generalization performance is cross validation (CV). In CV, the training dataset is randomly split into *k* disjoint subsets. A classifier is trained for *k* times on stochastic *k* -1 subsets and a subset is left out as the validation set to be used for estimating the generalization error at the same time [7]. Finally, the classifier with the lowest average estimated risk is chosen. However, CV is heuristic and can not guarantee the classifier to have good generalization performance in every case [7, 8, 9]. Moreover, the computational cost of CV grows linearly

with the number of samples and often becomes intolerable for practical application [9]. In addition, early stopping is another widely used technique to improve the generalization capability because it is simple to understand and implement. Just as CV, it splits the training data into a training set and a validation set, and stops training as soon as some stopping criterions being achieved in the validation set [10,11]. However, the method is also heuristic in nature and it is easy to prematurely stop at the local minimum. How to choose a proper stopping criterion is a key issue in the technique. Furthermore, as an alternative major approach, Vapnik-Chervonenkis (VC) theory [1] provides analytical generalization bounds that can be used for estimating generalization error by defining a new measure of complexity, called as the VC-dimension, which coincides with the number of parameters for linear classifier [9]. However, VC-theory can not be rigorously applied to nonlinear classifiers, such as neural networks, where the VC-dimension can not be accurately estimated and the empirical risk can not be reliably minimized [8, 9].

Different from the above state-of-the-art methods, the localized generalization error model (R_{SM}) recently proposed by Ng et al. [2, 12] provides a new look at the generalization error, although it has not received sufficient attention yet. R_{SM} is illuminated by the classifiers such as support vector machine (SVM) [13], radial basis function neural network (RBFNN) [5] and multilayer perceptron neural network (MLPNN) [5], which are really local learning machines and consider the unseen samples close to the training samples more important [2]. Hence, in R_{SM}, the generalization error for unseen samples is bounded within a Q-neighborhood of the training samples using stochastic sensitivity measure [14, 15, 16], i.e. the expectation of the squared output perturbation. And the Q-neighborhood is designed to be

the regular shape, such as a hyper-square, a sphere or a rectangle. Accordingly, a model selection method based on R_{SM} has been presented to train RBFNN [2]. For a given threshold, the method selects the optimal classifier by maximizing the value of Q, assuming that the mean square error (MSE) of all samples within the union of all Q-neighborhoods, including the training and testing samples, is smaller than the threshold [2]. The resulting RBFNN has better testing accuracy, fewer hidden neurons and less training time than the ones trained respectively using multiple folds cross validation and sequential learning. In addition, R_{SM} can also be generalized to feature selection [17], active learning [18], multiple classifier systems [19], image classification [12], and so on.

Although R_{SM} seems to provide a brand-new method to promote the generalization performance, in this paper, we will first prove the important relationship between R_{SM} and regularization, i.e. it is in nature another type of regularization, and demonstrate that the stochastic sensitivity measure in R_{SM} exactly corresponds to a regularizing term. Moreover, R_{SM} controls the generalization error for unseen samples through maximizing the Q-neighborhoods of training samples, which requires the Q-neighborhoods share common shape. However, in practice, R_{SM} can guarantee neither unseen samples always residing within the limited union of Q-neighborhoods, nor thus the comparable generalization performance for these samples. In order to tackle the above problem, we further develop a new generalization error bound from the regulation viewpoint, which is inspired by the proved relationship between R_{SM} and regularization. Moreover, we derive a new regularization method, called as locality regularization (LR), from the bound. Instead of optimizing Q, LR seeks for the classifier function $f(\boldsymbol{x})$ which minimizes a *quasi-Tikhonov*

functional directly on the basis of selecting a proper regularization parameter, just as in traditional regularization [5]. However, different from the existing regularization methods [5, 6] which *artificially and externally* append the regularizing term in order to smooth the solution, LR is *naturally and internally* deduced from the defined expected risk functional following R_{SM}. Furthermore, LR calculates the regularizing term by employing locally variable k-neighborhood rather than the common continuous Q-neighborhood in R_{SM} such that it can not only differentiate the relative position of different training samples automatically and avoid the complex computation of Q for various classifiers, but also further improve the generalization capability. Therefore, LR, on one hand, has the common advantage of traditional regularization method that can achieve a trade-off between the training accuracy and the classifier stability [6], which leads to the resulting classifier more stable than R_{SM} in terms of the different sampling of the samples. On the other hand, thanks to the introduction of the local structure information of the samples into the regularizing term through combining with spectral graph theory [20], LR yields more likely better generalization performance than traditional regularization method.

The rest of this paper is organized as follows. Section 2 briefly reviews the basic theory of R_{SM}. In Section 3, we prove the relationship between R_{SM} and regularization, and give the corresponding deduction. The new generalization error bound and the corresponding regularization method LR are proposed in Section 4. Section 5 shows some experimental results to demonstrate the better generalization performance of our method. Some conclusions are given in Section 6.

2. The Localized Generalization Error Model (R_{SM})

Let $S_Q^{(b)}$ denotes the Q-neighborhood of a training sample $\boldsymbol{x}^{(b)}$. And $S_Q^{(b)}$ is defined as $S_Q^{(b)} = \left\{ \boldsymbol{x} = \boldsymbol{x}^{(b)} + \triangle \boldsymbol{x} \right\}$ that fulfils $0 < \left| \triangle x_i \right| \le Q, \forall i = 1, \cdots, N$, where N denotes the number of features of the training sample and $\triangle \boldsymbol{x} = (\triangle x_1, \cdots, \triangle x_N)$ [17]. Then S_Q denotes the union of all $S_Q^{(b)}$, called as Q-Union.

R_{SM} is defined as the generalization error for the unseen samples, i.e. expected risk, within the Q-Union. With probability $1-\eta$, we have

$$\begin{aligned} R_{SM}(Q) &= \int_{S_Q} (f(\boldsymbol{x}) - F(\boldsymbol{x}))^2 p(\boldsymbol{x}) d\boldsymbol{x} \\ &\le \frac{1}{n} \sum_{b=1}^{n} \int_{S_Q^{(b)}} (f(\boldsymbol{x}) - F(\boldsymbol{x}))^2 \frac{1}{(2Q)^N} d\boldsymbol{x} + \varepsilon \\ &\le \left[\sqrt{R_{emp}} + \sqrt{E_s((\triangle y)^2)} + A \right]^2 + \varepsilon \;=\; R_{SM}^*(Q) \end{aligned} \tag{3}$$

Following [2], the notations in (3) are explained as follows:

1) $R_{emp} = \frac{1}{n} \sum_{b=1}^{n} (err^{(b)})^2$ is the usual empirical risk, where $err^{(b)} = f(\boldsymbol{x}^{(b)}) - F(\boldsymbol{x}^{(b)})$.

2) $E_s((\triangle y)^2) = \frac{1}{n} \sum_{b=1}^{n} \int_{S_Q^{(b)}} (\triangle y)^2 p(\boldsymbol{x}) d\boldsymbol{x}$ is the stochastic sensitivity measure, where $\triangle y = f(\boldsymbol{x}) - f(\boldsymbol{x}^{(b)})$ is the output perturbation which measures the output difference between the training sample $\boldsymbol{x}^{(b)} \in T$ and unseen sample in its Q-neighborhood $\boldsymbol{x} = \boldsymbol{x}^{(b)} + \triangle \boldsymbol{x} \in S_Q^{(b)}$.

3) $A = (\max(F(\boldsymbol{x})) - \min(F(\boldsymbol{x}))), \varepsilon = B\sqrt{\ln \eta(-2n)}$, where $B = \max((f(\boldsymbol{x}) - F(\boldsymbol{x}))^2)$. Both A and ε are constants for a given training dataset and a pre-selected upper bound of the classifier output values.

For RBFNN, with probability $1-\eta$, we have

$$R_{SM}^{*} \approx \left(\sqrt{\frac{1}{3} Q^2 \sum_{j=1}^{M} \upsilon_j + \frac{0.2}{9} Q^4 N \sum_{j=1}^{M} \zeta_j} + \sqrt{R_{emp}} + \sqrt{A} \right)^2 + \varepsilon \tag{4}$$

where $\upsilon_j = \varphi_j \left(\sum_{i=1}^{N} \left(\sigma_{x_i}^2 + \left(\mu_{x_i} - u_{ji} \right)^2 \right) / v_j^4 \right)$, $\zeta_j = \varphi_j / v_j^4$, detailed in [2].

Therefore, for a given threshold ρ, i.e. $R_{SM}^{*} = \rho$, the value Q can be computed by solving the following quadratic equation [2]:

$$Q^4 \frac{0.2}{3} N \sum_{j=1}^{M} \zeta_j + Q^2 \sum_{j=1}^{M} \upsilon_j - 3\left(\sqrt{\rho - \varepsilon} - \sqrt{R_{emp}} - \sqrt{A} \right)^2 = 0 \tag{5}$$

There are maximum four solutions for the Equation (5) and the smallest real solution will be used as final result [2, 17].

In a word, for the trained classifier whose concrete functional form is known, one could compute the maximum value of Q and the value indicates the coverage of the unseen samples whose generalization error in MSE is less than ρ [2, 16]. So if two classifiers yield the same $R_{SM}^{*}(Q)$ with different Q values respectively, the one that yields the larger Q has better generalization performance.

3. The Relationship between R_{SM} and Regularization

In this section, we reveal that the R_{SM} is actually another type of regularization method and the stochastic sensitivity measure corresponds to a generalized regularizing term.

We assume that the unknown learning classifier function $f(\boldsymbol{x})$ belongs to a specified *reproducing kernel Hilbert space* (RKHS) $\boldsymbol{H}$ [21, 22]. Let us denote the reproducing kernel of a functional Hilbert space $\boldsymbol{H}$ by $K(\boldsymbol{x}, \boldsymbol{x}')$. We will employ the following kernel model

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) \tag{6}$$

where $\{\alpha_i\}_{i=1}^n$ are parameters to be estimated from the training examples.

Theorem 1. If $f(\boldsymbol{x})$ belongs to $\boldsymbol{H}$, then the stochastic sensitivity measure in R_{SM} is equivalent to a generalized regularizing term, i.e.

$$E_s((\triangle y)^2) = c\|\boldsymbol{\alpha}\|_{\tilde{\mathbf{K}}} \tag{7}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$, $\tilde{\mathbf{K}}$ is a symmetric and positive semi-definite matrix depending on the training samples $\boldsymbol{x}_i (i = 1, \cdots, n)$, and c is a constant.

Proof We apply (6) in the stochastic sensitivity measure. Note that, here we also define that the shape of Q-neighborhood is a hyper-square just as in [2] for convenience. Then we have

$$\begin{aligned}
E_s((\triangle y)^2) &= \frac{1}{n}\sum_{b=1}^{n} \int_{S_Q^{(b)}} (\triangle y)^2 p(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{n}\sum_{b=1}^{n} \int_{S_Q^{(b)}} \left[f(\boldsymbol{x}) - f(\boldsymbol{x}^{(b)}) \right]^2 \frac{1}{(2Q)^N} d\boldsymbol{x} \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \int_{-Q}^{Q} \left[\sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}^{(b)} + \triangle \boldsymbol{x}, \boldsymbol{x}_i) - \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}^{(b)}, \boldsymbol{x}_i) \right]^2 d(\triangle \boldsymbol{x}) \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \int_{-Q}^{Q} \left\{ \sum_{i=1}^{n} \alpha_i \left[K(\boldsymbol{x}^{(b)} + \triangle \boldsymbol{x}, \boldsymbol{x}_i) - K(\boldsymbol{x}^{(b)}, \boldsymbol{x}_i) \right] \right\}^2 d(\triangle \boldsymbol{x}) \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \boldsymbol{\alpha}^T \left\{ \int_{-Q}^{Q} \boldsymbol{K}_b(\triangle \boldsymbol{x}) \boldsymbol{K}_b(\triangle \boldsymbol{x})^T d(\triangle \boldsymbol{x}) \right\} \boldsymbol{\alpha}
\end{aligned} \tag{8}$$

where $\boldsymbol{K}_b(\triangle \boldsymbol{x}) = \left[K(\boldsymbol{x}^{(b)} + \triangle \boldsymbol{x}, \boldsymbol{x}_1) - K(\boldsymbol{x}^{(b)}, \boldsymbol{x}_1), \cdots, K(\boldsymbol{x}^{(b)} + \triangle \boldsymbol{x}, \boldsymbol{x}_n) - K(\boldsymbol{x}^{(b)}, \boldsymbol{x}_n) \right]^T$.

To simplify the expression, let

$$\bar{\boldsymbol{K}}^{(b)} = \int_{-Q}^{Q} \boldsymbol{K}_b(\triangle \boldsymbol{x}) \boldsymbol{K}_b(\triangle \boldsymbol{x})^T d(\triangle \boldsymbol{x}) \tag{9}$$

Obviously, $\bar{\boldsymbol{K}}^{(b)}$ is a symmetric and positive semi-definite matrix. And (8) can be rewritten as the following form of a generalized regularizing term

$$E_s((\triangle y)^2) = \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \boldsymbol{\alpha}^T \bar{\boldsymbol{K}}^{(b)} \boldsymbol{\alpha} = \frac{1}{(2Q)^N} \|\boldsymbol{\alpha}\|_{\bar{\boldsymbol{K}}}$$

where $\tilde{\boldsymbol{K}} = \frac{1}{n}\sum_{b=1}^{n} \bar{\boldsymbol{K}}^{(b)}$. Obviously $\tilde{\boldsymbol{K}}$ is also a symmetric and positive semi-definite matrix and $\|\bullet\|_{\tilde{K}}$ denotes the weighted norm whose norm weighting matrix is $\tilde{\boldsymbol{K}}$. Let $c = 1/(2Q)^N$. This proves the theorem. ■

Corollary 1. If the reproducing kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ is chosen as the linear kernel, i.e.

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x} \tag{10}$$

then

$$E_s((\triangle y)^2) = \frac{Q^2}{3}\|Df\|^2 \tag{11}$$

where D is a linear differential operator, defined as $D = \frac{\partial}{\partial \mathbf{x}}$.

Proof We apply (10) in the stochastic sensitivity measure. Then we have

$$\begin{aligned}
E_s((\triangle y)^2) &= \frac{1}{n}\sum_{b=1}^{n} \int_{S_Q^{(b)}} \left[f(\boldsymbol{x}) - f(\boldsymbol{x}^{(b)}) \right]^2 p(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \int_{-Q}^{Q} \left[\sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T (\boldsymbol{x}^{(b)} + \triangle \boldsymbol{x}) - \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \boldsymbol{x}^{(b)} \right]^2 d(\triangle \boldsymbol{x}) \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \int_{-Q}^{Q} \left\{ \sum_{i=1}^{n} \alpha_i (\boldsymbol{x}_i^T \triangle \boldsymbol{x}) \right\}^2 d(\triangle \boldsymbol{x}) \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \boldsymbol{\alpha}^T \left[\int_{-Q}^{Q} \begin{pmatrix} \boldsymbol{x}_1^T \triangle \boldsymbol{x} \\ \vdots \\ \boldsymbol{x}_n^T \triangle \boldsymbol{x} \end{pmatrix} \begin{pmatrix} \triangle \boldsymbol{x}^T \boldsymbol{x}_1 & \cdots & \triangle \boldsymbol{x}^T \boldsymbol{x}_n \end{pmatrix} d(\triangle \boldsymbol{x}) \right] \boldsymbol{\alpha} \\
&= \frac{1}{(2Q)^N n}\sum_{b=1}^{n} \boldsymbol{\alpha}^T \left[\int_{-Q}^{Q} \begin{pmatrix} \boldsymbol{x}_1^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_1^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_n^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_n \end{pmatrix} d(\triangle \boldsymbol{x}) \right] \boldsymbol{\alpha}
\end{aligned} \tag{12}$$

It is well known that the matrix-valued function integral is equivalent to the integral to each element of the matrix [23], i.e.

$$\int_{-Q}^{Q} A(\triangle \boldsymbol{x}) d(\triangle \boldsymbol{x}) = \left(\int_{-Q}^{Q} a_{ij}(\triangle \boldsymbol{x}) d(\triangle \boldsymbol{x}) \right) \tag{13}$$

where the integral denotes N-fold multiple integral.

Hence, we have

$$\int_{-Q}^{Q} \boldsymbol{x}_i^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_j d(\triangle \boldsymbol{x}) = \boldsymbol{x}_i^T \left[\int_{-Q}^{Q} \begin{pmatrix} \triangle x_1^{\,2} & \cdots & \triangle x_1 \triangle x_N \\ \vdots & \ddots & \vdots \\ \triangle x_N \triangle x_1 & \cdots & \triangle x_N^{\,2} \end{pmatrix} d(\triangle \boldsymbol{x}) \right] \boldsymbol{x}_j \tag{14}$$

We compute the integral to each element respectively, and obtain

$$\int_{-Q}^{Q} \triangle x_i \triangle x_j d(\triangle \boldsymbol{x}) = \begin{cases} (2Q)^{N-1} \int_{-Q}^{Q} (\triangle x_i)^2 d(\triangle x_i) = \dfrac{2^N Q^{N+2}}{3} & i = j \\ (2Q)^{N-2} \int_{-Q}^{Q} \triangle x_i d(\triangle x_i) \int_{-Q}^{Q} \triangle x_j d(\triangle x_j) = 0 & i \neq j \end{cases} \tag{15}$$

Therefore,

$$\int_{-Q}^{Q} \boldsymbol{x}_i^T \triangle \boldsymbol{x} \triangle \boldsymbol{x}^T \boldsymbol{x}_j d(\triangle \boldsymbol{x}) = \frac{2^N Q^{N+2}}{3} \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{16}$$

From the above deduction, we can obtain

$$E_s((\triangle y)^2) = \frac{Q^2}{3} \boldsymbol{\alpha}^T \begin{pmatrix} \boldsymbol{x}_1^T \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_1^T \boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_n^T \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_n^T \boldsymbol{x}_n \end{pmatrix} \boldsymbol{\alpha}$$

$$= \frac{Q^2}{3} \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i^T \bullet \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i$$

$$= \frac{Q^2}{3} \left\| \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i \right\|^2$$

$$= \frac{Q^2}{3} \| Df \|^2$$

where D is a linear differential operator, defined as $D = \dfrac{\partial}{\partial \mathbf{x}}$. ■

Equation (11) is actually the standard form of a regularizing term in Tikhonov regularization theory [5, 6]. Here we select the linear kernel as an example for convenience. The reproducing kernel can also be taken as different kernels, such as polynomial kernel and Gaussian kernel, and in terms of Theorem 1, we can draw similar conclusions corresponding to the different kernels.

In summary, R_{SM} belongs to the framework of the regularization method. However, different from traditional regularization which considers the smoothness of the solution from the global view, R_{SM} focuses on the smoothness in each local Q-neighborhood of the training samples. The stochastic sensitivity measure measures the expectation of the squares of output perturbations between the training samples and unseen samples in the corresponding Q-neighborhoods [2]. Therefore, it can penalize the large output perturbation and guarantee that similar inputs correspond to similar outputs. For considering generalization error in every local neighborhood, R_{SM} has much better generalization performance than some existing methods [2] and provides us a brand-new viewpoint to improve the regularization method.

4. A New Locality Regularization Method (LR)

R_{SM} selects the optimal classifier through maximizing the value of Q in the condition that the generalization error is smaller than a pre-selected threshold ρ. To obtain a high-order equation w.r.t. Q which is easy to solve, the concrete functional form of the trained classifier must be known and the shape of Q-neighborhood should be regular. This puts a severe limitation on the applicability of R_{SM}. Moreover, when unseen samples reside in the limited Q-Union, R_{SM} can ensure the good generalization capability for these samples. However, once unseen samples reside outside the union, it can not guarantee the comparable generalization performance again. In order to tackle the above problems, we further develop a new generalization error bound which is inspired by the proved relationship between R_{SM} and regularization. Furthermore, we derive a new regularization method, called as LR, from minimizing the bound. Instead of optimizing Q, we seek for the classifier function $f(\boldsymbol{x})$ which

minimizes a *quasi-Tikhonov functional* directly on the basis of selecting a proper regularization parameter.

In the framework of regularization, the trained classifier function is often required to belong to RKHS and has a general form [5] as follows

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$$

In fact, we only know the values of f at a finite number of points in practice. Furthermore, though R_{SM} can compute the value of Q by solving a quadratic equation for RBFNN, as we mentioned in Section 2, it is difficult for R_{SM} to obtain such equation for various classifiers. As a result, the computation of Q will become more complex and even unworkable. So, instead of searching the continuous Q neighborhood, we search the k nearest neighbors of every training sample $\boldsymbol{x}_i$. In other words, here we consider the discrete version of the problem. Combining with class label information, we can increase the value of k until a sample whose label is different from $\boldsymbol{x}_i$ will reside in the neighborhood. Thus the size of the neighborhood is variable to different training samples. If $\boldsymbol{x}_i$ is far away from the boundary between classes, the corresponding neighborhood is relatively larger and includes more samples within the same class. On the other hand, if $\boldsymbol{x}_i$ is near the boundary, the neighborhood is shrinkable to avoid including the samples with different class labels. Therefore, the value of k can differentiate the relative position of different training samples automatically and make for further improvement to generalization capability of the trained classifier.

One important principle of regularization is the smoothness of the solution, in the sense that similar inputs correspond to similar outputs [5, 6]. For classification problems, it means that if two samples are close to each other, they should share the same label. Therefore, it is

reasonable to calculate the regularizing term with locality preserving property. We construct a nearest neighbor graph *G* to model the locally geometrical structure of the samples. Let *S* be the weight matrix of *G*. A possible definition of *S* is as follows [24, 25]:

$$S_{ij}=\begin{cases} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{t}} & \text{if } \mathbf{x}_i \text{ is among the k nearest neighbors of } \mathbf{x}_j, \\ & \text{or } \mathbf{x}_j \text{ is among the k nearest neighbors of } \mathbf{x}_i; \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

where *t* is a suitable constant and the function $\exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/t)$ is called as heat kernel [26]. Note that, the above definition reflects the intrinsic manifold structure of the data space.

Recall that in R_{SM}, the generalization error is bounded in the *Q*-Union:

$$\begin{aligned} R_{SM}(Q) &= \int_{S_Q}(f(\boldsymbol{x})-F(\boldsymbol{x}))^2 p(\boldsymbol{x})d\boldsymbol{x} \\ &\le \left[\sqrt{R_{emp}}+\sqrt{E_s((\triangle y)^2)}+A\right]^2+\varepsilon \;=\; R^*_{SM}(Q) \end{aligned}$$

where *A* and ε are constants for a given classification problem.

Inspired by the relationship between R_{SM} and regularization, we further deduce a new generalization error bound as described below, which introduces the local structure information of the samples into the bound.

Theorem 2. (Generalization Error Bound) Let $\boldsymbol{H}$ denote the RKHS. For random probability distribution $P(\boldsymbol{x})$ generating the training input-output pairs $T=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^n$, with probability $1-\eta$, the generalization error of random classifier $f\in H$ is no more than

$$\underset{P}{err}(f)\le\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{\alpha}^T\hat{\mathbf{K}}_i-y_i\right]^2}+\lambda\sqrt{\boldsymbol{\alpha}^T\tilde{\mathbf{K}}'\boldsymbol{\alpha}}+A\right)^2+\varepsilon \tag{18}$$

where $\hat{\mathbf{K}}_i=\left[K(\mathbf{x}_1,\mathbf{x}_i),\cdots,K(\mathbf{x}_n,\mathbf{x}_i)\right]^T$, $\tilde{\mathbf{K}}'$ is a symmetric and positive semi-definite matrix depending on $\boldsymbol{x_i}(i=1,\cdots,n)$, and λ is the regularization parameter.

Proof As we mentioned above, firstly we employ the discrete *k*-neighborhood rather than

the continuous *Q*-neighborhood in the stochastic sensitivity measure. And by spectral graph theory [20], the distribution density $p(\boldsymbol{x})$ on the *k*-neighborhoods can be discretely estimated by the matrix *D*, where the *ij*th entry is defined as $d_{ij} = \frac{S_{ij}}{D_{ii}}$ and let $D_{ii} = \sum_{j=1}^{k} S_{ij}$. Hence, we have

$$
\begin{aligned}
E_s((\triangle y)^2) &= \frac{1}{n}\sum_{b=1}^{n} \int_{S_Q^{(b)}} \left[f(\boldsymbol{x}) - f(\boldsymbol{x}^{(b)}) \right]^2 p(\boldsymbol{x}) d\boldsymbol{x} \\
&\doteq \frac{1}{n}\sum_{b=1}^{n} \left\{ \sum_{j=1}^{k} \left[f(\boldsymbol{x}_j) - f(\boldsymbol{x}^{(b)}) \right]^2 \frac{S_{bj}}{D_{bb}} \right\} \\
&= \frac{1}{n}\sum_{b=1}^{n} \left\{ \sum_{j=1}^{k} \left[\sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}^{(b)}) \right]^2 \frac{S_{bj}}{D_{bb}} \right\} \\
&= \frac{1}{n}\sum_{b=1}^{n} \left\{ \sum_{j=1}^{k} \left\{ \sum_{i=1}^{n} \alpha_i \left[K(\boldsymbol{x}_i, \boldsymbol{x}_j) - K(\boldsymbol{x}_i, \boldsymbol{x}^{(b)}) \right] \right\}^2 \frac{S_{bj}}{D_{bb}} \right\} \\
&= \frac{1}{n}\sum_{b=1}^{n} \boldsymbol{\alpha}^T \left[\sum_{j=1}^{k} \boldsymbol{K}_b'(\triangle \boldsymbol{x}) \boldsymbol{K}_b'(\triangle \boldsymbol{x})^T \frac{S_{bj}}{D_{bb}} \right] \boldsymbol{\alpha}
\end{aligned} \tag{19}
$$

where $\boldsymbol{K}_b'(\triangle \boldsymbol{x}) = \left[K(\boldsymbol{x}_1, \boldsymbol{x}_j) - K(\boldsymbol{x}_1, \boldsymbol{x}^{(b)}), \cdots, K(\boldsymbol{x}_n, \boldsymbol{x}_j) - K(\boldsymbol{x}_n, \boldsymbol{x}^{(b)}) \right]^T$.

As Theorem 1, we let

$$
\bar{\boldsymbol{K}}'^{(b)} = \sum_{j=1}^{k} \boldsymbol{K}_b'(\triangle \boldsymbol{x}) \boldsymbol{K}_b'(\triangle \boldsymbol{x})^T \frac{S_{bj}}{D_{bb}} \tag{20}
$$

and

$$
\tilde{\boldsymbol{K}}' = \frac{1}{n}\sum_{b=1}^{n} \bar{\boldsymbol{K}}'^{(b)} \tag{21}
$$

then

$$
E_s((\triangle y)^2) \doteq \boldsymbol{\alpha}^T \tilde{\mathbf{K}}' \boldsymbol{\alpha} \tag{22}
$$

In the regularization theory, the trade-off between the bias and variance of the classifier can be achieved through adjusting the regularization parameter λ [5]. So here we introduce

λ into the bound in order to make it more flexible. Referring to the bound in R_{SM}, by the Hoeffding's inequality [27], with probability $1-\eta$, we have the new generalization error bound of f

$$\underset{P}{err}(f) \le \left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{\alpha}^T\hat{\mathbf{K}}_i - y_i\right]^2} + \lambda\sqrt{\boldsymbol{\alpha}^T\tilde{\mathbf{K}}'\boldsymbol{\alpha}} + A \right)^2 + \varepsilon$$

where $\hat{\mathbf{K}}_i$ and λ is defined as before. ■

Consequently, we derive our new regularization method, LR, from Theorem 2 through minimizing the generalization error bound, which is equivalent to minimizing

$$\min_{\boldsymbol{\alpha}} \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\boldsymbol{\alpha}^T\hat{\mathbf{K}}_i - y_i\right]^2} + \lambda\sqrt{\boldsymbol{\alpha}^T\tilde{\mathbf{K}}'\boldsymbol{\alpha}} \qquad (23)$$

Here it is necessary to point out that different in the two aspects from the existing regularization methods, one is its derivation, LR is actually *naturally* deduced from the defined expected risk functional and calculated by employing locality information. The other aspect is its form, as shown in (23), though a bit formal difference from the standard regularization formulation, LR still embodies the similar principle to stabilize and smooth the solution, and further calibrate its generalization ability. Hence, we abuse the terminology, i.e. *quasi-Tikhonov functional*, here to denote Equation (23). Similarly, the influence of the regularizing term on the final solution is controlled by λ. Through selecting a proper value of λ, we can obtain an optimal solution of Equation (23) and the resulting classifier more likely yields better generalization capability than R_{SM} which only considers the generalization error in the limited Q-Union. However, just as in the traditional regularization methods, the selection of λ in LR is difficult and time-consuming. Fortunately, some improvements have been made in simplifying the computation of λ [28, 29]. These techniques can also be

employed in LR.

5. Experiments

In this section, we test the effectiveness of LR for classification. First of all, artificial problems are studied to evaluate the generalization capability between LR and R_{SM} in terms of the different sampling of the testing samples. Then the classification experiments on two real world databases are performed. The first one is the partial UCI database, and the second one is the Benchmark database[1] used in [30]. In all the experiments, RBFNN is still used to demonstrate the use of R_{SM} and the given threshold ρ is also selected as 0.25, just as in [2]. In order to correspond to RBFNN, we choose the RBF kernel as the reproducing kernel in LR. And following [5], we select the regularization parameter λ by cross validation in these experiments. Furthermore, throughout the experiments, we adopt the restarted Fletcher-Reeves conjugate gradient algorithm [31] to solve the optimization problem (23) in LR.

5.1 Artificial problems

In this subsection, we compare LR and R_{SM} on two different artificial databases of binary class problems, corresponding to the normal distribution and uniform distribution respectively. The objective of these experiments is to evaluate the generalization performances of LR and R_{SM} in terms of the different distributions of the databases as well as testing samples.

[1] available at http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

5.1.1 Normal distribution

Normal distribution is the most common distribution of the samples in real world problems. Many learning theory and algorithms are derived on the premise that the patterns follow the normal distribution. Therefore, here we firstly present an artificial database in normal distribution. The database contains two datasets in which each class contains 150 samples and the samples are generated randomly from the bivariate normal distribution. The means in the two classes are [0, 0] and [1.75, 1.75] respectively, and the variance is uniform diag[1, 1]. In the first dataset, we stochastically select 135 samples in respective classes to combine the training set and take the remaining 30 samples as the testing set, in order to guarantee all the testing samples reside in the *Q*-Union produced by the 270 training samples as far as possible. In contrast, in the second dataset, we select 15 samples in each class as the training set and consequently most of the remaining 270 testing samples reside outside the *Q*-Union. Figure 1 and Figure 2 respectively illustrate the different distributions of the testing samples in the two datasets, where the black squares around the training samples denote the corresponding *Q*-Union.

Table 1 and Table 2 show the training and testing accuracies of the two methods on the respective datasets. Compared the two tables, it is obvious that the testing accuracies of LR are much better than that of R_{SM} in the both datasets. And the gap between the classification accuracies is much wider in the second dataset than in the first dataset. This fact validates that when the testing samples reside outside the *Q*-Union, R_{SM} can not guarantee the comparable generalization performance due to the disadvantage of the construction of the regular *Q*-Union.

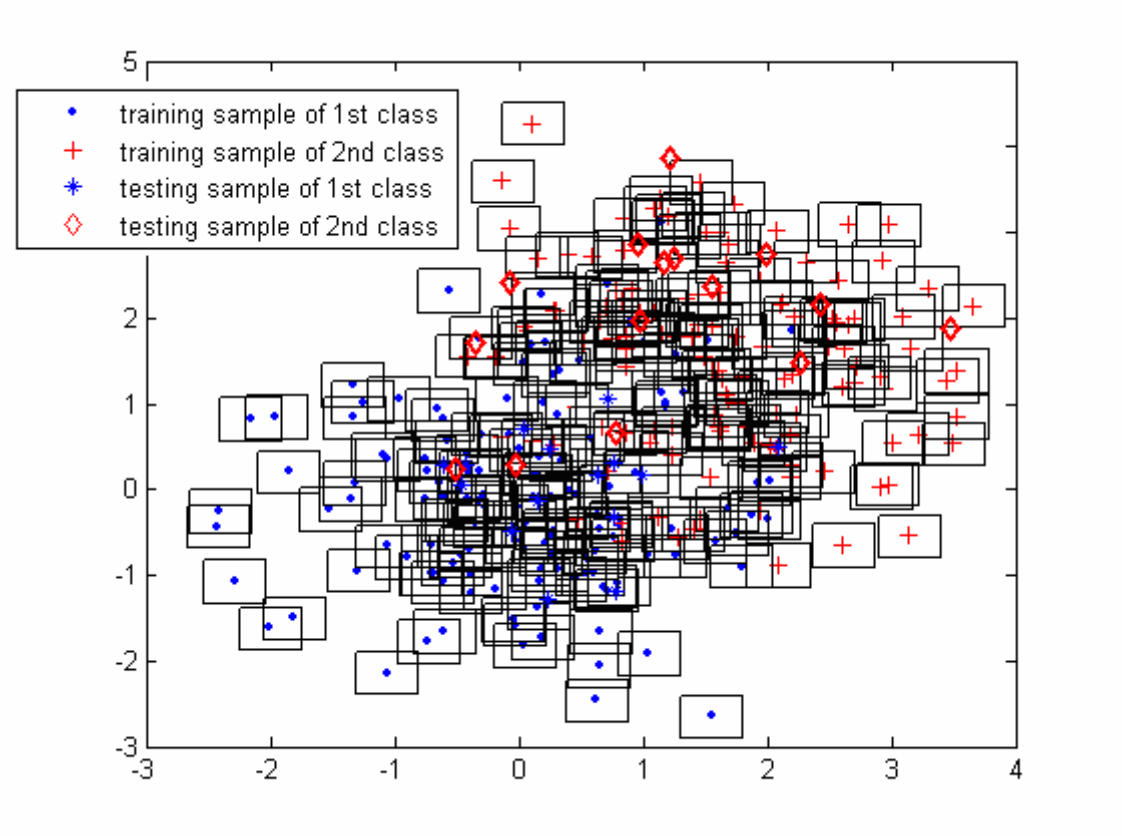

Figure 1. Testing samples almost all residing in the *Q*-Union of the training samples in the first artificial dataset in normal distribution

Table 1. Training and testing accuracy in the first artificial dataset in normal distribution

	LR	R_{SM}
Training Accuracy	0.8741	0.8741
Testing Accuracy	**0.8667**	0.8000

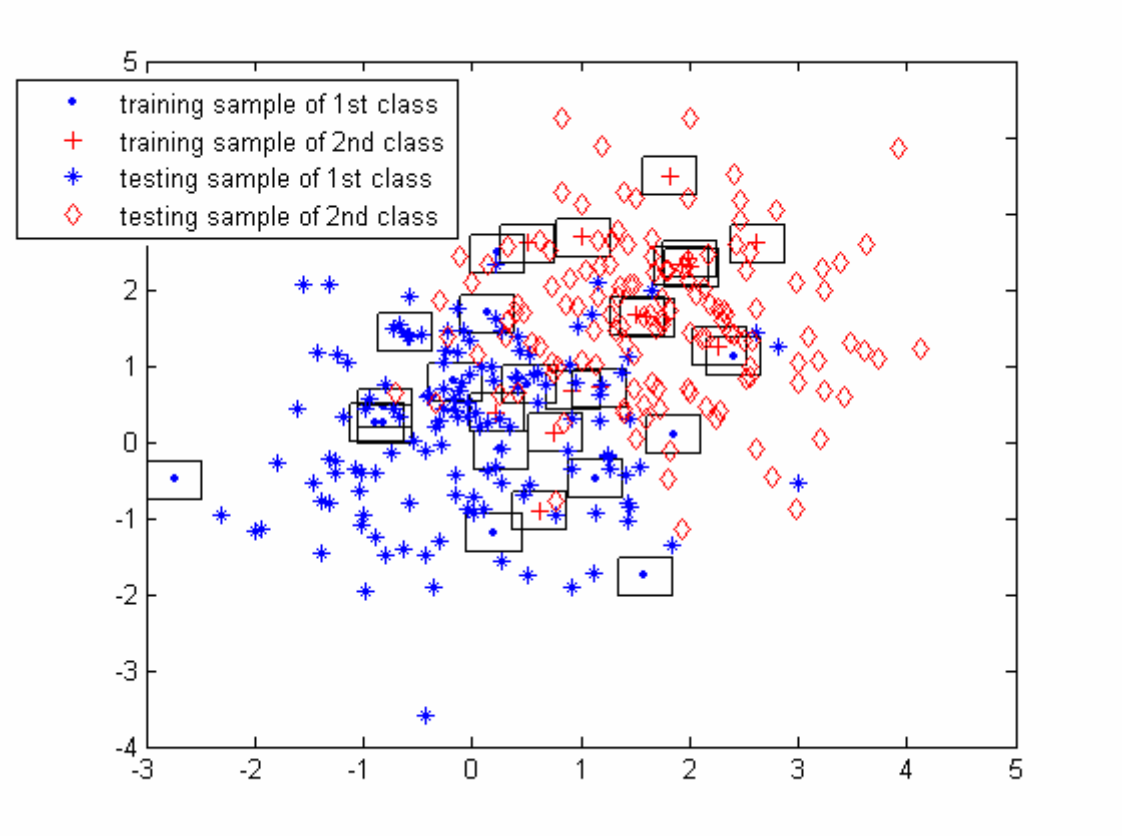

Figure 2. Most of the testing samples residing outside the *Q*-Union in the second artificial dataset in normal distribution

Table 2. Training and testing accuracy in the second artificial dataset in normal distribution

	LR	R_{SM}
Training Accuracy	**0.9000**	0.8667
Testing Accuracy	**0.8963**	0.7556

5.1.2 Uniform distribution

Uniform distribution is a bit difficult for pattern classification than normal distribution. In practice, when we have no prior knowledge of the distribution of the samples in real world, sometimes we may assume that the samples follow the uniform distribution. So we give another artificial database in the bivariate uniform distribution with fewer samples than in the first artificial problem. The database also has two datasets. Each dataset contains 50 samples

in each class. Similarly to the first artificial problem, in the first dataset we randomly choose 45 samples in each class as the training set and the remaining 10 samples as the testing set. In contrast, in the second dataset, we select 10 samples to combine the training set and the remaining 90 samples as the testing set. Figure 3 and Figure 4 also respectively illustrate the different distribution of the testing samples in the two datasets. Table 3 and Table 4 show that the testing accuracies of LR are still superior to R_{SM} on the two datasets in the uniform distribution, which further validates our conclusion. Moreover, although the training accuracies of the two methods are almost equivalent, R_{SM} is apparently overfitting in the second dataset. In contrast, since the training and testing accuracies of LR are basically comparable, this does not appear in LR. Further, the generalization capability of R_{SM} is much sensitive to the sampling of the testing samples and thus the gap between the testing accuracies between the two methods is also much wider in the second dataset.

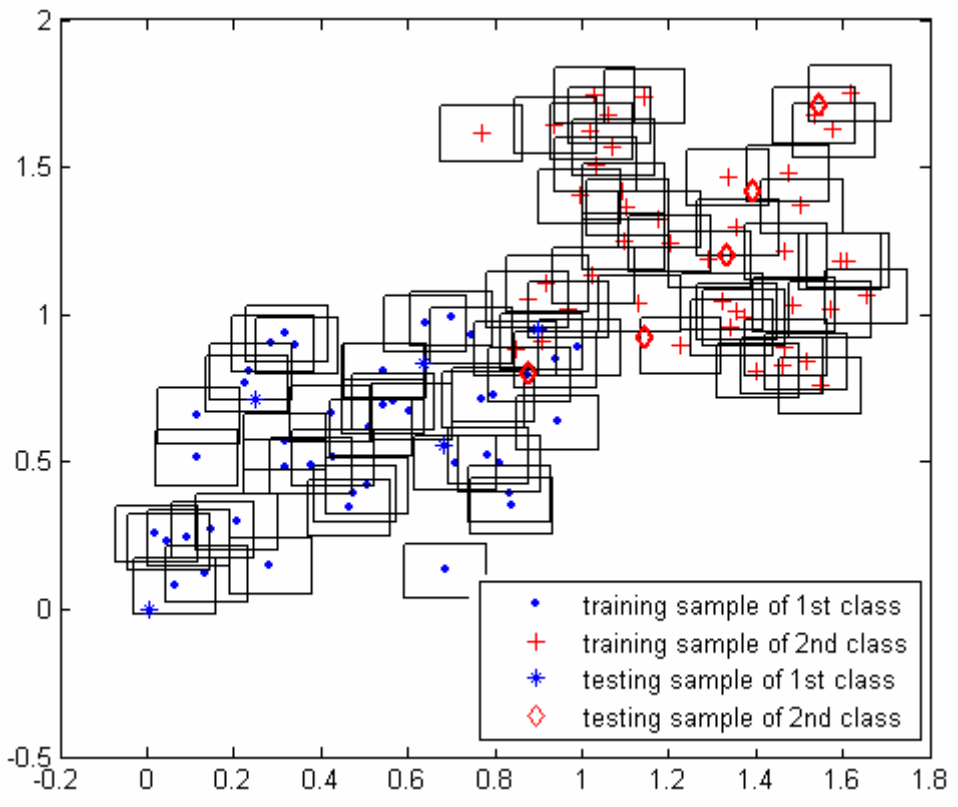

Figure 3. Testing samples all residing in the *Q*-Union of the training samples in the first artificial dataset in uniform distribution

Table 3. Training and testing accuracy in the first artificial dataset in uniform distribution

	LR	R_{SM}
Training Accuracy	0.9667	0.9667
Testing Accuracy	**0.9000**	0.8000

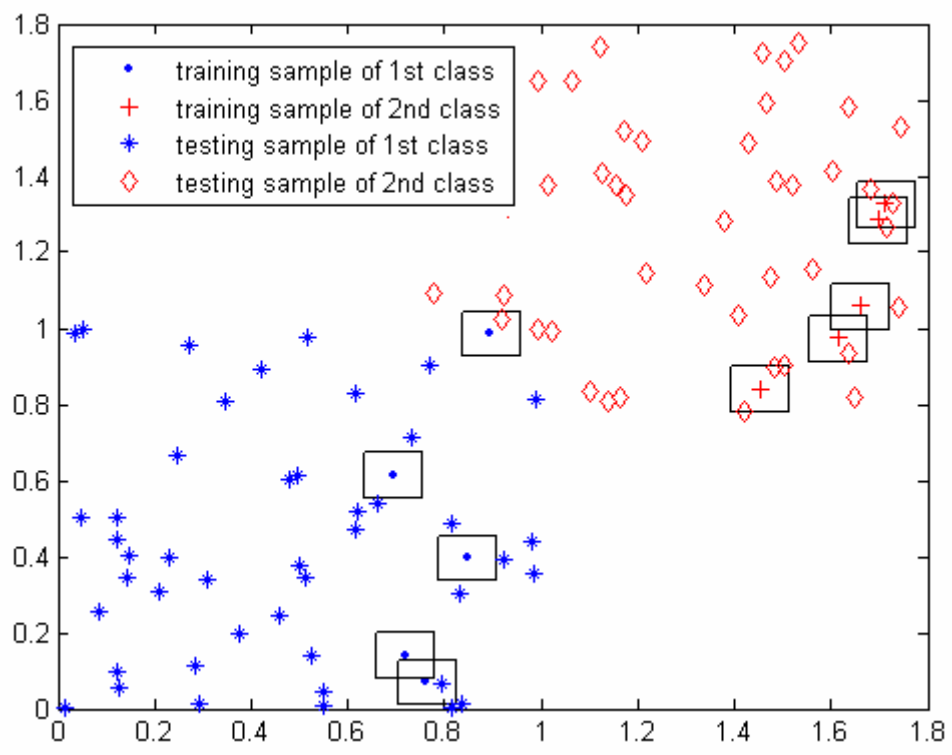

Figure 4. Most of the testing samples residing outside the *Q*-Union in the second artificial dataset in uniform distribution

Table 4. Training and testing accuracy in the second artificial dataset in uniform distribution

	LR	R_{SM}
Training Accuracy	1	1
Testing Accuracy	**0.9889**	0.7667

5.2 Experiments on Partial UCI Database

We choose six datasets, Iris, Sonar, Ionosphere, Wdbc, Pid and Spambase, in the UCI database (the UCI Machine Learning Repository) as examples, where for Iris, the second class and the third class are selected for classification, just because the two classes are linearly inseparable. As shown in Table 5, in the six datasets, Iris is a small-scale one with the amounts of samples between classes evenly, Sonar is middle-scale one with a little uneven amounts, Ionosphere and Wdbc are a bit large-scale and more uneven, and Pid and Spambase

are more large-scale sets and the samples in each class distribute more unevenly. We compare the classification accuracy between LR and R_{SM} in the six datasets with different scales and distributions respectively. For each dataset, we divide the samples into two non-overlapping parts: training and testing sets. This is repeated ten times to generate ten independent runs for each dataset. The testing set is treated as future unseen samples.

We test LR and R_{SM} in three different cases. Firstly, we select almost half of samples in each class respectively and combine them as the training set. The remaining samples are taken as the testing set. Table 6 shows that the average classification accuracies of LR and R_{SM} seem basically comparable in this case. Although LR is superior to R_{SM} in five datasets, the gaps between the accuracies are relatively small in these datasets except for Ionosphere and Spambase, owing to the relatively even sampling of the training and testing samples within classes. The testing samples reside almost uniformly in the Q-Union in R_{SM}. Hence R_{SM} can preserve good generalization capability for these samples.

Secondly, we mix all the samples of two classes and stochastically select almost half of samples as the training set. In this circumstance, the training and testing samples within classes are not sampled evenly again. As a result, some testing samples may be outside the Q-Union in R_{SM}. It likely influences the generalization performance of R_{SM}. Table 7 shows the average classification accuracies of the two methods in this case. As can be seen, LR outperforms R_{SM} consistently in the six datasets. A comparison between Table 7 and Table 6 shows that the classification accuracies of R_{SM} seem to decrease relatively greatly. In contrast, the classification accuracies of LR seem to be more stable and coincidental to those in the first case basically.

In order to further demonstrate our conclusions, we perform the *t*-test on the classification results of the ten runs in the above two cases respectively, to calculate the statistical significance of LR. The null hypothesis H_0 demonstrates that there is no significant difference between the mean number of patterns correctly classified by LR and R_{SM}. If the hypothesis H_0 of each dataset is rejected at the 5% significance level, i.e., the *t*-test value is more than 1.7341, the corresponding results in Table 6 and Table 7 will be denoted '*'. Consequently, compared Table 6 with Table 7, it can be clearly found that the difference of generalization performance between LR and R_{SM} is much more significant in the second case than in the first one. This just accords with our conclusions.

Finally, we test the generalization capability of the two methods in an extreme case on the Iris dataset. The reason that we only consider Iris is the dimension of the other five datasets is too high to be visualized. As we all know, the Iris dataset contains four attributes of an iris. To visualize the problem we restrict ourselves to the two features that contain the most information about the classes, namely the petal length and the petal width [32]. We randomly select 45 samples in the second class and 5 samples in the third class as the training set. And the remaining samples are combined as the testing set. In R_{SM}, the computed value of Q is only close to 10^{-19}. Hence almost all the testing samples are outside the Q-Union. Table 8 shows the training and testing accuracies of the two methods on the biased dataset respectively. For R_{SM}, although the training accuracy is 100%, the testing accuracy is only 78%. So it apparently leads to an overfitting tendency because the optimal Q value is now almost equal to zero and thus the stochastic sensitivity measure does not come to play in optimization. Consequently, the minimization of the generalization error for R_{SM} boils down

to just optimizing R_{emp}. Hence, the value of Q can not only determine the generalization capacity of the classifier, but also be used to explain whether the classifier is overfitting or not. In contrast, LR is relatively stable on both accuracies and has much better generalization performance in this extreme case. The discriminant boundries in Figure 5 further validate our conclusion.

Table 5. The dimension and the respective class sizes of the 6 datasets in the UCI database

Dataset	Dimension	Class I size	Class II size
Iris	4	50	50
Sonar	60	97	111
Ionosphere	34	225	126
Wdbc	30	212	357
Pid	8	500	268
Spambase	57	2788	1813

Table 6. Average classification accuracy when the training and testing samples sampled evenly

Dataset	Classification accuracy	
	LR	R_{SM}
Iris	**0.9800**	0.9780
Sonar	**0.8357**	0.8314
Ionosphere	**0.9119**	0.8480*
Wdbc	0.9447	**0.9477**
Pid	**0.7596**	0.7471*
Spambase	**0.8609**	0.7283*

'*' Denotes that the difference between LR and R_{SM} is significant at 5% significance level, i.e., t-value >1.7341

Table 7. Average classification accuracy when the training and testing samples sampled unevenly

Dataset	Classification accuracy	
	LR	R_{SM}
Iris	**0.9820**	0.8860*
Sonar	**0.8154**	0.5596*
Ionosphere	**0.9080**	0.7937*
Wdbc	**0.9302**	0.8951*
Pid	**0.7427**	0.6622*
Spambase	**0.8457**	0.6718*

Table 8. Training and testing accuracy in the biased Iris dataset

	LR	R_{SM}
Training Accuracy	0.9400	**1.0000**
Testing Accuracy	**0.9800**	0.7800

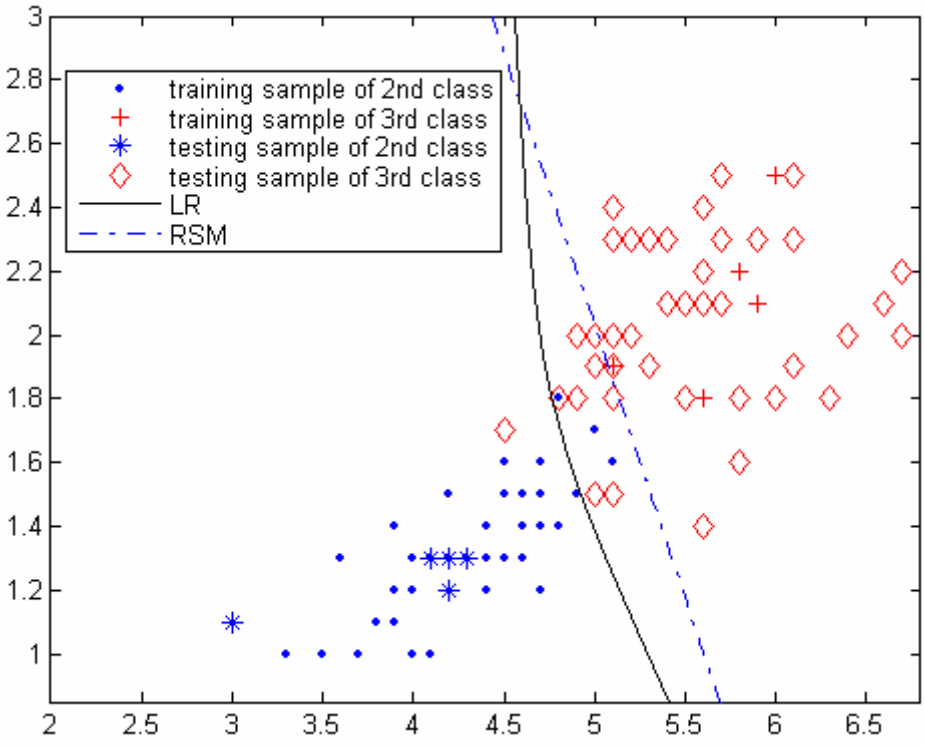

Figure 5. The classification discriminant boundries in the biased Iris dataset

5.3 Experiments on Benchmark Database

Furthermore, the Benchmark database [30] is also used in this test, which consists of 13 datasets. These datasets all contain two classes. We use the training and testing sets offered by the database. Table 9 presents a brief description of these datasets, which are typical sets that the training and testing samples are sampled unevenly within classes. For R_{SM} has been verified superior to multiple folds cross validation and sequential learning in [2], in this experiment, we compare LR with R_{SM}, regularization network (RN) [33, 34], and three

ad-hoc methods. RN originates directly from the regularization theory. RBF function is chosen to be the activation function of the individual hidden units in this network, corresponding to R_{SM}. Thanks to the specific selection of the kernel function, RN is empirically equivalent to SVM here [6, 35, 36]. The ad-hoc "min (k)" and "max (k)" methods are to fix the number of the nearest neighbors which are equivalent to the minimal and maximal values of k in the LR respectively. "k+1" is the method that increases the value of k until the neighborhood contains a sample whose label is different from $\boldsymbol{x}_i$. We perform independently repeatedly 100 runs and 20 runs respectively for the first 11 datasets and the last two datasets.

Experimental results in Table 10 indicate that LR outperforms R_{SM} consistently in almost all the datasets, just as in the last subsection. And the t-test further demonstrates the significant superiority of LR to R_{SM}, in terms of the classification performance in these typical datasets in which the samples are sampled unevenly in training. Furthermore, LR is also superior to RN in most of the datasets, except for Thyroid and Image. And the t-test shows that there is no significant difference between the two methods in the classification accuracies in the two datasets in the statistical senses. As we can see from the table, "min (k)" is obviously underfitting. And the classification accuracies of "max (k)" and "k+1" are both worse than LR in all the datasets. The poor performance of the three ad-hoc methods demonstrates that the way to generate the k nearest neighbors in LR is the optimal option in practice.

Table 9. The dimension, training and testing set size of the 13 datasets in the Benchmark database

Dataset	Dimension	Training set size	Testing set size
Banana	2	400	4900
B.-cancer	9	200	77
Diabetis	8	468	300
F.-Solar	9	666	400
German	20	700	300
Heart	13	170	100
Ringnorm	20	400	7000
Thyroid	5	140	75
Titanic	3	150	2051
Twonorm	20	400	7000
Waveform	21	400	4600
Image	18	1300	1010
Splice	60	1000	2175

Table 10. Average classification accuracy for the testing set in the 13 datasets

Dataset	Classification accuracy					
	LR	R_{SM}	RN	min(k)	Max(k)	k+1
Banana	**0.9002**	0.6768*	0.8790*	0.4470*	0.8852*	0.8869*
B.-cancer	**0.7597**	0.6909*	0.7156*	0.3065*	0.7338*	0.7403*
Diabetis	**0.7697**	0.7230*	0.7553*	0.3430*	0.7377*	0.7510*
F.-Solar	**0.6800**	0.6055*	0.6487*	0.5585*	0.6675*	0.6668*
German	**0.7750**	0.6947*	0.7693*	0.2993*	0.7347*	0.7457*
Heart	**0.8490**	0.7940*	0.7920*	0.4560*	0.8340*	0.8400*
Ringnorm	**0.9856**	0.9151*	0.9487*	0.4947*	0.8296*	0.8275*
Thyroid	0.9527	0.8920*	**0.9573**	0.3013*	0.9040*	0.9267*
Titanic	**0.7794**	0.7573*	0.7757*	0.3230*	0.7752*	0.7752*
Twonorm	**0.9871**	0.9704*	0.9668*	0.5004*	0.9786*	0.9786*
Waveform	**0.9198**	0.8189*	0.8958*	0.3295*	0.8834*	0.8874*
Image	0.9554	0.6796*	**0.9587**	0.5670*	0.8792*	0.9317*
Splice	**0.8920**	0.6960*	0.8859*	0.4787*	0.8273*	0.8292*

‘*’ Denotes that the difference between LR and other methods is significant at 5% significance level, i.e., *t*-value >1.7341

6. Conclusion

In this paper, through proving the relationship between R_{SM} and regularization, we first

develop a new generalization error bound, and then derive a new regularization method, called as LR, from minimizing the bound. Different from traditional regularization methods, LR calculates the regularizing term based on the local *k*-neighborhood of every training sample. Due to combining the global and local structure information, LR has better generalization performance. Besides, compared with R_{SM}, LR can achieve the trade-off between the training accuracy and the generalization capacity of the classifier, instead of bounding the generalization error in the limited *Q*-Union. Furthermore, LR applies a general form to all classifiers, which is a linear combination of kernels, instead of a certain form to a specific classifier. Hence LR can choose different kernels based on various data distribution of different pattern classification problems. The experimental results demonstrate that LR is superior to R_{SM} in terms of generalization capability, especially in the case that training and testing samples are sampled unevenly within classes.

LR is used in supervised learning in this paper. It can also be generalized to semi-supervised learning and multiple kernel learning. Furthermore, LR can be combined with manifold learning. These issues will be our future research directions.

Reference

[1] V. Vapnik. Statistical Learning Theory. Wiley,1998

[2] Wing W. Y. Ng, D. S. Yeung, D. Wang, E. C. C. Tsang, and X. Wang. Localized generalization error and its application to RBFNN training. Proc. of intl. Conf. on Machine Learning and Cybernetics, China, 2005

[3] A. N. Tikhonov. On solving incorrectly posed problems and method of regularization.

Doklady Akademii Nauk USSR, vol.151,501-504,1963

[4] A. N. Tikhonov, V. Y. Aresnin. Solutions of Ill-posed Problems. Washington, DC:W.H. Winston, 1977

[5] S. Haykin. Neural Networks: A Comprehensive Foundation. Tsinghua University Press, 2001

[6] Z. Chen, S. Haykin. On different facets of regularization theory. Neural Computation, 14(12), 2791-2846, 2002

[7] R. O. Duda, P. E. Hart, and D.G. Stork. Pattern Classification. Wiley, 2001

[8] V. Cherkassky, F. Mulier. Learning From Data: Concepts, Theory and Methods. Wiley, 1998

[9] V. Cherkassky, X. Shao, F. Mulier, and V. Vapnik. Model complexity control for regression using VC generalization bounds. IEEE Trans. on Neural Networks, vol. 10, 1075-1089,1999

[10] W. Finnoff, F. Hergert, and H. G. Zimmermann. Improving model selection by nonconvergent methods. Neural Networks, vol.6, 771-783, 1993

[11] L. Prechelt. Early stopping – but when? Lect. Notes Comput., SC 1524, 55-69, 1998

[12] Wing W. Y. Ng et al. Image classification with the use of radial basis function neural networks and the minimization of the localized generalization error. Patter Recognition, vol. 40(1), 4-18, 2007

[13] N. Cristianini, J. S. Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, 2000

[14] Wing W. Y. Ng, D. S. Yeung. Selection of weight quantization accuracy for radial basis

function neural network using stochastic sensitivity measure. IEE Electronic Letters, vol.39, 787-789, 2003

[15] Wing W. Y. Ng, D. S. Yeung, X. Wang, and I. Cloete. A study of the difference between partial derivative and stochastic neural networks sensitivity analysis for applications in supervised pattern classification problems. Proc. of intl. Conf. on Machine Learning and Cybernetics, 4283-4288, 2004

[16] Wing W. Y. Ng, D. S. Yeung, and I.Cloete. Quantitative study on effect of center selection to RBFNN classification performance. IEEE Proc. of int'l Conf. on Systems, Man, Cybernetic, 3692-3697, 2004

[17] D. S. Yeung, Wing W. Y. Ng. Feature selection using generalization error for supervised classification problem. SICE Annual Conf., Japan, 2005

[18] P. K. Partrick, Wing W. Y. Ng, and D. S. Yeung. Active Learning Using Localized Generalization Error of Candidate Sample as Criterion. IEEE Proc. of int'l Conf. on Systems, Man, Cybernetic, 3604 - 3609, 2005

[19] Wing W. Y. Ng, A. P. F. Chan, D. S. Yeung, and Eric C. C. Tsang. Quantitative Study on the Generalization Error of Multiple Classifier Systems. IEEE Proc. of int'l Conf. on Systems, Man, Cybernetic, 889-894, 2005

[20] Fan R. K. Chung. Spectral Graph Theory. Regional Conference Series in Mathematics, number 92, 1997

[21] C. A. Micchelli, M. Pontil. Learning the kernel function via regularization. Machine Learning Research, vol.6, 1099-1125,2005

[22] A. Argyriou, M. Herbster, and M. Pontil. Combing graph Laplacians for semi-supervised

learning. NIPS, 2005

[23] W. Rudin. Principles of Mathematical Analysis. McGraw-Hill Science, 1976

[24] X. He, D. Cai, and P. Niyogi. Tensor subspace analysis. NIPS, 2005

[25] K. Lu, J. Zhao, and D. Cai. An algorithm for semi-supervised learning in image retrieval. Pattern Recognition, vol.39, 717-720, 2006

[26] M. Belkin, P. Niyogi. Laplacian eigenmaps and spectral technique for embedding and clustering. NIPS, vol.15, Vancouver, British Columbia, Canada, 2001

[27] W. Hoeffding. Probability inequalities for sums of bounded random variables. Journal of American Statistic Association, vol.58, 13-30, 1963

[28] K. Pelckmans, J. A. K. Suykens, and B. De Moor. Additive regularization trade-off: fusion of training and validation levels in kernel methods. Machine Learning, vol.62, 217-252, 2006

[29] K. Pelckmans. Primal-dual Kernel Machines. PhD thesis, Faculty of Engineering, K. U. Leuven, 2005

[30] G. Ratsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. Machine Learning, vol.42, 287-320, 2001

[31] J. Nocedal, S. J. Wright. Numerical Optimization. Springer, 2000

[32] S. R. Gunn. Support vector machines for classification and regression. Technical Report, 1998

[33] Poggio, T., F. Girosi. Networks for approximation and learning. Proceedings of the IEEE, vol. 78, 1481-1497, 1990a

[34] Poggio, T., F. Girosi. Regularization algorithms for learning that are equivalent to

multilayer networks. Science, vol. 247, 978-982,1990b

[35] Girosi, F. An equivalence between sparse approximation and support vector machines. Neural Computation, vol. 10, 1455-1480, 1998

[36] Evgeniou, T., Pontil, M., and Poggio, T. Regularization networks and support vector machines. Advances in Computational Mathematics, 13(1), 1-50, 2000