Expert Systems With Applications 51 (2016) 85–106

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing

Pilsung Kang[a], Dongil Kim[b,*], Sungzoon Cho[c]

[a] School of Industrial Management Engineering, Korea University, 02841 Seoul, South Korea
[b] Smart Manufacturing Technology Group, Korea Institute of Industrial Technology, 31056 Cheonan, South Korea
[c] Department of Industrial Engineering, Seoul National University, 08826 Seoul, South Korea

ARTICLE INFO

Keywords:
Semi-supervised learning
Support vector regression
Probabilistic local reconstruction
Data generation
Virtual metrology
Semiconductor manufacturing

ABSTRACT

Dataset size continues to increase and data are being collected from numerous applications. Because collecting labeled data is expensive and time consuming, the amount of unlabeled data is increasing. Semi-supervised learning (SSL) has been proposed to improve conventional supervised learning methods by training from both unlabeled and labeled data. In contrast to classification problems, the estimation of labels for unlabeled data presents added uncertainty for regression problems. In this paper, a semi-supervised support vector regression (SS-SVR) method based on self-training is proposed. The proposed method addresses the uncertainty of the estimated labels for unlabeled data. To measure labeling uncertainty, the label distribution of the unlabeled data is estimated with two probabilistic local reconstruction (PLR) models. Then, the training data are generated by oversampling from the unlabeled data and their estimated label distribution. The sampling rate is different based on uncertainty. Finally, expected margin-based pattern selection (EMPS) is employed to reduce training complexity. We verify the proposed method with 30 regression datasets and a real-world problem: virtual metrology (VM) in semiconductor manufacturing. The experiment results show that the proposed method improves the accuracy by 8% compared with conventional supervised SVR, and the training time for the proposed method is 20% shorter than that of the benchmark methods.

1. Introduction

Support vector regression (SVR), a regression version of support vector machines (SVM) (Vapnik, 1995), was proposed to solve nonlinear regression problems with a maximum margin algorithm (Smola & Schölkopf, 2002). SVR employs an ε-insensitive loss function (see Fig. 1); the training data whose margins are less than ε are not considered to be an error. Hence, an ε-sized insensitive tube (ε-insensitive tube or ε-tube) is constructed during SVR training. The data located on or outside the ε-tube are called support vectors, and the SVR regression function is formed as a linear combination of support vectors. SVR has the same advantages as those of SVM. SVR also maximizes the generalization performance by employing the structural risk minimization (SRM) principle with an ε-insensitive loss function, and it is capable of solving nonlinear problems with the kernel trick. With those advantages, SVR has been successfully applied to various areas: response modeling (Kim & Cho, 2012), virtual metrology (VM) (Kang, Kim, Lee, Doh, & Cho, 2011), finance prediction (Pai & Lin, 2005), time-series prediction (Thissen, van Brakel, de Weijer, Melssen, & Buydens, 2003), and environment application (Ortiz-García, Salcedo-Sanz, Pérez-Bellido, Portilla-Giqueras, & Prieto, 2010).

SVR was originally designed for application to the supervised learning problem. In supervised learning, the training dataset consists only of labeled data, $L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|L|}, y_{|L|})\}$, where $\mathbf{x}_l$, y_l, and $|L|$ represent the d-dimensional input variables, corresponding labels, and number of labeled data, respectively (see Eq. (1)). However, in some applications, because the labeled data are difficult, expensive, or time consuming to acquire, the amount of training data used is not sufficient for obtaining effective model performance. Conversely, the unlabeled data, $U = \{\mathbf{x}_{|L|+1}, \ldots, \mathbf{x}_{|L|+|U|}\}$, where $|U|$ is the number of unlabeled data, contains only the input variables (see Eq. (2)) and can be collected with less effort than the labeled data. Such unlabeled data are abundant in many applications, and hence the concept of training a model from those

* Corresponding author. Tel.: +82 31 8040 6774; fax: +82 31 8040 6170.
E-mail addresses: pilsung_kang@korea.ac.kr (P. Kang), dikim01@kitech.re.kr (D. Kim), zoon@snu.ac.kr (S. Cho).

http://dx.doi.org/10.1016/j.eswa.2015.12.027
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

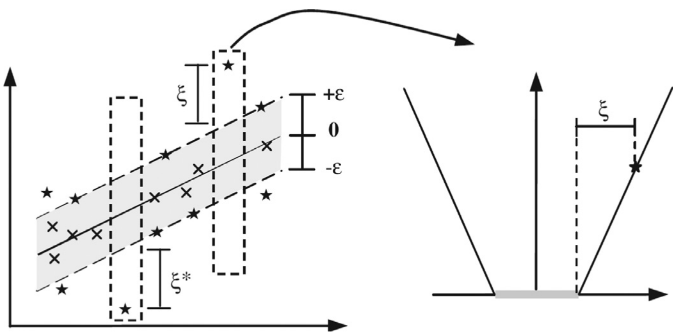

Fig. 1. ε -tube based on the margin of training data, and the ε-loss function of SVR (reprinted from Chen & Wang, 2007).

unlabeled and labeled data in order to improve model performance is proposed (Zhu, 2007).

$$L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|L|}, y_{|L|})\}, \quad \mathbf{x}_l \in \mathbb{R}^d, \quad y_l \in \mathbb{R}. \tag{1}$$

$$U = \{\mathbf{x}_{|L|+1}, \ldots, \mathbf{x}_{|L|+|U|}\}, \quad \mathbf{x}_u \in \mathbb{R}^d. \tag{2}$$

Semi-supervised learning (SSL) is one of the principal approaches for training a model from both labeled and unlabeled data. “SSL addresses this problem by using large amounts of unlabeled data, together with the labeled data, to build better classifiers” (Zhu, 2007). The research efforts for SSL can be categorized into five main directions: generative (Nigam & Ghani, 2000; Seeger, 2000), self-training (Mihalcea, 2004; Rosenberg, Hebert, & Schneideman, 2005), co-training (Blum & Mitchell, 1998; Mitchell, 1999), low density separation (Chapelle, Sindhwani, & Keerthi, 2008; Vapnik, 1998), and graph-based methods (Belkin, Niyogi, & Sindhwani, 2006; Sindhwani, Niyogi, & Belkin, 2005). Those methods have been applied widely to various real-world problems; however, most works for SSL were designed for classification problems. Because the label is a continuous real numbered variable for regression problems, most of these methods cannot be applied to regression problems directly (Cortes & Mohri, 2007; Pozdnoukhov & Bengio, 2006; Wang, Hua, Song, Dai, & Zhang, 2006).

Recently, SSL for regression has been proposed. SSL for regression is more complicated than SSL for classification. In order to utilize the unlabeled data for training, SSL for regression must estimate continuous-valued labels, explicitly or implicitly; only binary labels are required for SSL for classification. Co-training (Wang, Fu, , & Ma, 2011; Wang, Ma, & Wang, 2010; Zhou & Li, 2007), graph-based methods (Pozdnoukhov & Bengio, 2006), and kernel-based methods (Cortes & Mohri, 2007; Wang et al., 2006) have been proposed for SSL regression. However, these methods have limitations. The uncertainty of estimating labels for the unlabeled data is not considered. Because the labels for the unlabeled data should be estimated using mathematical models, a labeling uncertainty always exists. The estimated labels for the unlabeled data should be addressed differently based on their uncertainties. Moreover, because SVR is a margin-based method, only the estimated labels located on or outside the ε-tube influence the final SVR model. However, those methods, which employ a weighted average of nearest neighbors or a regression function trained by the labeled data, tend to estimate the labels of the unlabeled data inside the ε-tube. These do not improve the model accuracy. Finally, the time complexities of these methods are relatively high. Because co-training is a wrapper-based iterative approach, two base learners should be trained in each iteration. In addition, the graph-based methods need to calculate the entire kernel-based weight matrix for all data, including the unlabeled data.

In this paper, we propose a self-training based non-iterative semi-supervised support vector regression algorithm that estimates the label distribution of each unlabeled data point and oversamples based on the uncertainty of the labeling. The proposed method is designed to consider both accuracy and efficiency. The principal contribution of the proposed method can be summarized as follows:

(1) **The proposed method considers the uncertainty of the estimated labels of the unlabeled data.** In order to consider such labeling uncertainty, the proposed method estimates the label distribution (not a label value) of each unlabeled data point. Then, each unlabeled data point has an input value and corresponding Gaussian label distribution that consists of the mean and variance. The lower the variance of the estimated label distribution, the lower is the uncertainty of the unlabeled data point. Conversely, if the variance of the estimated label distribution is greater, the labeling uncertainty for the unlabeled data point is higher. Probabilistic local reconstruction (PLR) (Lee, Kang, & Cho, 2014), a local topology-based linear reconstruction method, is employed for estimating the label distribution. The proposed method employs two PLR models with different settings to capture and conjugate the local and global topology of the unlabeled data.

(2) **The proposed method generates data by oversampling from the unlabeled data and their estimated label distribution.** The proposed method randomly generates multiple training data from the unlabeled data and their estimated label distributions in order to increase the probability of the unlabeled data affecting the final SVR training. Moreover, the sampling rate is different based on the uncertainty of the estimation for each unlabeled data point. For those unlabeled data with low labeling uncertainty, only a few samples are generated from the estimated label distribution. Conversely, for those unlabeled data with high labeling uncertainty, more samples are generated in order to represent the entire estimated label distribution.

(3) **The training complexity of the proposed method is relatively low.** The proposed method employs a non-iterative algorithm. In addition, the proposed method does not need to construct a large-sized graph on the labeled and unlabeled data. Hence, the proposed method demands significantly less complexity than the iterative or graph-based methods. Moreover, to reduce the training data generated by oversampling, an additional training data selection method,expected margin-based pattern selection (EMPS) (Kim & Cho, 2012), is employed for training efficiency.

The performance of the algorithm is verified using sufficient benchmark regression datasets. Based on the experiments conducted on 30 regression datasets, it is verified that the proposed method improves accuracy by approximately 8% over the conventional supervised SVR. In addition, the training time of the proposed method, including the construction of the final SVR, is reduced to only 20% of the benchmark methods. We also demonstrate a real-world application of VM in semiconductor manufacturing. VM is a mathematical model that employs regression algorithms to estimate the quality of wafers in a manufacturing process, which has a few labeled data and abundant unlabeled data. The proposed method shows excellent performance for the semi-supervised VM problem.

The remainder of this paper is organized as follows. In Section 2, a review of the SSL literature is presented. In Section 3, the proposed method for SS-SVR is proposed. In Sections 4 and 5, the experiment results on benchmark datasets and a real-world semiconductor manufacturing problem are presented, respectively. Finally, Section 6 concludes this paper with a summary and limitations of the proposed method and future works are discussed.

2. Literature review

2.1. Semi-supervised learning

Because the field of SSL is evolving rapidly, it is difficult to categorize all aspects of SSL explicitly. However, past literature suggests that the main research areas for SSL can be summarized into five groups: generative, self-training, co-training, low density separation, and graph-based methods (Chapelle, Schölkopf, & Zien, 2006; Zhu, 2007). The generative method is one of the oldest SSL methods. This method estimates the parameters of the underlying (mixture) distribution with labeled and the unlabeled data. The existence of abundant unlabeled data can guarantee more accurate estimation of underlying distribution for the input space. Ideally, if there exists one labeled data point per component, labels can be assigned for all unlabeled data. Nigam and Ghani (2000) proposed an EM algorithm for a mixture of multinomial distribution, and the algorithm was applied for a text classification task (Nigam, McCallum, & Mitchell, 2006). Seeger (2000) proposed an overview of the Bayesian framework for SSL, including EM algorithm and co-training. Fujino, Ueda, and Saito (2005) proposed a hybrid approach of generative and discriminative methods that employs a bias correction model for the generative part and maximum entropy principle for the discriminative part. Some studies employed clustering algorithms instead of estimating the probability distribution (Demiriz, Bennett, & Embrechts, 1999). The generative methods tend to degrade classification accuracy when estimation of the mixture distribution or clustering results is incorrect.

Self-training is a commonly used technique for SSL (Zhu, 2007). The theory of self-training is to estimate labels for the unlabeled data and select the unlabeled data that can improve model performance. First, a model trained only on the labeled data predicts labels for the unlabeled data. Then, the most confident unlabeled data are added to the training set with predicted labels, and the model is re-trained. These procedures are repeated until a convergence condition is satisfied. The generative methods can be viewed as a special case of self-training (Zhu, 2007). Mihalcea (2004) proposed a Naïve Bayesian-based self-training method and applied it to a word sense disambiguation system. Rosenberg et al. (2005) applied a self-training method for an object-detection problem for images. Li, Guan, Li, and Chin (2008) proposed a self-training SVM algorithm, including model selection and convergence analysis, and the algorithm was applied to a brain computer interface system. Adankon and Cheriet (2011) proposed a help-training SVM that improved a classical self-training method by employing generative models to help discriminative classifiers in order to overcome the misclassification problems of self-training. Maulik and Chakraborty (2011) proposed an ensemble-based self-training SVM that employs a fuzzy *c*-means for labeling the unlabeled data, and the algorithm was successfully applied to the pixel classification problem. Self-training is considered a basic concept for involving the most confident unlabeled data in training. However, self-training tends to suffer from local optimum.

Co-training methods (Blum & Mitchell, 1998; Mitchell, 1999) have been proposed to overcome this self-training issue by constructing two different base models, each of which is trained from a different perspective of the labeled dataset. Co-training has three strong assumptions: (1) features can naturally be split into two sets, (2) each sub-feature set is sufficiently good for training a model, and (3) the sub-feature sets are conditionally independent. Co-training trains two base models with different perspectives (different sub-feature sets) of the training data. Each base model evaluates the unlabeled data, and each unlabeled data point has two labels, one from each base model. Then, each base model is trained from the new training datasets by adding unlabeled data individually. The unlabeled data point that improves the model's accuracy the most is added to the other model's training dataset. Brefeld and Scheffer (2004) proposed co-EM, which employs a strategy for labeling only a few confident data points, and also showed that if there were no natural feature split, a random split of features into two subsets would be helpful. Balcan, Blum, and Yang (2004) proposed an "expansion assumption" that reduced the strong theoretical requirements of co-training, and they showed that co-training works well when each base model is accurate and two perspectives are sufficiently different, although not necessarily independent. Zhou and Goldman (2004) proposed a single-view multi-learner democratic co-training algorithm. They employed different learning models as base learners, each of which was trained from the complete feature set to make the inductive bias. The majority voting from ensemble learning was used for the final classification. Zhou and Li (2005) proposed the tri-training method that uses three classifiers as base learners. In order to construct robust base models, co-regularization learning was proposed to penalize the high level of confidence from different perspectives by employing a regularization term (Krishnapuram et al., 2005). Recently, a Bayesian co-training algorithm that employs the probabilistic model based on the co-regularization method has been proposed (Yu, Krishnapuram, Rosales, & Rao, 2011). The Bayesian co-training method has provided a strong theoretical background and efficient optimization. Co-training has been expanded as the general multi-view learning (Zhu, 2007). Co-training has become the most powerful approach to SSL; however, the training complexity of co-training is relatively high.

The low density separation method is another direction of studies in SSL. Transductive SVM (TSVM) (Vapnik, 1998) is one of the popular methods for low density separation. TSVM is an extended version of the conventional SVM that involves unlabeled data in the optimization formula of SVM. TSVM predicts the labels for the unlabeled data, so that a decision boundary has the maximum margin on both the labeled data and test (unlabeled) data. Hence, the decision boundary of TSVM is located in a low data-density region in the input space. Originally, transductive learning constructs a model with the labeled data and test (unlabeled) data, whereas inductive learning constructs a prediction model with the labeled and unlabeled data, and the constructed model is assumed to be applied to the unseen test data. However, because TSVM learns an inductive rule defined over the entire input space, TSVM is also named semi-supervised SVM (S^3VM) (Chapelle et al., 2008; Zhu, 2007). Given that obtaining the exact solution of TSVM is NP-hard, many studies have focused on efficient optimization techniques for TSVM. Bennett and Demiriz (1999) proposed an early version of approximation algorithms for TSVM, whereas Joachims (1999) implemented an efficient SVM^{light} algorithm with local combination search. Chapelle and Zien (2005) proposed ∇SVM by approximating a non-convex loss function of SVM with a Gaussian function, so that the gradient descent could be applied. In that research, they proposed a density-sensitive distance measure to construct a graph, which was designed to enforce the cluster assumption in TSVM. Sindhwani and Keerthi (2006) proposed an efficient TSVM based on a deterministic annealing approach for a large-scale dataset, and Reddy, Shevade, and Murty (2011) employed a quasi-Newton method for the SVM optimization term in order to obtain a fast and generalized performance. Recently, a Bayesian version of TSVM (Chakraborty, 2011) and the cost-sensitive version of TSVM have been proposed (Li, Kwok, & Zhou, 2010; Qi, Tian, Shi, & Yu, 2013). However, TSVM studies have mainly focused on classification problems, and they cannot be applied to regression problems directly (Cortes & Mohri, 2007). The principle of low density separation has been applied to other approaches, such as Gaussian process (Lawrence & Jordan, 2005), information regularization (Corduneanu & Jaakkola, 2003), and entropy minimization (Grandvalet & Bengio, 2005).

The graph-based method is the other main approach of SSL. This method constructs a graph where the nodes are the labeled and unlabeled data in the dataset, and the edges reflect the similarity based on local topologies of data. The distance metric, optimization method, and regularization term are slightly different; however, they have similar approaches (Culp & Michailidis, 2008; Nie, Xiang, Liu, & Zhang, 2010; Zhang & Wang, 2009; Zhu, 2007). The graph-based method is inherently transductive. Some studies proposed inductive graph-based methods. Zhao, Chow, Zhang, and Li (2015) proposed a compact graph-based semi-supervised learning (CGSSL) that constructed a compact graph by minimizing reconstruction errors with adjacent neighbors. The manifold regularization-based method is one of the special cases of the graph-based method. However, it concentrates more on the manifold assumption that data actually belong to some lower dimensional manifold. A kernel deformation was proposed to modify a kernel function in order to reflect the unlabeled data space (Belkin, Mateeva, & Niyogi, 2004; Belkin et al., 2006). The basic concept of those methods is that the local topology of the input variables is the key in the estimation of the unlabeled data. Sindhwani et al. (2005) proposed a "point-cloud norm" defined by $L\bigcup U$ using a Laplacian matrix of a graph for the regularization term, and this concept has been successfully expanded to various manifold regularization algorithms (Pozdnoukhov & Bengio, 2006). A new kernel matrix that considers both labeled and unlabeled data can be calculated as a modification of the base kernel function defined in the original space. The limitations of the graph-based methods are: (1) training complexity, (2) vulnerability to noisy data, and (3) requirement for re-training when the dataset changes.

2.2. Semi-supervised learning for regression

Many studies for SSL regression have focused on algorithms that estimate the confident continuous labels of the unlabeled data in order to construct an accurate SSL model. Co-training is a great example. The advantage of co-training for SSL regression is that a regression model can be directly involved for estimating the labels of the unlabeled data, and focus on improving model accuracy with iterative steps. COREG (Zhou & Li, 2007) is one of the most well-developed algorithms for SSL regression. In COREG, a k-NN regression model is employed as the base model for co-training. To create a different view for the two base models, two initial training datasets are constructed by sampling from the labeled data randomly. Moreover, each k-NN is set to use a different parameter k, the number of nearest neighbors, and different distance measures, Mahalanobis and Euclidean distance. Extensive experimentation has shown that COREG can improve the performance of conventional supervised regression models or self-training based methods. Co-SVR (Wang et al., 2011; Wang et al., 2010) has also been proposed. The framework for Co-SVR is similar to that for COREG, with the exception of Co-SVR employing SVR for the base models. There are some limitations with Co-SVR: because SVR is a function estimation model, not a local topology-based model, Co-SVR has the possibility of easily suffering from local optimum. The additional unlabeled data might not have new information for the underlying function. Moreover, because the base model for Co-SVR is SVR, the training complexity of Co-SVR is relatively higher than that of other co-training methods.

Other research directions have focused on graph-based and kernel-based methods. Pozdnoukhov and Bengio (2006) proposed a modification of the original manifold regularization-based method (Sindhwani et al., 2005) to solve regression problems. A data dependent kernel that captures the inner geometry of the data, including the unlabeled data, replaced the original kernel of SVR with the Laplacian matrix of the graph built on unlabeled data. Wang et al. (2006) proposed semi-supervised kernel regression (SSKR) that extended the conventional kernel regression to train from both the labeled and unlabeled data with a weight factor to modulate the effect of unlabeled data. They also indicated that the derivations of the graph-based and kernel-based methods are closely related. Brefeld, Gärtner, Scheffer, and Wrobel (2006) proposed the co-regularized least squares regression (coRLSR) that employed multi-view learning for regularized kernel regression, and proposed a semi-parametric approximation that scaled linearly for the optimization process of coRLSR. In Cortes and Mohri (2007), a kernel ridge regression-based transductive method was proposed. At the first stage, the labels for the unlabeled data are estimated based on the kernel distance from their nearest neighbors. Then, global optimization is conducted by applying standard ridge regression with an added regularization term to unlabeled data. However, because this method is transductive, it has a limitation in predicting unseen test data. Xu, An, Qiao, Zhu, and Li (2011) proposed a semi-supervised least-squares SVR (S^2LS-SVR) method by applying the concept of the kernel ridge-based transductive method to the least-squares SVR (LS-SVR). Because LS-SVR replaced the convex quadratic optimization problem with a convex linear problem, S^2LS-SVR had advantages for training with a large dataset.

3. Proposed method

In this paper, we focus on the idea that the estimated labels of the unlabeled data have uncertainties. The estimated labels for some unlabeled data can be somewhat accurate, but others can be more uncertain. The proposed method considers label uncertainty. The algorithm can be summarized into four detailed stages: (1) estimating the label distribution of the unlabeled data, (2) training data generation with oversampling based on the labeling uncertainty, (3) training data selection to reduce the training complexity, and (4) training the final SVR model. In the first stage, the label distribution of the unlabeled data is estimated using two PLR regression models (2-PLR), and the estimated label for each unlabeled data point is estimated in a Gaussian probabilistic distribution form, not a single value. In the second stage, the training data are generated from the unlabeled data and their estimated label distribution. With the uncertainty of the estimated labels for the unlabeled data, the data generation rate varies. Then, an integrated set is constructed from the labeled data and those generated from the unlabeled data. In the third stage, the data selection method is applied to reduce the training complexity increased by oversampled data. In the last stage, the final SVR model is trained by the selected set from the third stage. Fig. 2 compares the training procedure of the co-training based iterative SSL for regression (a) and the overall procedure of the proposed method (b).

3.1. Estimating label distribution for unlabeled data

In order to estimate the uncertainty for the labels of the unlabeled data, PLR is employed. PLR is a probabilistic regression model that uses the local topology of the input and test data. For k-NN, which is the basic local topology-based method, a label for a given test data, $\mathbf{x}_i$, is estimated with the following equation,

$$\hat{y}_i = \sum_{j=1}^{k} w_{\mathrm{NN}(\mathbf{x}_i,j)} y_{\mathrm{NN}(\mathbf{x}_i,j)} = \mathbf{w}^T_{\mathrm{NN}(\mathbf{x}_i)} \mathbf{y}_{\mathrm{NN}(\mathbf{x}_i)}, \tag{3}$$

where $\mathrm{NN}(\mathbf{x}_i, j)$, $\mathrm{NN}(\mathbf{x}_i)$, and k denote the jth nearest neighbor of $\mathbf{x}_i$, all nearest neighbors of $\mathbf{x}_i$, and the number of nearest neighbors, respectively. Under the assumption that a data point in the input space can be described by a combination of its neighbors, the locally linear reconstruction (LLR) (Kang & Cho, 2008) method attempts to describe $\mathbf{x}_i$ by the linear reconstruction of its nearest

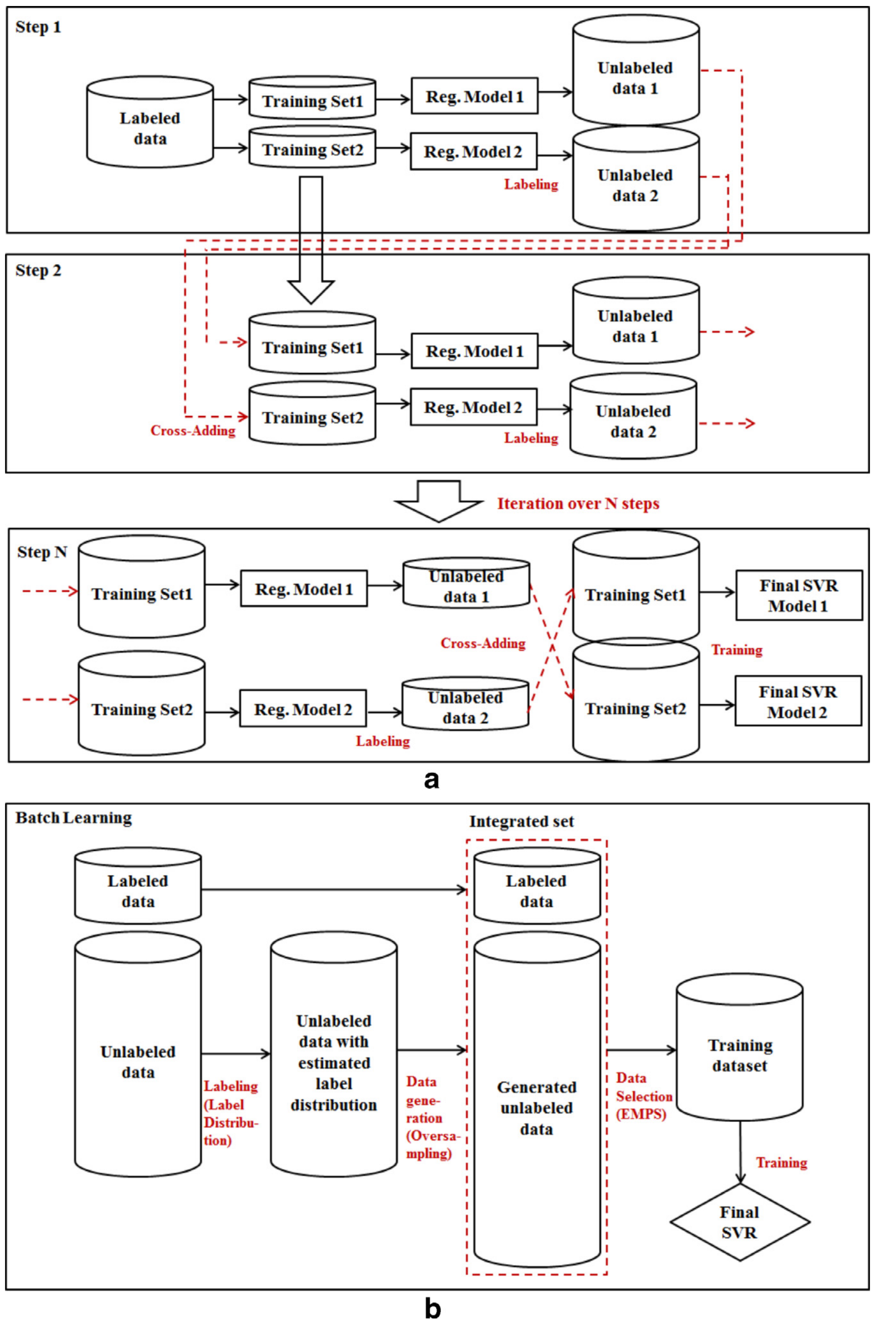

Fig. 2. (a) Training procedure of the co-training based iterative SSL for regression and (b) overall procedure of the proposed method.

neighbors as follows,

$$\mathbf{x}_i = \mathbf{X}_{\text{NN}}\mathbf{w}, \tag{4}$$

where $\mathbf{X}_{\text{NN}} = [\mathbf{x}_{\text{NN}(\mathbf{x}_i,1)}, \ldots, \mathbf{x}_{\text{NN}(\mathbf{x}_i,k)}]$ indicates the matrix of nearest neighbors, and $\mathbf{w} = \mathbf{w}_{\text{NN}(\mathbf{x}_i)} = [w_{\text{NN}(\mathbf{x}_i,1)}, \ldots, w_{\text{NN}(\mathbf{x}_i,k)}]^T$ denotes the weight vector, by minimizing the reconstruction error $E(\mathbf{w})$ in Eq. (5),

$$E(\mathbf{w}) = \frac{1}{2}(\mathbf{x}_i - \mathbf{X}_{\text{NN}}\mathbf{w})^T(\mathbf{x}_i - \mathbf{X}_{\text{NN}}\mathbf{w}). \tag{5}$$

PLR is a probabilistic version of LLR that employs a Bayesian probabilistic form to capture the reconstruction uncertainty. Eq. (4) can be transformed into probabilistic form by adding ϵ, the uncertainty of the reconstruction, as follows:

$$\mathbf{x}_i = \mathbf{X}_{\text{NN}}\mathbf{w} + \epsilon, \quad \epsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I}). \tag{6}$$

Then, the likelihood of the test data point $\mathbf{x}_i$ is

$$p(\mathbf{x}_i|\mathbf{X}_{\text{NN}}, \mathbf{w}) = N(\mathbf{x}_i|\mathbf{X}_{\text{NN}}\mathbf{w}, \sigma^2\mathbf{I}). \tag{7}$$

The posterior of the weights can be calculated using Bayes' rule by employing a zero mean Gaussian prior of the weight parameters, $p(\mathbf{w}) = N(\mathbf{w}|\mathbf{0}, \Sigma)$,

$$\begin{aligned} p(\mathbf{w}|\mathbf{X}_{\text{NN}}, \mathbf{x}_i) &\propto p(\mathbf{x}_i|\mathbf{X}_{\text{NN}}, \mathbf{w})p(\mathbf{w}) \\ &= N(\mathbf{x}_i|\mathbf{X}_{\text{NN}}\mathbf{w}, \sigma^2\mathbf{I})N(\mathbf{w}|\mathbf{0}, \Sigma) \\ &= N(\mathbf{w}|\mu_p, \Sigma_p), \end{aligned} \tag{8}$$

where the posterior mean is $\mu_p = \sigma^{-2}(\sigma^{-2}\mathbf{X}_{\text{NN}}^T\mathbf{X}_{\text{NN}} + \Sigma^{-1})^{-1}\mathbf{X}_{\text{NN}}^T\mathbf{x}_i$, and the posterior covariance matrix is $\Sigma_p = (\sigma^{-2}\mathbf{X}_{\text{NN}}^T\mathbf{X}_{\text{NN}} + \Sigma^{-1})^{-1}$. The final predictive distribution for $\mathbf{x}_i$ that employs a kernel mapping, $\mathbf{K}$, to solve the nonlinear problems

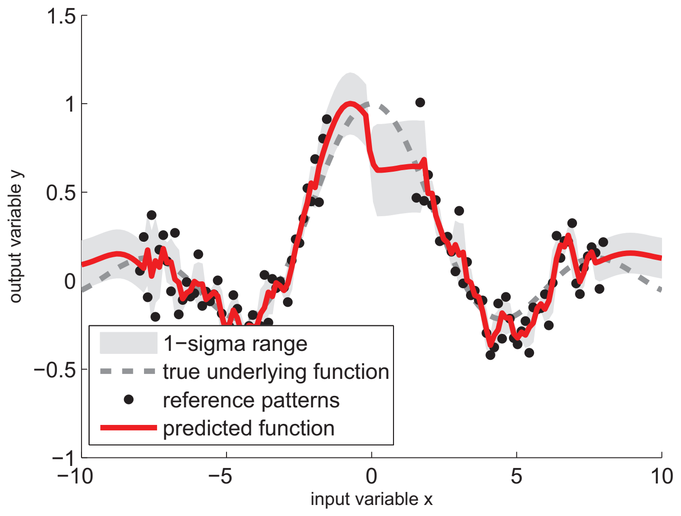

Fig. 3. Estimation of target variances from PLR regression with the RBF kernel.

is,

$$\begin{aligned} y_i &= \mu_p^T(\mathbf{y}_{\text{NN}} - \bar{y}_{\text{NN}}\mathbf{1}_{k\times 1}) + \bar{y}_{\text{NN}} \\ &= \mathbf{k}_i^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}_{\text{NN}} + \bar{y}_{\text{NN}}[1-\mathbf{k}_i^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{1}_{k\times 1}], \end{aligned} \quad (9)$$

$$\sigma_i^2 = \frac{1}{k}(\mathbf{y}_{\text{NN}} - \bar{y}_{\text{NN}}\mathbf{1}_{k\times 1})^T(\sigma^{-2}\mathbf{K}+\mathbf{I})^{-1}(\mathbf{y}_{\text{NN}} - \bar{y}_{\text{NN}}\mathbf{1}_{k\times 1}), \quad (10)$$

where $\bar{y}_{\text{NN}}$ is the mean value of the nearest neighbors' targets, and $\mathbf{1}_{k\times 1}$ is the $k \times 1$ column vector where every element is one.

Because the PLR output is formed as a Gaussian distribution for each test data point, a target value and its estimation variance can be obtained as the output for PLR for each test data point. If the test data point can be described with the training data (i.e., reference patterns) effectively, the target variance is small. Conversely, a significant target variance implies that the label of the test data point cannot be described by the training data. An example of PLR results with predicted target variance is depicted in Fig. 3. The estimation variance shown as a shaded area increases for an uncertain region.

In PLR, k, which represents the number of nearest neighbors, is an important parameter to set. PLR was designed to be less sensitive to k; however, risk for noisy data remains when estimating the label distribution. In order to overcome this risk and obtain robust results, two PLR models are employed. The first model, PLR$_{\text{local}}$, focuses on the local area of each unlabeled data point with small k. This model captures the local topology of the nearest labeled data for each unlabeled data. However, PLR$_{\text{local}}$ tends to be sensitive to noisy data. Hence, we employ a second model, PLR$_{\text{global}}$, that focuses on the global distribution of the labeled data with large k. PLR$_{\text{global}}$ might not capture the local topology as well as PLR$_{\text{local}}$; however, PLR$_{\text{global}}$ can be effective at capturing the global distribution of the labeled data. Hence, those two PLR models execute with a different k value: k_{local} and k_{global}. Fig. 4(a) depicts the output of PLR$_{\text{local}}$ that focuses on the local topology. Fig. 4(b) depicts the output of PLR$_{\text{global}}$ that captures the global distribution.

The outputs of both PLRs are obtained in Gaussian probabilistic distribution form, $\hat{y}_i = N(\bar{y}_i, \sigma^2)$. The proposed method combines both outputs into a single Gaussian probabilistic distribution form by the conjugation used in the Bayesian method (Bishop, 2006). In the conjugation, the prior and likelihood probabilities can be conjugated as the posterior probability. Because PLR$_{\text{local}}$ has a more data-focused perspective, it can be considered the likelihood of the Bayesian method. Conversely, because PLR$_{\text{global}}$ focuses on the global distribution of the underlying function, it can be the prior of the Bayesian method. The conjugated $\bar{y}$ and conjugated σ^2 can be calculated in Eq. (11), where n is the number of data used to calculate the likelihood term: PLR$_{\text{local}}$.

$$\bar{y}_{\text{conjugate}} = \frac{\frac{\bar{y}_{\text{global}}}{\sigma^2_{\text{global}}} + \frac{n\times\bar{y}_{\text{local}}}{\sigma^2_{\text{local}}}}{\frac{1}{\sigma^2_{\text{global}}} + \frac{n}{\sigma^2_{\text{local}}}}, \quad \sigma^2_{\text{conjugate}} = \frac{1}{\frac{1}{\sigma^2_{\text{global}}} + \frac{n}{\sigma^2_{\text{local}}}}. \quad (11)$$

3.2. *Data generation with oversampling from estimated label distribution*

The proposed method generates training data by oversampling based on the unlabeled data and their estimated label distribution obtained by the conjugation of 2-PLR results. Let us depict the unlabeled data as $\mathbf{x}_u$ and its estimated labeled distribution as $\hat{y}_u = N(\bar{y}_u, \sigma_u^2)$. Then, the unlabeled data with estimated label can be described as $U_u = \{(\mathbf{x}_u, N(\bar{y}_u, \sigma_u^2))\}$. The integrated dataset, D_I, can be constructed with the generated dataset of the unlabeled data, U_G, and the labeled data: $D_I = L\bigcup U_G$. The proposed method uses σ_u^2, the target variance of each unlabeled data point, to consider the uncertainty of the estimating labels during oversampling. The proposed method uses a different sampling rate for each unlabeled data point to be a proportion of σ_u^2. For the unlabeled data with high uncertainties, more training data should be generated in order to obtain more information for that region. Moreover, if σ_u^2 is large, which means the confidence of $\bar{y}_u$ is low, the single $\bar{y}_u$ has a low probability of being the label for $\mathbf{x}_u$. Hence, for those unlabeled data, the probability of being sampled should be high. Conversely, for the unlabeled data with low uncertainties, less training data is needed to express those unlabeled data. If the sampling rate were large for those unlabeled data, duplicated information might be created, which only increases training complexity. In this research, the sampling rate, p_u, is calculated by the scaling of σ_u^2, as in Eq. (12). Each unlabeled data point has t independent trials to be sampled. p_u determines whether an unlabeled data point is generated for every t trial, independently. Fig. 5 compares the integrated dataset with the labeled and unlabeled data with their single labels (a) and integrated dataset constructed using the labeled and generated data (b). The integrated dataset has more information for the unlabeled data located in uncertain regions, and prevents redundancies in certain regions. Moreover, the generated data are widely distributed near the margin space in order to support the ε-insensitive learning of SVR.

$$p_u = \frac{\sigma_u^2 - \min(\sigma^2)}{\max(\sigma^2) - \min(\sigma^2)}. \quad (12)$$

3.3. *Data selection and support vector regression*

SSL is a time-consuming method. The basic assumption of SSL is that the number of unlabeled data is very large, at least 10–100 times that of labeled data. Moreover, the proposed method employs data generation with oversampling. The size of the training dataset is even larger than the original training dataset. Hence, an additional method is required for decreasing training complexity. Because the proposed method is designed for SS-SVR, the training time complexity is based on SVR. The main issue of the training complexity of SVM is the amount of training data. The training complexity of SVR is strongly correlated to the quantity of training data: $O(n^3)$ for the training time complexity, and $O(n^2)$ for the training memory complexity, where n is the number of training data. Hence, the data selection method proposed in Kim and Cho (2012), EMPS, is employed in order to reduce the training complexity of SVR. Because the regression function of SVR is a linear combination of support vectors, SVR can construct the same regression results by training only on support vectors. Hence, the goal of EMPS is to select a small subset of training data expected

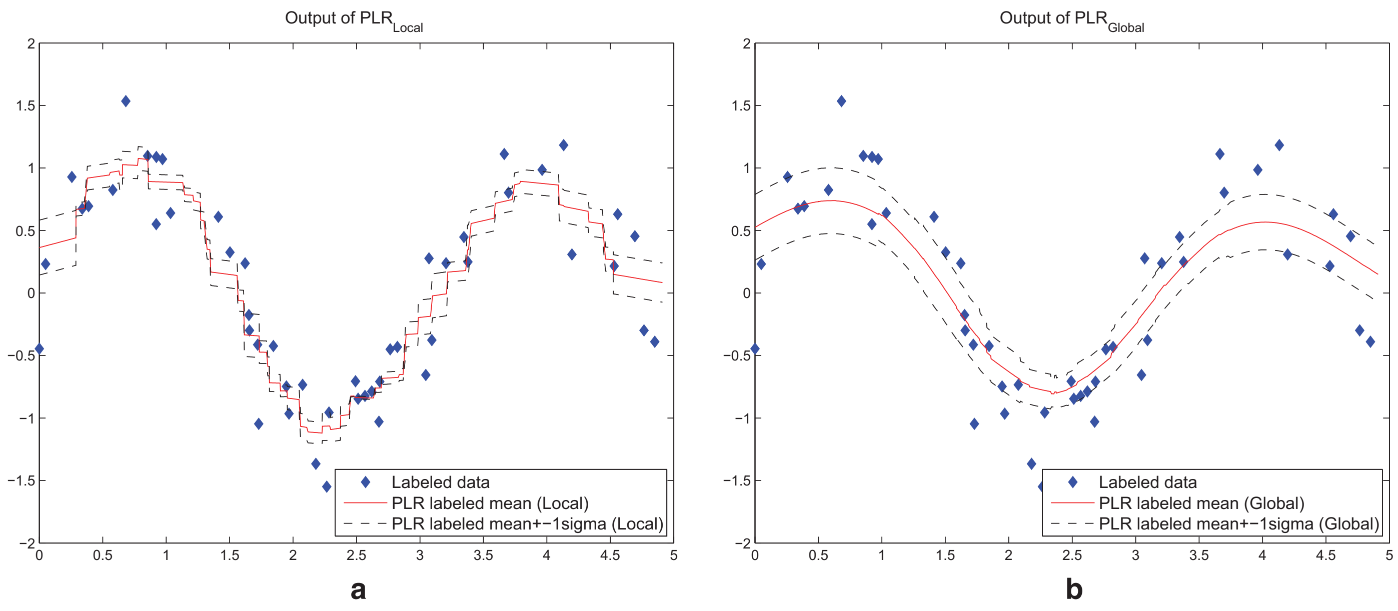

Fig. 4. The outputs of (a) PLR_{local} and (b) PLR_{global}.

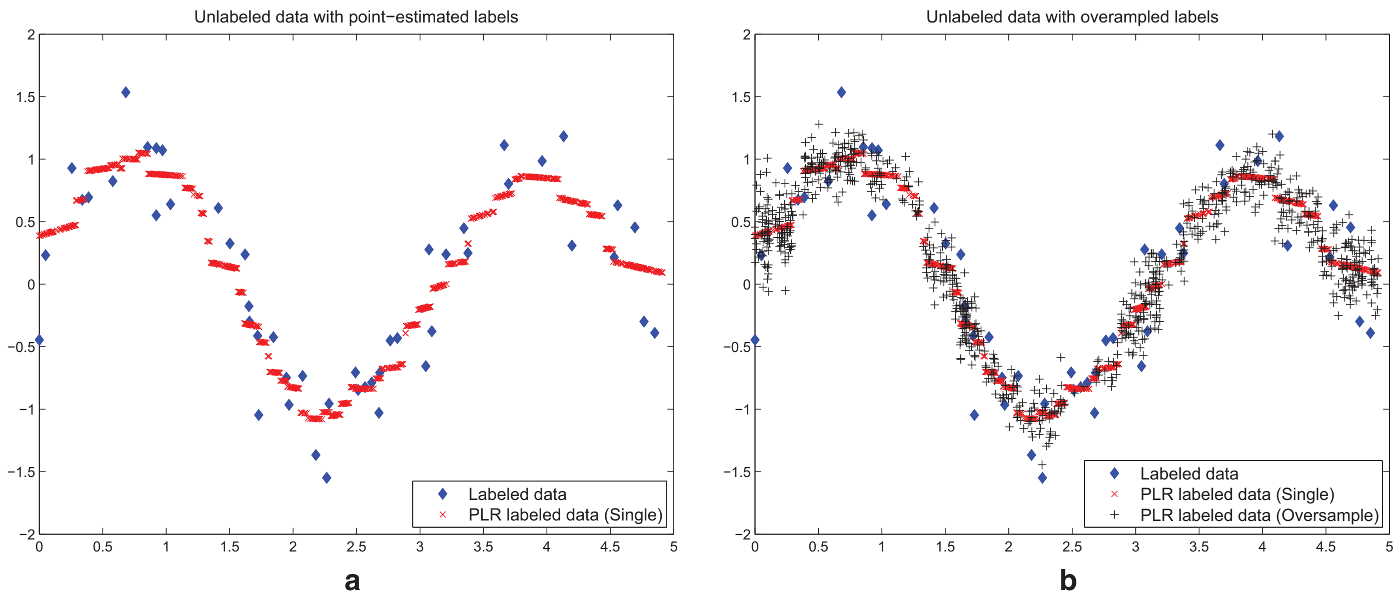

Fig. 5. Integrated dataset constructed with the unlabeled data ((a) single label values and (b) generated data).

to become support vectors. EMPS estimates the margin for all training data with multiple bootstrap learning, and selects those data whose margin is equal to or greater than the predefined SVR hyper-parameter ε. The training complexity of EMPS is $O(l^3)$, where l is the number of data in a bootstrap, and $l \ll n$. In Kim and Cho (2012), the experiment results show that the training time complexity of EMPS is approximately 38% that of the original SVR on average. The main procedure of EMPS is illustrated in Fig. 6. The algorithm for the proposed method is summarized in Fig. 7.

4. Experiment results

4.1. Experiment setting

For the experiments, 30 benchmark datasets for regression tasks, including 2 synthetic datasets and 28 real-world datasets, were selected. Real-world benchmark datasets were gathered from Delve datasets,[1] Time Series Data Library (TSDL),[2] and Statlib.[3] The datasets are summarized in Table 1.

The features of the regression datasets can be partitioned into three types: "Synthetic," "T.S.," and "Non-T.S.," indicating synthetic, time series, and non-time series multivariate, respectively. An artificial dataset was generated based on a mathematical function, $y = \sin(2x) + \xi$ where $x \in [0, 5]$ and $\xi \sim N(0, 0.5^2)$. Add10 is an artificial dataset gathered from the Delve datasets. We used only five relevant input features, excluding five noise terms. The time-series datasets were reformulated as regression problems using the previous ten values to estimate the following single value. This is a typical method for solving time-series problems. The Wind dataset was reformulated to estimate the wind speed of the Dublin station

[1] Delve dataset: http://www.cs.toronto.edu/~delve/data/datasets.html/.
[2] TSDL: http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/.
[3] Statlib: http://lib.stat.cmu.edu/datasets/.

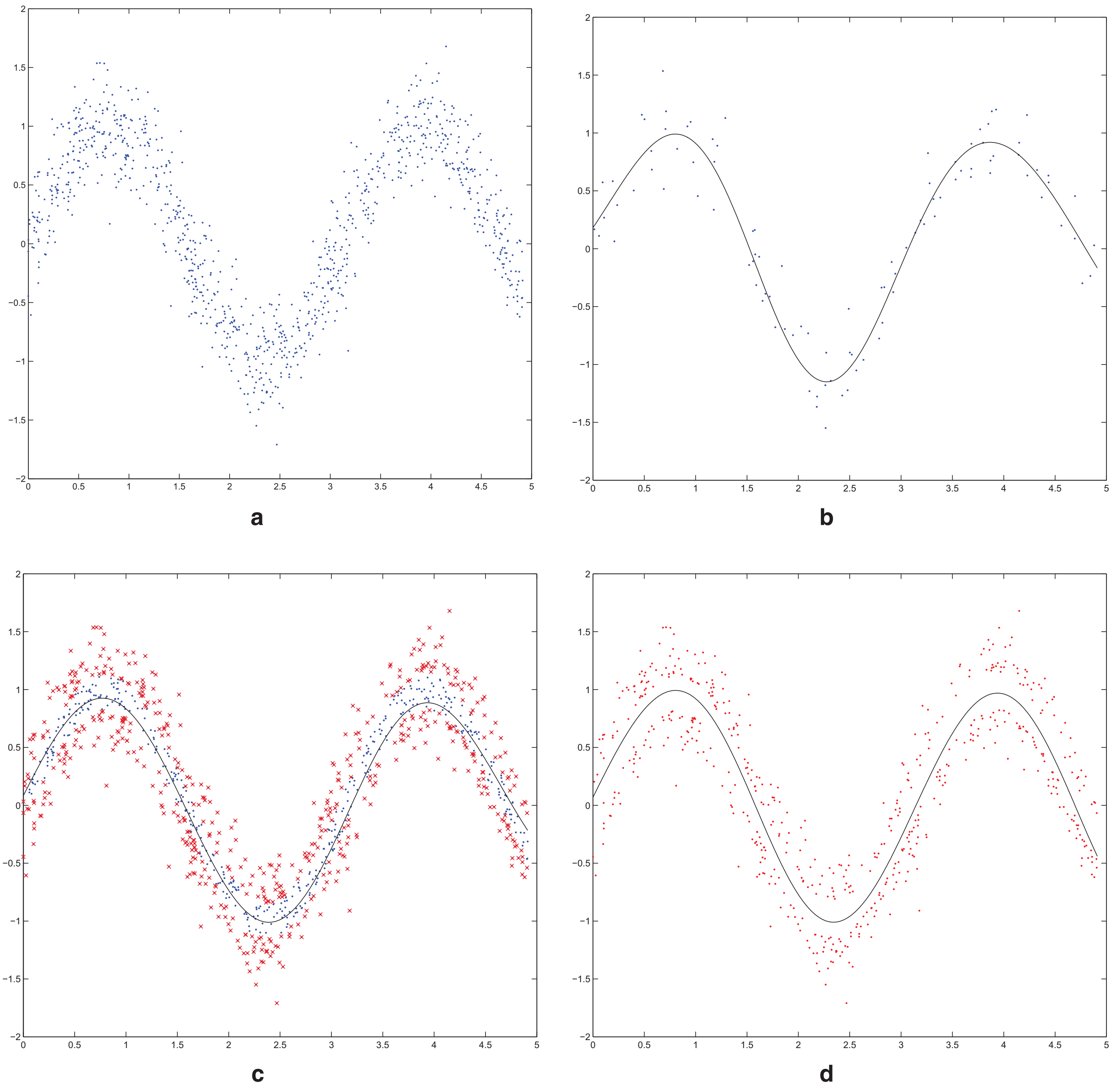

Fig. 6. (a) Original dataset, (b) a bootstrap sample with an SVR trained on it, (c) original dataset and data located outside the ε-tube marked as red and (d) selected data and the resulting trained SVR (reprinted from Kim & Cho, 2012). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

using the wind speeds of 11 other observed stations. Datasets 15–22 in Table 1 are from the three datasets: Bank, Pumadyn, and Census House. The number that follows the dataset name denotes the number of features used. FM, NH, L, and H represent “fairly linear-moderate noise,” “nonlinear-high noise,” “low task difficulty,” and “high task difficulty,” respectively. We evaluated the model performance against the different characteristics of the training datasets, such as linearity and training complexity. The DMEF4 dataset is a real-world marketing dataset. We preprocessed the DMEF4 dataset to be a regression task following Kim and Cho (2012) by randomly splitting into training and test data. In order to determine the hyper-parameters of SVR, we followed the previous research from Kim and Cho (2012). Ten-fold cross-validation was employed with C $\times$ ε = {0.1, 0.5, 1, 3, 5, 7, 10, 20, 50, 100} $\times$ {0.01, 0.05, 0.07, 0.1, 0.15, 0.3, 0.5, 0.7, 0.9, 1}. Basically, cross-validation for the parameter selection was conducted only for the training dataset of each experiment, while excluding the test dataset. Then, the training dataset was partitioned into ten folds randomly, and a conventional SVR model trained with data in nine folds evaluated data in the other fold. For some experiment settings having with fewer than ten labeled data, cross-validation was similar to the leave-one-out cross-validation. The parameter set that showed the

```
The Proposed SS–SVR Method Algorithm

  1. Initialization
     $k_{local}$: The number of nearest neighbors for PLR$_{local}$
     $k_{global}$: The number of nearest neighbors for PLR$_{global}$
     $t$: The number of trials for data generation
     $U_G$: An empty set for the unlabeled data

  2. Labeling the unlabeled data
     $N_{local} \leftarrow \mathrm{PLR}_{local}(\mathbf{x}_u, L, k_{local})$
     $N_{global} \leftarrow \mathrm{PLR}_{global}(\mathbf{x}_u, L, k_{global})$
     $N_{conjugate} \leftarrow \mathrm{Conjugation}(N_{local}, N_{global})$

  3. Data generation
     $p_u = \frac{\sigma_u^2 - min(\sigma)}{max(\sigma) - min(\sigma)}$
     FOR all $\mathbf{x}_u$
        FOR 1 to $t$
           $r \leftarrow Uniform(0, 1)$
           IF $p_u > r$
              $U_G \leftarrow U_G \bigcup (\mathbf{x}_u, \hat{y}_u \sim N(\bar{y}_u, \sigma_u^2))$
           END IF
        END FOR
     END FOR
     $D_I = L \bigcup U_G$

  4. Data selection
     $\{Select_x, Select_y\}$ =EMPS$(D_I)$

  5. Run SVR
     $\hat{y} = f(x) \leftarrow$SVR$(Select_x, Select_y)$
```

Fig. 7. The algorithm of the proposed SS-SVR method.

Table 1
Datasets used in experiments.

No.	Name	# Train	# Test	# Attribute	Origin	Feature
1	Artificial	1000	1000	1	Kim and Cho (2012)	Synthetic
2	Add10	2000	2000	5	Delve datasets	Synthetic
3	Santa Fe A	890	100	10	Santa Fe comp.	T.S.
4	Santa Fe D	2000	2000	10	Santa Fe comp.	T.S.
5	Santa Fe E	1490	500	10	Santa Fe comp.	T.S.
6	Sun spot	2000	1000	10	TSDL	T.S.
7	Melbourne temp.	2000	1000	10	TSDL	T.S.
8	Gold	700	300	10	TSDL	T.S.
9	S&P500	2000	2000	10	TSDL	T.S.
10	Wind	2000	2000	11	Statlib	Non-T.S.
11	No2	400	100	7	Statlib	Non-T.S.
12	Pollen	2000	1848	4	Statlib	Non-T.S.
13	Abalone	2000	2000	10	Delve datasets	Non-T.S.
14	Computer activity	2000	2000	12	Delve datasets	Non-T.S.
15	Bank 8FM	2000	2000	8	Delve datasets	Non-T.S.
16	Bank 8NH	2000	2000	8	Delve datasets	Non-T.S.
17	Pumadyn 8FM	2000	2000	8	Delve datasets	Non-T.S.
18	Pumadyn 8NH	2000	2000	8	Delve datasets	Non-T.S.
19	Census house 8L	2000	2000	8	Delve datasets	Non-T.S.
20	Census house 8H	2000	2000	8	Delve datasets	Non-T.S.
21	Census house 16L	2000	2000	16	Delve datasets	Non-T.S.
22	Census house 16H	2000	2000	16	Delve datasets	Non-T.S.
23	Boston housing	400	106	13	UCI	Non-T.S.
24	Wine quality red	1200	399	11	UCI	Non-T.S.
25	Wine quality white	4000	898	11	UCI	Non-T.S.
26	Skillcraft	2000	1338	18	UCI	Non-T.S.
27	Concrete	700	300	8	UCI	Non-T.S.
28	Parkinson monitor	2000	2000	16	UCI	Non-T.S.
29	Power plant	2000	2000	4	UCI	Non-T.S.
30	DMEF4	4786	4785	15	DMEF foundation	Non-T.S.

Table 2
Benchmark methods for experiments.

Method	Labeling unlabel data	Learning process	Final model
COREG$_{k\mathrm{NN}}$ (Zhou & Li, 2007)	k-NN (local topology method)	Iteration with incremental data addition	k-NN
COREG$_{\mathrm{SVR}}$ (Zhou & Li, 2007)	k-NN (local topology method)	Iteration with incremental data addition	SVR
Co-SVR (Wang et al., 2010)	SVR (function-based method)	Iteration with incremental data addition	SVR
L-SVR (Pozdnoukhov & Bengio, 2006)	–	Non-iterative (kernel deformation with Laplacian)	SVR
Proposed	PLR	Non-iterative (oversampling)	SVR

best accuracy was selected for each dataset. The RBF kernel was used as a kernel function and the kernel parameter σ was fixed to 1.0 for all datasets. Although the kernel parameter is known to be sensitive to the experiment results, some previous studies have suggested that when comparing multiple benchmark methods, using the fixed kernel parameter for normalized dataset is acceptable (Kim & Cho, 2012; Kim, Lee, & Cho, 2008). All input features of the 30 datasets were normalized to be the standard scores.

The proposed method was compared with COREG$_{k\mathrm{NN}}$, COREG$_{\mathrm{SVR}}$, Co-SVR, and L-SVR (see Table 2). COREG was proposed by Zhou and Li (2007). Both COREG-based methods employ k-NN for labeling the unlabeled data. The difference between COREG$_{k\mathrm{NN}}$ and COREG$_{\mathrm{SVR}}$ is that COREG$_{k\mathrm{NN}}$ and COREG$_{\mathrm{SVR}}$ employ k-NN and SVR for the final training function, respectively. The value of k was set to three and five for the two k-NN based models used for COREG. Co-SVR is obtained from Wang et al. (2010). Co-SVR uses SVR as the base learner, whereas COREG uses k-NN as the base learner. With the experiment results from COREG$_{\mathrm{SVR}}$ and Co-SVR, we expect the difference between the local topology and function-based models as base learners of SSL to be comparable. The hyper-parameters for Co-SVR were set to the same values used for the original training dataset. The genetic algorithm (GA)-based feature selection part for Co-SVR was omitted in order to reduce training complexity. For the initialization of both co-training based methods, the labeled data were randomly partitioned into two training sets. The number of data in the working set of the unlabeled data was set to 100 and 40 for COREG and Co-SVR, respectively. The number of maximum iterations was set to 100 for the co-training based methods. We also implemented a manifold regularization-based method from Pozdnoukhov and Bengio (2006) (L-SVR) by modifying the original source code from Sindhwani et al. (2005). The number of nearest neighbors selected was set to three, γ for controlling the effect of the Laplacian matrix was set to 0.5. The proposed method also uses parameters. The number of nearest neighbors for 2-PLR was set to five and ten for k_{local} and k_{global}, respectively. The number of trials for each unlabeled data point in the data generation, t, was set to three, five, and seven. The parameters for EMPS were set to ten for all experiments.

The original training data were sampled randomly to be the labeled data; the remainder was used as the unlabeled data. The labels for the unlabeled data were masked and not used in the training and test. The ratios of the labeled data were 20%, 10%, 5%, and 1% of the original data. Because the experiment results could be biased by samplings for the labeled section and data generation, the experiment results were averaged over ten-time repetitions. The performance of each method was measured using root mean squared error (RMSE) of the test dataset that was entirely excluded during training. In order to measure the computational complexity, training time (in seconds) was recorded during all the steps of model training, including the SSL procedures.

4.2. Experiment results

Figs. 8–11 present the experiment results for the 30 datasets. Figs. 8 and 9 illustrate the experiment results for each dataset in terms of RMSE for all methods varied by labeled ratio from 20% to 1%. The experiment results of COREG$_{k\mathrm{NN}}$, COREG$_{\mathrm{SVR}}$, Co-SVR, and L-SVR are plotted as upper triangles, lower triangles, x-marks, and cross-marks, respectively. The number after “Proposed” indicates parameter t. The experiment results for the proposed method with various t are plotted as squares, diamonds, and circles. The RMSE trend tends to increase with a smaller labeled ratio. In terms of RMSE, COREG$_{k\mathrm{NN}}$ and the proposed method result in better performance compared with the others. For some datasets, COREG$_{k\mathrm{NN}}$ shows the most accurate result; however, the accuracy of the proposed method is comparable. The proposed method shows low RMSE consistently with various types of datasets. Moreover, the proposed method outperforms the methods that employ SVR as the final model: COREG$_{\mathrm{SVR}}$, Co-SVR, and L-SVR. Meanwhile, Figs. 10 and 11 illustrate the experiment results for each dataset in terms of the training time for all methods varied by labeled ratio from 20% to 1%. The training time of the proposed method is shorter than the others, whereas COREG$_{k\mathrm{NN}}$ records a long training time. The training time of the co-training methods tends to be sensitive to the labeled ratio because the number of training data for each base learner and convergence condition largely affect the efficiency of co-training. The training time for the proposed method is approximately 30% that of COREG$_{kNN}$. The proposed method demonstrates better efficiency than the other methods. Combining the experiment results of RMSE with the training time, the proposed method shows remarkable results. The RMSE of the proposed method is lower than that of the others, with the exception of COREG$_{k\mathrm{NN}}$, and the training time is the shortest for all datasets. For the proposed method, parameter t is not sensitive to most datasets. It seems that for even a small number of trials, data generation can ensure an effective range for the training space.

Fig. 12 illustrates the RMSE ratio for all datasets. Fig. 13 illustrates the training time ratio for the same. In order to summarize all experiment results in a figure, such results should be scaled. RMSE ratio is the fraction of a method's RMSE to the RMSE of only the trained labeled data. For example, the RMSE ratio of the proposed method is calculated as $\text{RMSE ratio} = \frac{\text{RMSE}_{\text{proposed}}}{\text{RMSE}_{\text{labeled}}} \times 100$. The smaller the RMSE ratio, the better is the method performed. The RMSE ratio measures SSL improvement over conventional supervised learning. The RMSE ratio of the proposed method is stable regardless of the labeled ratio and datasets. In the interest of comparing training efficiency, the training time of other benchmark methods is compared with the proposed method. The proposed method outperforms COREG$_{\mathrm{SVR}}$, Co-SVR, and L-SVR, and it is comparable to COREG$_{k\mathrm{NN}}$. The training time of the proposed method is the best among the other benchmark methods.

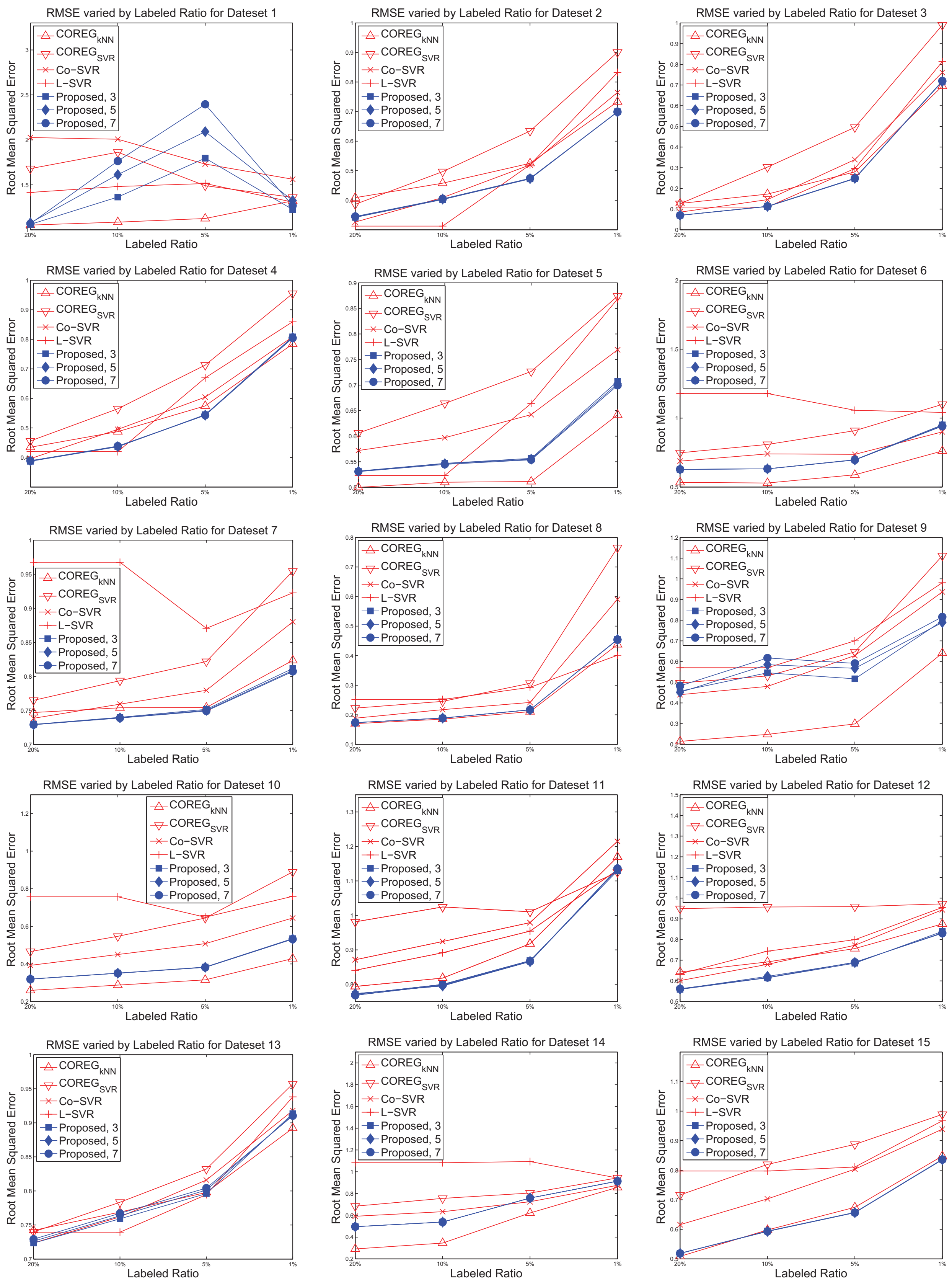

Fig. 8. Experiment results in terms of RMSE for Dataset 1–15.

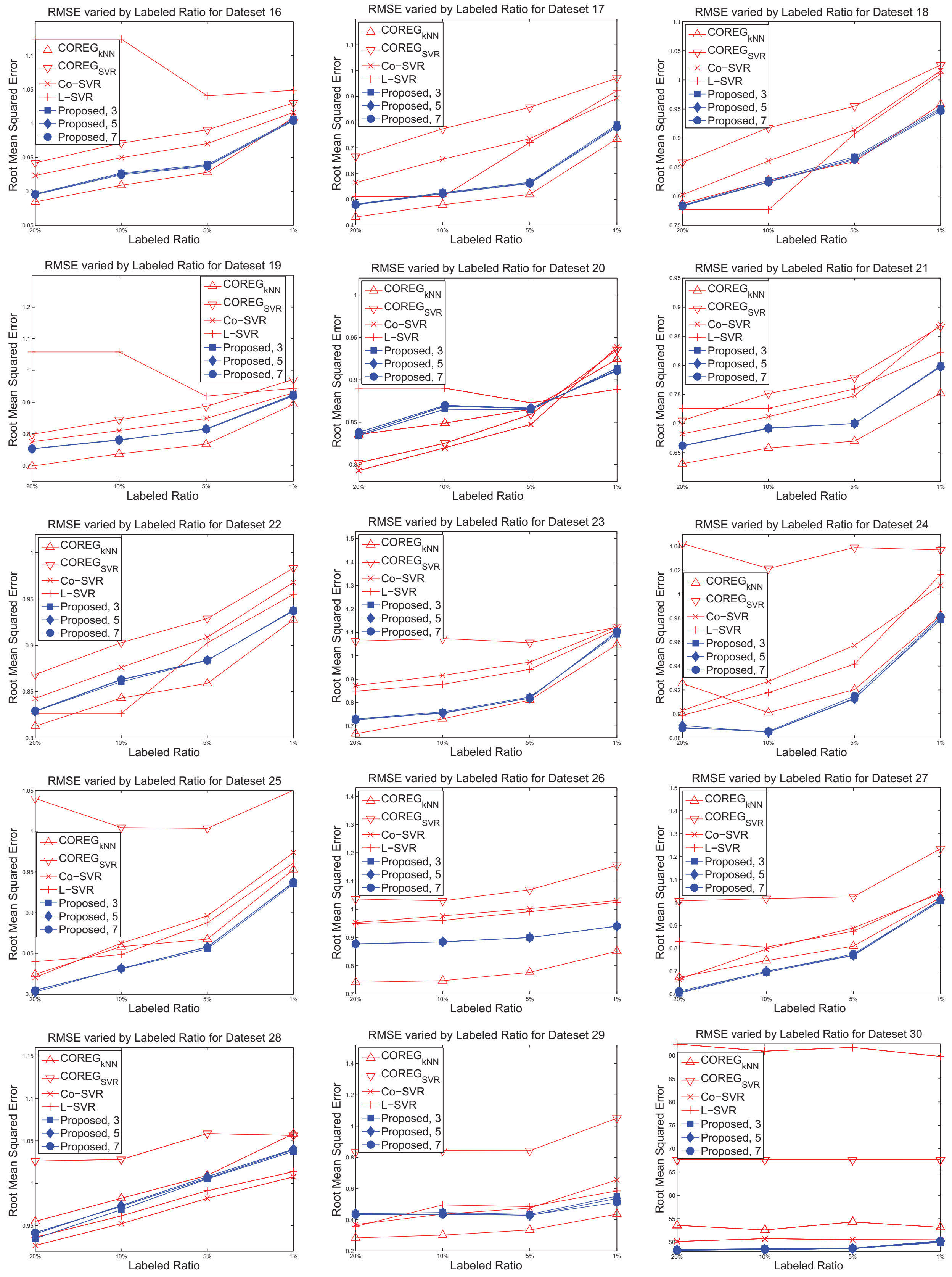

Fig. 9. Experiment results in terms of RMSE for Dataset 16–30.

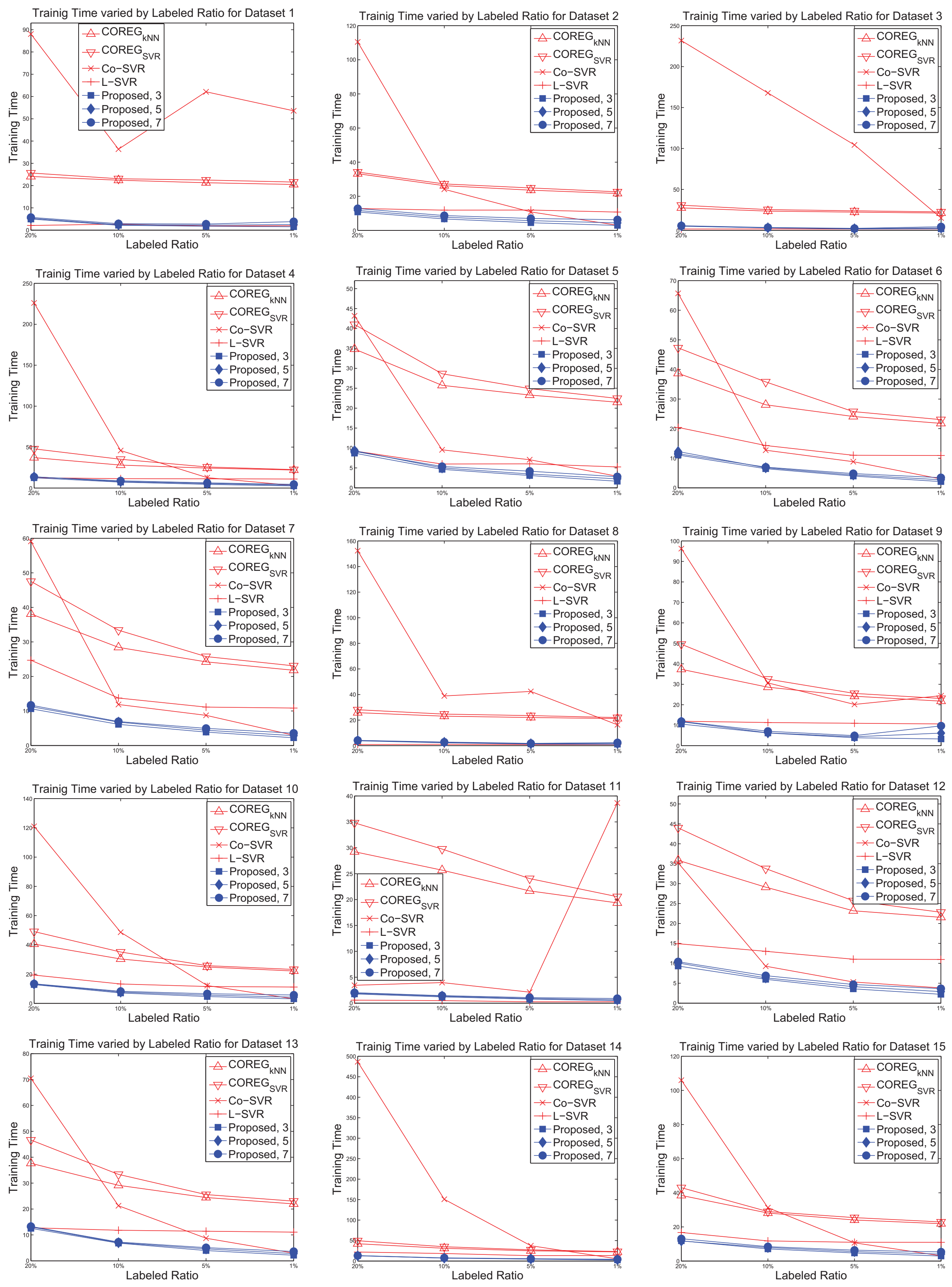

Fig. 10. Experiment results in terms of training time for Dataset 1–15.

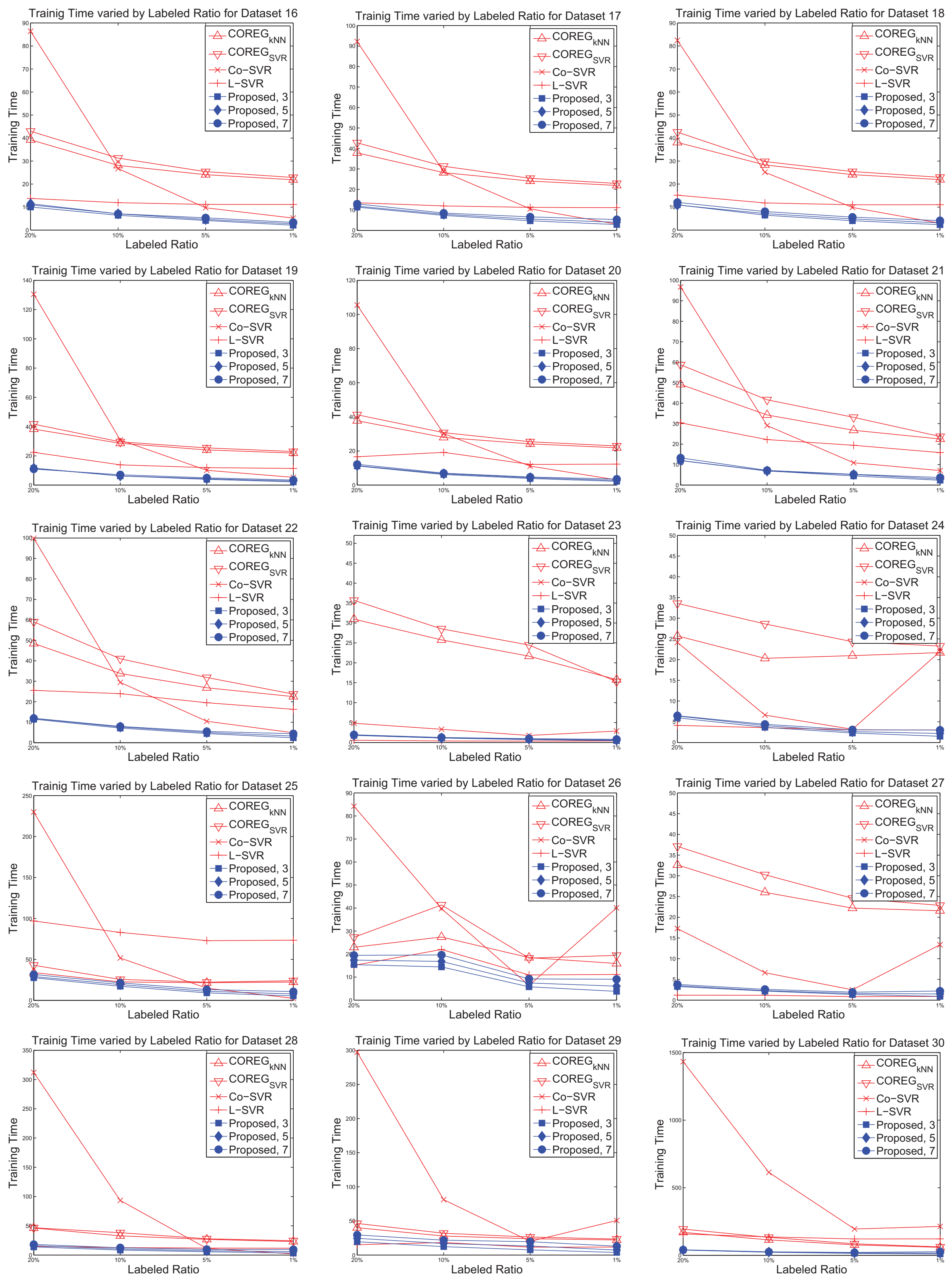

Fig. 11. Experiment results in terms of training time for Dataset 16–30.

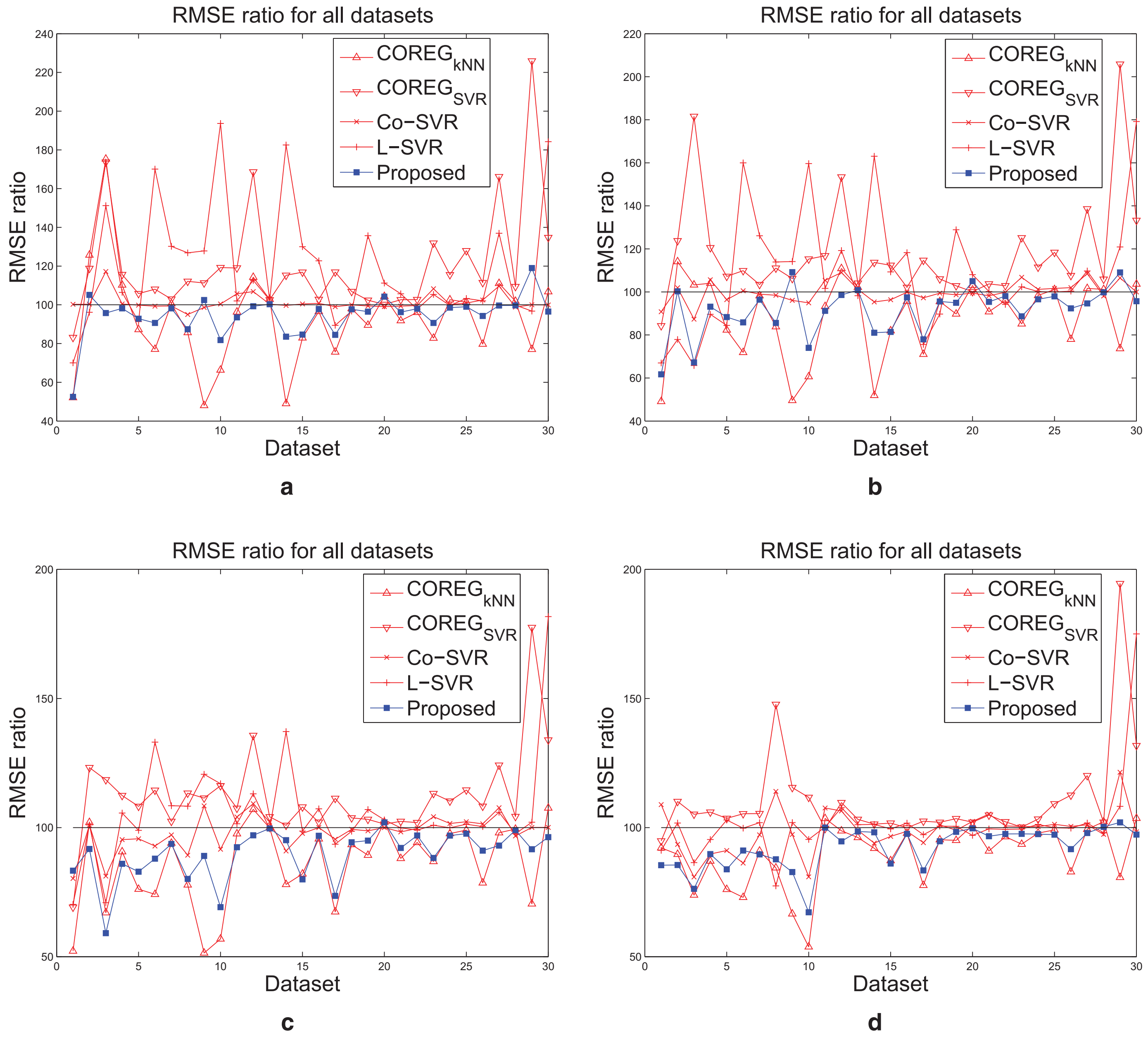

Fig. 12. RMSE ratio for all datasets, (a) $L = 20\%$, (b) $L = 10\%$, (c) $L = 5\%$, and (d) $L = 1\%$.

Table 3
Summary of the experiment results (rank for RMSE).

	$L = 20\%$	$L = 10\%$	$L = 5\%$	$L = 1\%$
COREG$_{k\text{NN}}$	15 (16)	16 (17)	14 (16)	16 (16)
COREG$_{\text{SVR}}$	1 (1)	0 (2)	0 (2)	0 (1)
Co-SVR	1 (3)	2 (3)	2 (5)	2 (6)
L-SVR	2 (5)	1 (3)	6 (10)	2 (6)
Proposed, 3	5 (16)	5 (20)	3 (1)	5 (20)
Proposed, 5	1 (24)	3 (24)	2 (21)	1 (23)
Proposed, 7	4 (25)	3 (21)	3 (18)	4 (18)

Table 4
Summary of the experiment results (rank for training time).

	$L = 20\%$	$L = 10\%$	$L = 5\%$	$L = 1\%$
COREG$_{k\text{NN}}$	0 (0)	0 (0)	0 (0)	0 (0)
COREG$_{\text{SVR}}$	0 (0)	0 (0)	0 (0)	0 (0)
Co-SVR	3 (12)	0 (1)	0 (0)	0 (0)
L-SVR	5 (7)	5 (8)	6 (8)	9 (13)
Proposed, 3	22 (30)	25 (30)	23 (30)	16 (30)
Proposed, 5	0 (30)	0 (30)	1 (30)	4 (28)
Proposed, 7	0 (11)	0 (21)	0 (22)	1 (19)

We summarize the overall experiment results in terms of ranks and averages. Tables 3 and 4 summarize the ranks of each method among a total of seven methods for 30 datasets. Each number indicates the rank number assigned to the method for best performance, whereas each number in brackets indicates the rank number assigned to a method in the top three of each dataset. The rank number for COREG$_{k\text{NN}}$ in the best RMSE is greater than for the others. COREG$_{k\text{NN}}$ shows the best RMSE for approximately 15 datasets among 30 for each labeled ratio. However, the proposed method shows comparable results. The proposed method ranks in the top three RMSE for most datasets, which is better than the others, including COREG$_{k\text{NN}}$. The accuracy of the proposed method is stable to various types of datasets. Among the SVR-based methods, the proposed method results in the best accuracy. In terms of training time, the proposed method has the best results. The proposed method shows the shortest training time for most datasets, which

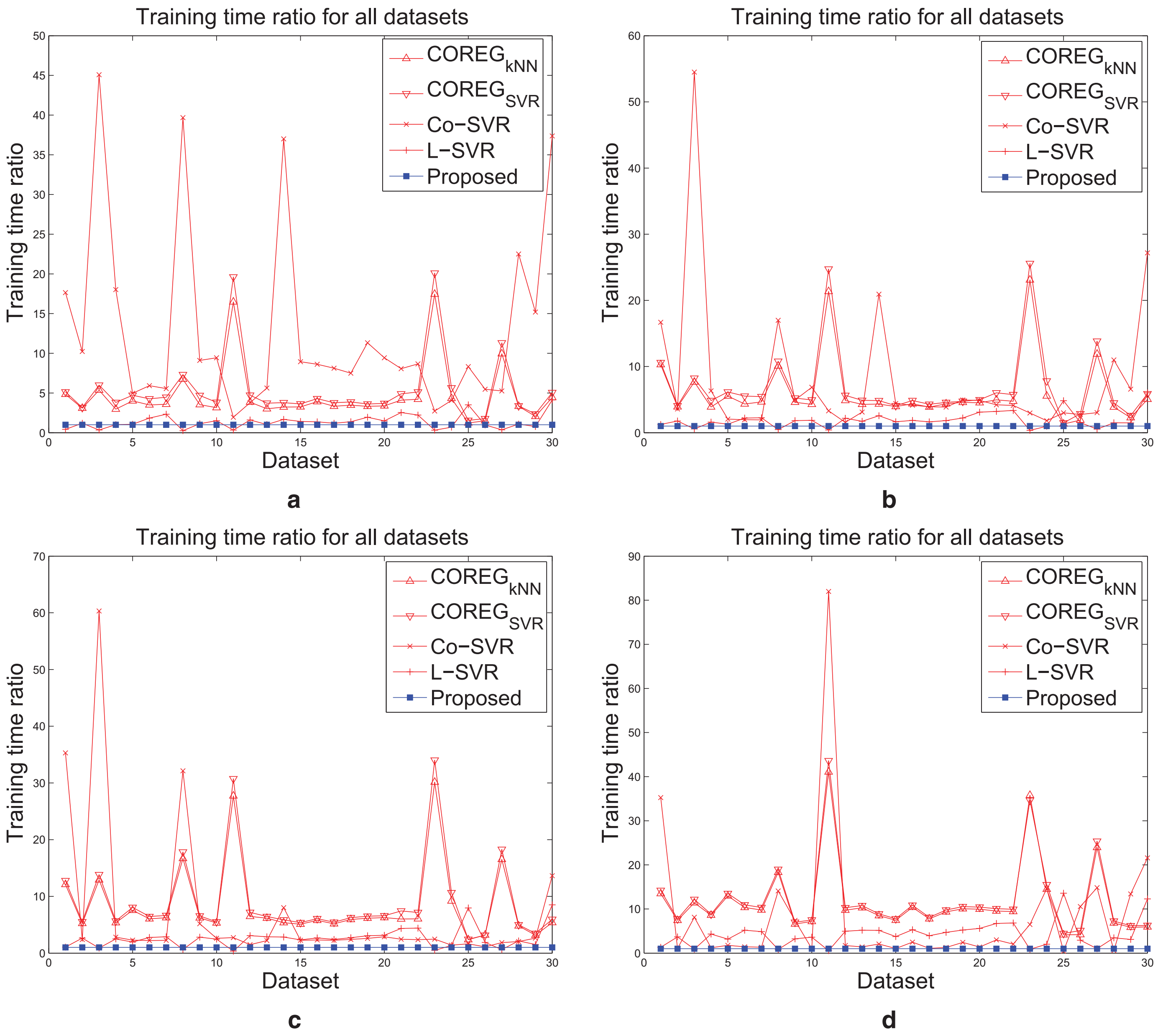

Fig. 13. Training time ratio for all datasets, (a) $L = 20\%$, (b) $L = 10\%$, (c) $L = 5\%$, and (d) $L = 1\%$.

Table 5
Summary of the experiment results (average of RMSE ratio). (The boldface numbers mean just average values over all experiments for each method.)

	$L = 20\%$	$L = 10\%$	$L = 5\%$	$L = 1\%$	Avg.
COREG$_{kNN}$	92.96	88.05	85.81	89.27	**89.02**
COREG$_{SVR}$	121.02	118.06	111.91	110.38	**115.34**
Co-SVR	101.37	99.85	98.07	98.83	**99.53**
L-SVR	119.36	109.21	106.06	101.43	**109.05**
Proposed, 3	94.61	91.79	89.71	92.19	**92.07**
Proposed, 5	94.70	91.48	90.40	92.23	**92.20**
Proposed, 7	94.89	91.83	90.03	91.97	**92.18**
Proposed w/o oversamples	99.79	92.77	89.36	95.47	**94.34**

Table 6
Summary of the experiment results (average of training time). (The boldface numbers mean just average values over all experiments for each method.)

	$L = 20\%$	$L = 10\%$	$L = 5\%$	$L = 1\%$	Avg.
COREG$_{kNN}$	40.40	30.45	25.30	22.61	**29.69**
COREG$_{SVR}$	47.05	35.15	27.38	23.86	**33.36**
Co-SVR	168.76	57.95	22.79	16.15	**66.41**
L-SVR	20.87	17.38	15.14	14.82	**17.17**
Proposed, 3	11.47	6.96	4.24	2.61	**6.32**
Proposed, 5	12.22	7.69	5.18	3.89	**7.24**
Proposed, 7	12.92	8.52	6.18	5.57	**8.29**
Proposed w/o oversamples	6.69	5.74	5.82	4.96	**5.80**

verifies the efficiency of the proposed method. Tables 5 and 6 summarize the experiment results for each method for 30 datasets on average. The proposed method outperforms the methods that employ SVR as the final model: COREG$_{SVR}$, Co-SVR, and L-SVR. RMSE for the proposed method is approximately 3% points higher than COREG$_{kNN}$. However, as indicated in Table 6, the training time of the proposed method is only 20–25% that of COREG$_{kNN}$. Because training time complexity of the base learner for Co-SVR is greater than that for COREG, the training time for Co-SVR is more sensitive to the number of training data. Moreover, the training time for Co-SVR is highly dependent on the convergence condition. Sometimes, an unlabeled data point evaluated by a function-based base learner

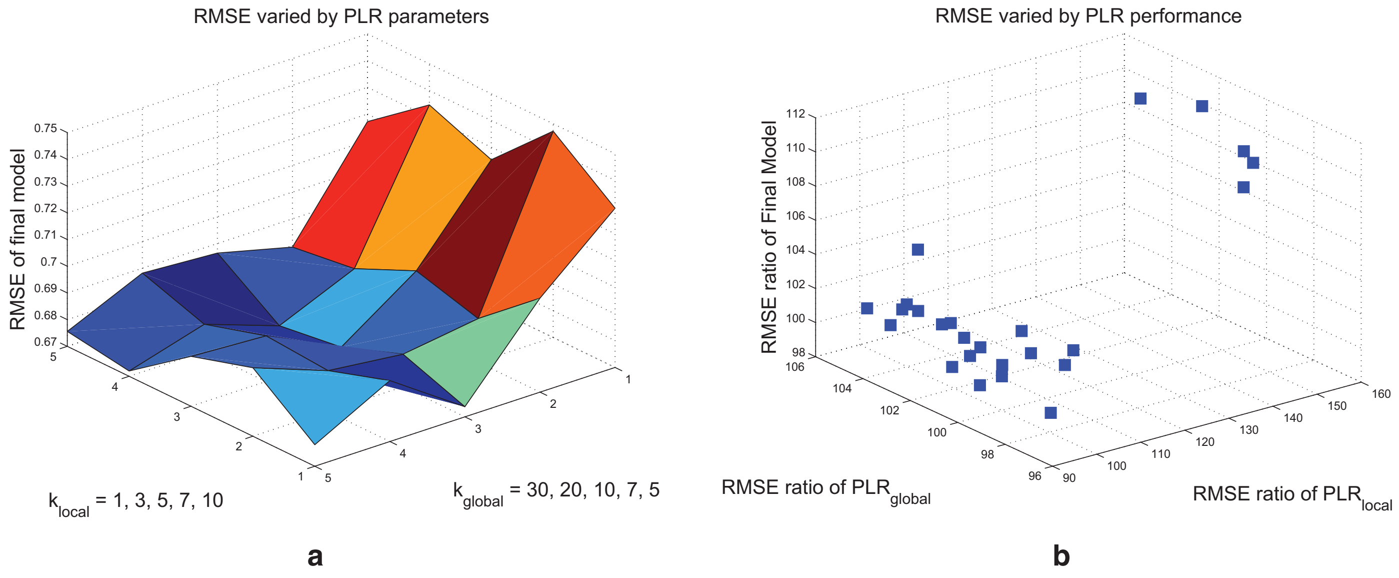

Fig. 14. Model performance varied by (a) PLR parameters and (b) PLR performance.

might not improve the base model, and unlabeled data are rarely added to the training dataset. "Proposed w/o oversamples" is the experiment result of the proposed method without oversampling in the data generation stage. As demonstrated in Table 5, oversampling in the data generation stage can improve accuracy, which is one of the main contributions of this paper.

4.3. Parameter analysis

The proposed method contains multiple steps. In order to identify whether each step can contribute to the final model performance, we conducted experiments with various model parameters. First, the proposed method employs two PLR models to estimate the label distributions for the unlabeled data. The PLR model has two parameters: k_{local} and k_{global}. Fig. 14(a) depicts the final performance of the proposed method varied by k_{local} and k_{global}. The experiments were conducted for the combination $k_{local} \times k_{global} = \{1, 3, 5, 7, 10\} \times \{5, 7, 10, 20, 30\}$. The model performances were averaged for 30 experiment datasets. The best performance was obtained when $\{k_{local}, k_{global}\}$ were {5, 20}, {5, 10}, and {10, 20}. When the parameters were not set properly, the final RMSE increased approximately 10% from the best model. In addition, relationship between PLR performance and the final model was investigated. Fig. 14(b) shows the RMSE ratio for the final models with regard to various RMSE ratios of each PLR model. The performance of PLR_{local} seems very sensitive to the final model performance, whereas the performance of PLR_{global} seems less sensitive. Because PLR_{local} is employed to capture the local topology of the data space, it makes sense that the RMSE for PLR_{local} significantly affects the final model performance. Based on these experiment results, we can conclude that the performance of the first step, which employs two PLR models, affects the final performance of the proposed method.

Because we provided experiment results with various t in Section 4, we can finally compare the experiment results with or without EMPS. Fig. 15 illustrates the RMSE and training time ratios of the proposed method without EMPS compared with the proposed method with EMPS. The gap between the horizontal line at 1 (dashed line) and the RMSE ratio (blue line with cross-marks) indicates the change of RMSE by excluding EMPS. The gap between the horizontal line at 1.0 and the training time ratio (red line with x-marks) indicates the change of training time by excluding EMPS. When the RMSE ratio is smaller than 1.0, EMPS degrades the model accuracy. When the training time ratio is greater than 1, EMPS improves training efficiency. For some datasets, RMSE for the proposed method without EMPS is smaller than that with EMPS. However, the model accuracies with or without EMPS are comparable for most datasets. Meanwhile, the training time ratio is sensitively increased when EMPS is removed. With EMPS, the training time decreases by approximately about 12%; however, RMSE increases only 0.5% on average for all experiments.

5. Experiment results for virtual metrology

5.1. Virtual metrology

Semiconductor manufacturing consists of hundreds of different manufacturing processes, such as photolithography (photo), etching, diffusion, and so on. A wafer that contains thousands of semiconductors is processed according to recipes through manufacturing processes (Kim et al., 2015). A metrology process is employed after each manufacturing process in order to inspect the wafer quality indicators, such as the critical dimension of the etching process or the axes distortion of the photo process (Kang et al., 2011). However, because actual metrology processes require additional costs, increased human resources, and longer cycle times, only a few wafers are sampled for inspection (Chang, Kang, Hsu, Chang, & Chan, 2006; Cheng & Cheng, 2005). To overcome this limitation, VM is proposed to predict metrology values with a mathematical model trained by the fault detection and classification (FDC) data collected from sensors in process equipment and previous metrology values (Besnard & Toprac, 2006). Because a few wafers have labels by inspection of the actual metrology step, most wafers only have FDC data without labels. VM can decrease manufacturing costs by removing the actual metrology steps, and increase manufacturing quality by providing metrology values for all wafers. The bottom of Fig. 16 illustrates the concept of VM, whereas the top depicts the actual metrology. Nonlinear regression methods, including artificial neural networks and SVR, are commonly employed for VM modeling.

5.2. Experiment setting

A real-world semiconductor manufacturing dataset was collected from two units of equipment (EQ) from the photo process of a South Korean semiconductor manufacturing company. Seven

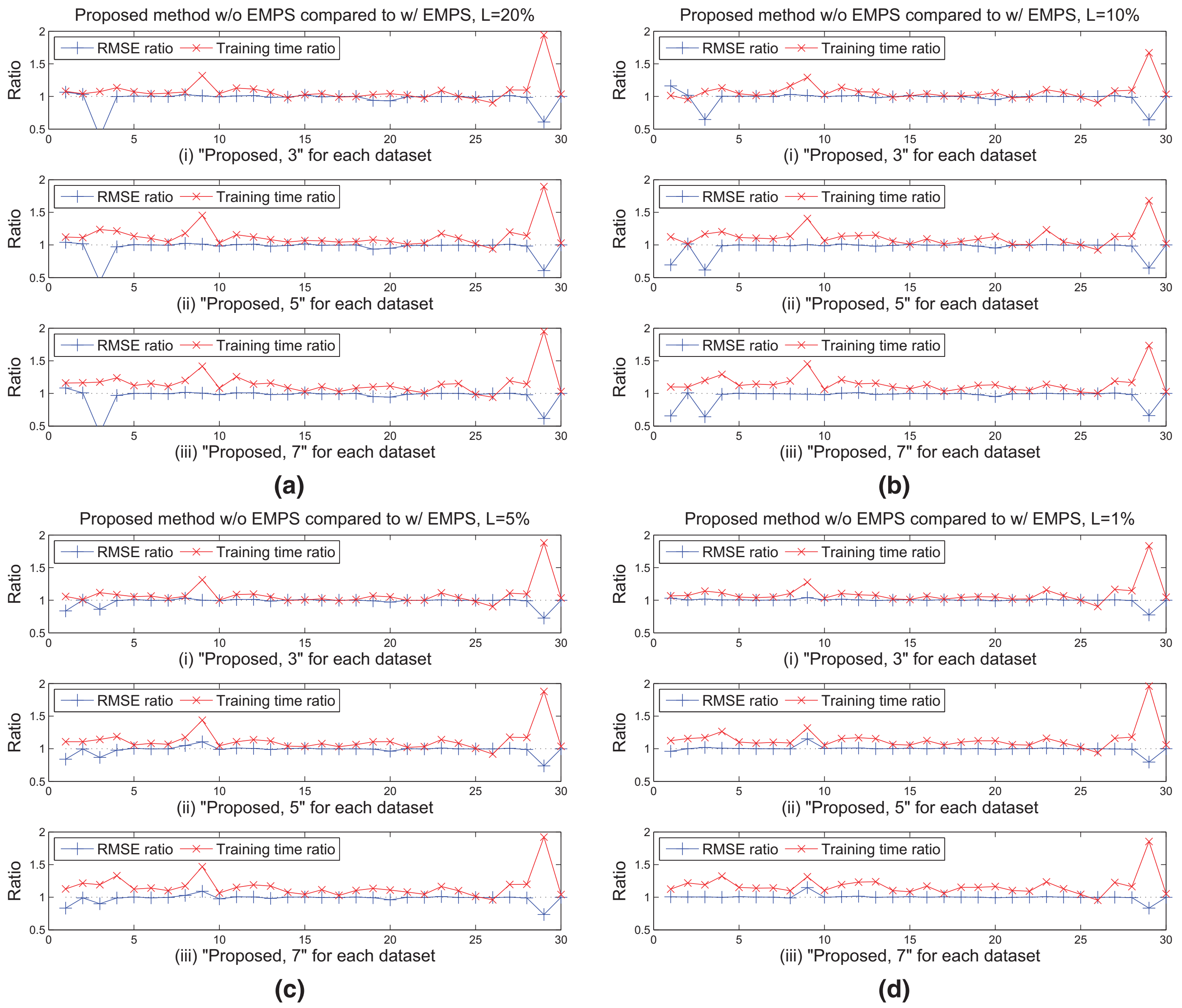

Fig. 15. Experiment result of the proposed method without EMPS compared to with EMPS, (a) $L = 20\%$, (b) $L = 10\%$, (c) $L = 5\%$, and (d) $L = 1\%$.

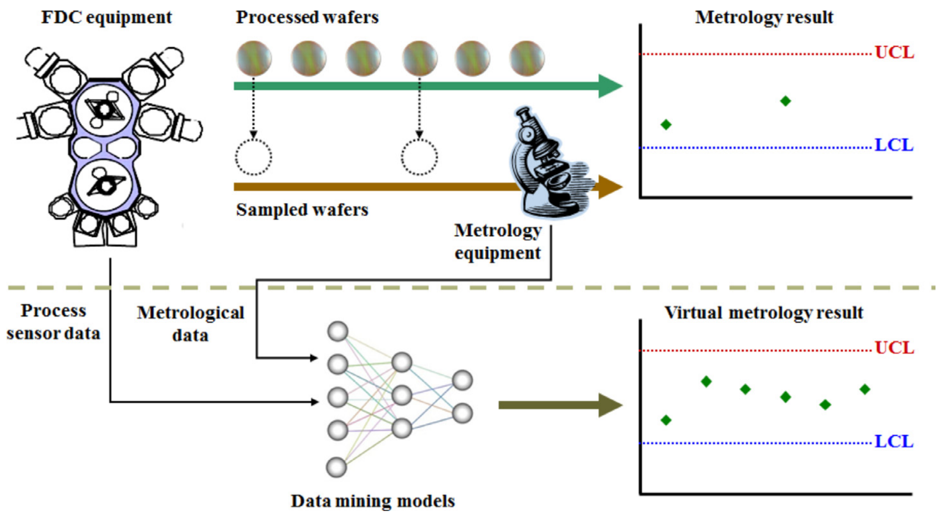

Fig. 16. Concept of the actual metrology (upper) and virtual metrology (lower) (reprinted from Kim et al., 2015).

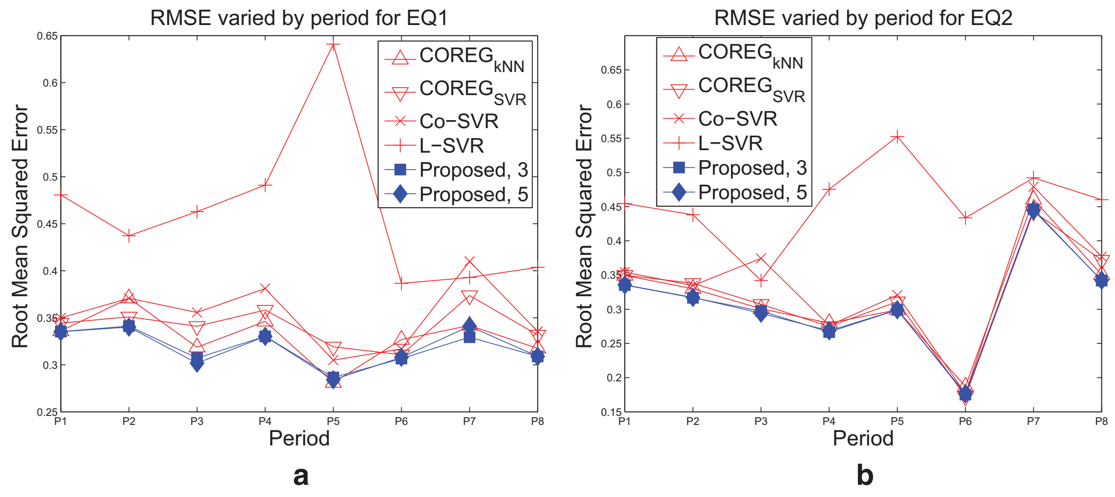

Fig. 17. Experiment results in terms of RMSE for VM.

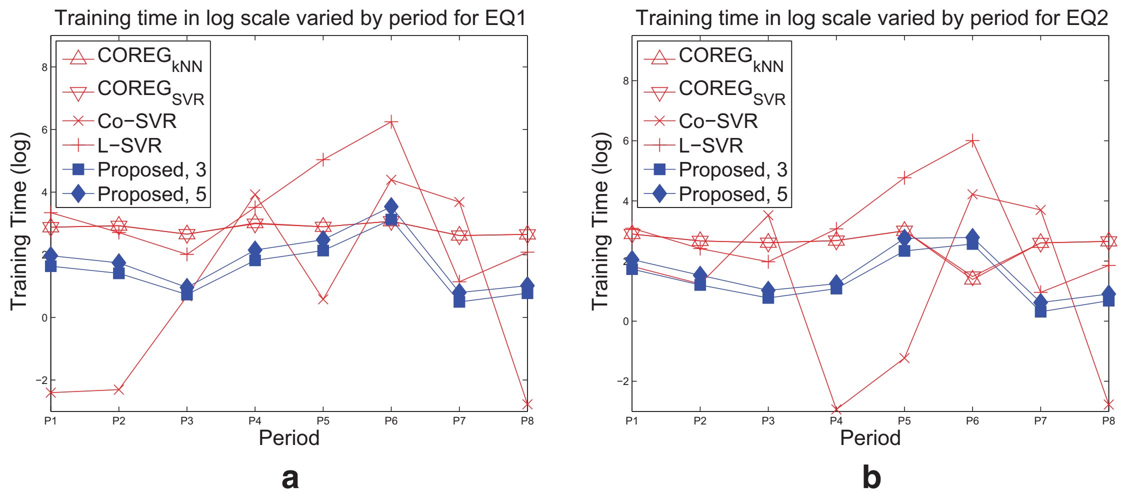

Fig. 18. Experiment results in terms of (log) training time for VM.

Table 7
VM dataset.

Process equipment	Period	# of the labeled data	# of the unlabeled data	# of FDC variables (input variables)
EQ1	P1	230	3670	117
	P2	172	2872	117
	P3	137	2313	112
	P4	167	3870	112
	P5	452	6546	112
	P6	818	9929	112
	P7	138	1654	112
	P8	195	2325	112
EQ2	P1	226	3377	117
	P2	180	2639	117
	P3	136	2293	112
	P4	170	3398	112
	P5	450	6008	112
	P6	816	9100	112
	P7	138	1523	112
	P8	195	2132	112

preventive maintenance (PM) processes were performed during the data collection period. PM caused significant changes in the recipe and the sensors of the processing equipment. Hence, the original dataset was partitioned into eight datasets (P1–P8) by PM period in order to construct independent VM models for each instance of PM. The number of data in the dataset is summarized in Table 7.

The features were selected by the GA (Mitchell, 1996) with the number of populations, maximum iterations, crossover rate, and mutation rate set to 100, 300, 0.5, and 0.03, respectively, as in Kang et al. (2011). The target variable for this dataset was an overlay mis-alignment of wafers after a photo process.

Five-fold cross-validation was used for the performance evaluation. The VM dataset that consists of both the labeled and unlabeled data was randomly partitioned into five folds. A model was trained with data from four folds, and the labeled data of the other fold was used as the test data. The performance of each method was measured using RMSE of the cross-validation results. Because RMSE measures the average of the difference between the actual target and estimated target values using VM, it is commonly used in actual semiconductor manufacturing. The computational complexity was measured using the training time in seconds during training, including the SSL procedure. Similar to Section 4, COREG$_{kNN}$, COREG$_{SVR}$, Co-SVR, and L-SVR were implemented as benchmark methods. The proposed method with different parameters, $t = \{3, 5\}$, was experimented.

5.3. Experiment results

Fig. 17 presents the experiment results in terms of RMSE for each EQ varied by periods of VM data. Fig. 18 presents the experiment results in terms of training time. For readability, a log-scaled training time is illustrated in Fig. 18. The experiment results

Table 8
RMSE of the experiment results for EQ1. (The boldface numbers mean the best method for each dataset (P1 ∼ P8).)

	Labeled	COREG$_{k\mathrm{NN}}$	COREG$_{\mathrm{SVR}}$	Co-SVR	L-SVR	Proposed, 3	Proposed, 5
P1	0.3461	0.3360	0.3443	0.3499	0.4807	0.3353	**0.3352**
P2	0.3543	0.3708	0.3510	0.3708	0.4372	0.3412	**0.3401**
P3	0.3547	0.3187	0.3410	0.3557	0.4629	0.3072	**0.3017**
P4	0.4056	0.3462	0.3582	0.4811	0.4911	**0.3302**	0.3302
P5	0.3192	**0.2810**	0.3191	0.3049	0.6410	0.2865	0.2841
P6	0.3167	0.3268	0.3107	0.3169	0.3866	**0.3066**	0.3080
P7	0.4021	0.3418	0.3736	0.4100	0.3928	**0.3293**	0.3410
P8	0.3319	0.3174	0.3318	0.3356	0.4036	**0.3089**	0.3089
Avg.	0.3538	0.3298	0.3412	0.3531	0.4629	**0.3181**	0.3186

Table 9
Training time of the experiment results for EQ1.

	COREG$_{k\mathrm{NN}}$	COREG$_{\mathrm{SVR}}$	Co-SVR	L-SVR	Proposed, 3	Proposed, 5
P1	17.78	17.82	0.09	28.35	5.13	7.16
P2	18.78	18.61	0.09	15.04	4.08	5.07
P3	14.25	14.28	1.96	7.47	2.07	2.59
P4	20.28	19.93	50.65	33.61	6.18	8.60
P5	18.05	18.05	1.78	153.87	8.45	11.93
P6	21.58	21.57	80.84	516.58	22.73	34.32
P7	13.64	13.63	39.78	3.11	1.64	2.21
P8	14.17	14.15	0.06	8.02	2.16	2.75
Avg.	17.32	17.25	21.91	95.76	6.55	9.41

Table 10
RMSE of the experiment results for EQ2. (The boldface numbers mean the best method for each dataset (P1 ∼ P8).)

	Labeled	COREG$_{k\mathrm{NN}}$	COREG$_{\mathrm{SVR}}$	Co-SVR	L-SVR	Proposed, 3	Proposed, 5
P1	0.3602	0.3493	0.3498	0.3544	0.4545	**0.3353**	0.3356
P2	0.3502	0.3300	0.3380	0.3346	0.4382	**0.3168**	0.3173
P3	0.3497	0.3008	0.3073	0.3744	0.3420	0.2974	**0.2941**
P4	0.2709	0.2798	0.2747	0.2763	0.4754	**0.2667**	0.2685
P5	0.3136	**0.2990**	0.3110	0.3207	0.5525	0.2991	0.2992
P6	0.1727	0.1870	**0.1724**	0.1763	0.4336	0.1753	0.1765
P7	0.4772	0.4605	**0.4429**	0.4793	0.4922	0.4461	0.4445
P8	0.3719	0.3503	0.3719	0.3768	0.4601	**0.3419**	0.3423
Avg.	0.3331	0.3196	0.3210	0.3366	0.4560	0.3098	**0.3098**

Table 11
Training time of the experiment results for EQ2.

	COREG$_{k\mathrm{NN}}$	COREG$_{\mathrm{SVR}}$	Co-SVR	L-SVR	Proposed, 3	Proposed, 5
P1	18.15	18.13	6.19	21.97	5.64	7.76
P2	14.41	14.46	3.44	11.16	3.32	4.60
P3	13.60	13.59	33.88	7.32	2.16	2.78
P4	14.56	14.57	0.05	21.51	2.94	3.47
P5	20.18	20.24	0.29	117.96	10.35	15.70
P6	3.99	4.41	67.97	406.30	13.00	16.15
P7	13.52	13.50	40.47	2.60	1.36	1.85
P8	14.24	14.17	0.06	6.39	1.97	2.46
Avg.	14.08	14.13	19.04	74.39	5.09	6.85

of COREG$_{k\mathrm{NN}}$, COREG$_{\mathrm{SVR}}$, Co-SVR, and L-SVR are plotted as upper triangles, lower triangles, x-marks, and cross-marks, respectively. The experiment results of the proposed method with different t are plotted as squares and diamonds. The proposed method shows great performance in terms of both RMSE and training time. The RMSE of the proposed method shows the best results. Meanwhile, the proposed method results in the shortest training time on average. The performance of each method can be varied by training period; however, the proposed method shows the most stable results.

Tables 8 and 9 depict the detailed experiment results for EQ1 in terms of RMSE and training time, respectively. The proposed method is superior for seven out of eight datasets. However, “Proposed, 3” is the best on average for RMSE and “Proposed, 5” is the second. Table 9 lists the training time of each method. The proposed method is faster than the other benchmark methods. Both “Proposed, 3” and “Proposed, 5” demonstrate better efficiency than the others. The reason for the training time of Co-SVR being sensitive is that Co-SVR uses SVR as the base learner. Because SVR is a function-based method, an unlabeled data point with a single estimated label can be located in a margin area without affecting SVR performance. Hence, only few newly added unlabeled data could lead to an early convergence of the algorithm. In addition, given that L-SVR should construct kernel and Laplacian matrices for all data points, including the unlabeled points, the training time for L-SVR increases relatively when the number of unlabeled data is large.

Tables 10 and 11 present the detailed experiment results for EQ2 in terms of RMSE and training time, respectively. The proposed method outperforms the others on five datasets. However, "Proposed, 5" is the best on average for RMSE and "Proposed, 3" is the second. COREG$_{SVR}$ provides inferior results compared with COREG$_{kNN}$ in terms of RMSE average. The proposed method indicates superior stability compared with the others. Table 11 presents the training time of the experiment results for EQ2. "Proposed, 3" produces the best training time; L-SVR is the worst. Based on Tables 8–11, the proposed method provides the best RMSE average for the various datasets. The training time of the proposed method, varying parameter t, is shorter than the training time of COREG$_{kNN}$ and Co-SVR. Consequently, the proposed method produces the most efficient experiment results for various datasets in terms of RMSE and training time.

6. Conclusion and discussion

In this paper, a new method for SS-SVR was proposed. In SSL, a large number of unlabeled data are used for training. Because the labels of the regression data are continuous variables, SSL for regression results in additional uncertainty for estimating the labels of unlabeled data compared with SSL for classification. The proposed method considers this labeling uncertainty. The label distribution of the unlabeled data was estimated using PLR. To improve stability with noisy data, 2-PLR was employed. PLR$_{local}$ captured the local topology of the unlabeled data. PLR$_{global}$ captured the global distribution of the underlying function. The final output was constructed by conjugating both PLR outputs. Then, the training data were generated with oversampling from the unlabeled data and their estimated label distributions. The unlabeled data with high uncertainty had a high probability of being sampled to generate additional information. Conversely, to avoid redundancy, the unlabeled data with low uncertainty had a low probability of being generated. The sampling rate varied by p_u, which was calculated by the scaling of σ_u^2. Because the proposed method generated more training data than the original training data, a data selection method, EMPS, was employed to reduce training complexity. As mentioned in Section 1, one of the main goals of the proposed method is to construct an efficient SSL for regression model with large dataset size. Oversampling was employed to obtain sufficient information of the underlying function without any iteration, and EMPS was employed to select a reduced subset that had only useful information among the data generated from oversampling.

Experiments were conducted on 30 datasets. COREG$_{kNN}$, COREG$_{SVR}$, Co-SVR, and L-SVR were employed as the performance evaluation benchmarks for the proposed method. The ratio of the labeled data varied at 20%, 10%, 5%, and 1% of the original training dataset. The proposed method outperformed all the SVR retraining methods: COREG$_{SVR}$, Co-SVR, and L-SVR. RMSE of the proposed method was approximately 3% points higher than that of COREG$_{kNN}$. The training time of the proposed method was only 20–25% that of the training time of COREG$_{kNN}$. The accuracy was stable with various values for parameter t. The proposed method also demonstrated excellent performance for the experiments conducted on a real-world semiconductor manufacturing dataset.

In terms of discussion, we provide an analysis of the conditions when the proposed method is effective. First, according to the experiment results on the benchmark datasets and VM applications, it seems that there is no tendency for the proposed method's performance in terms of size and dataset characteristics: synthetic, time series, non-time series, linearity, noise levels, and task difficulty. On the other hand, the performance of the proposed method was the best when the labeled ratio was 10% and 5%. We can conclude that the number of labeled data should be sufficient for estimating the label distribution of the unlabeled data, which was at least approximately 5% of all training datasets in the experiments. For the algorithm, the proposed method employs two PLR models to estimate the label distribution. As analyzed in Section 4.3, the PLR parameters affect the final SSL regression results. The proposed method performs well when the likelihood estimation, PLR$_{local}$, is sufficiently accurate. On the other hand, PLR$_{global}$ does not need to be accurate, but it has to learn from abundant neighbors with a large k_{global}. We can conclude that accuracy is important for estimating the likelihood distribution, whereas smoothness is important for estimating the prior distribution of the underlying function. Moreover, because the proposed method employs a probability distribution-based batch approach, the labeled data should be randomly sampled from the underlying function. In that case, estimation of the label distribution and data generation from the label distribution performs well. However, if the given labeled data were a biased sample set, performance could degrade. Finally, the proposed method can train SSL models efficiently by employing a batch approach instead of the iterative wrapper approach, which makes the proposed method recommended for training a large dataset. Based on the analysis above, we suggest the proposed method for applications in which the learning efficiency is important, and the labeled data are sampled almost randomly from the underlying function.

There are some limitations and future works for the proposed methods. First, some unlabeled data must be removed when training the final SVR model. For SS-SVR, uncertainty was used to determine the data generation rate. However, the estimated labels for some unlabeled data could be excessively uncertain to be used for training. These data should be rejected for the training of the final regression model. Co-training based methods tend to reject unlabeled data that does not upgrade the model accuracy. An unlabeled data point rejection step should be considered in order to avoid training from the unlabeled data with arbitrary target values. Moreover, this approach can be applied to other regression methods. A future research direction could be the generalization of this work for other regression model-based SSL. Moreover, when the number of labeled data is not sufficient, which is a common problem in SSL learning, we need to pay more attention to the selection of the hyper-parameter. In this paper, we used ten-fold cross-validation; however, this was more closer to the leave-one-out method when the labeled ratio was low. We intend to expand our research to develop a method for selecting suitable and robust parameters for the proposed method. Finally, the proposed method is designed for SVR only. However, the concept of considering label uncertainty and non-iterative oversampling can be applied to classification problems. We intend to expand the proposed method to SVM classifiers.

Acknowledgment

This work was supported by Basic Science Research Program through the National Research Foundation of Korea, South Korea (NRF) funded by the Ministry of Science, ICT, & Future Planning (NRF-2014R1A1A1004648).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.eswa.2015.12.027.

References

Adankon, M. M., & Cheriet, M. (2011). Help-training for semi-supervised support vector machines. *Pattern Recognition, 44*(9), 2220–2230.
Balcan, M.-F., Blum, A., & Yang, K. (2004). Co-training and expansion: Towards bridging theory and practice. *Advances in neural information processing systems*, 89–96.
Belkin, M., Mateeva, I., & Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *Learning theory: 17th annual conference on learning theory (COLT-04), Banff, Canada* (pp. 624–638).

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research, 7*, 2399–2434.
Bennett, K. P., & Demiriz, A. (1999). Semi-supervised support vector machine. *Advances in Neural Information Processing Systems, 11*, 368–374.
Besnard, J., & Toprac, A. (2006). Wafer-to-wafer virtual metrology applied to run-to-run control. In *Proceedings of the 3rd ISMI symposium on manufacturing effectiveness, Austin, TX, USA*.
Bishop, C. M. (2006). *Pattern recognition and machine learning*. Cambridge, UK: Springer.
Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the workshop on computational learning theory, New York, NY, USA* (pp. 92–100).
Brefeld, U., Gärtner, T., Scheffer, T., & Wrobel, S. (2006). Efficient co-regularised least squares regression. In *Proceedings of the 23rd international conference on machine learning, Pittsburgh, PA, USA*.
Brefeld, U., & Scheffer, T. (2004). Co-EM support vector learning. In *Proceedings of the 21st international conference on machine learning, Banff, Alberta, Canada*.
Chakraborty, S. (2011). Bayesian semi-supervised learning with support vector machine. *Statistical Methodology, 8*, 68–82.
Chang, Y., Kang, Y., Hsu, C., Chang, C., & Chan, T. (2006). Virtual metrology technique for semiconductor manufacturing. In *Proceedings of the 2006 international joint conference on neural networks, Vancouver, Canada* (pp. 5289–5293).
Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Cambridge, UK: MIT Press.
Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research, 9*, 203–233.
Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. In *Proceedings of the 10th international workshop on artificial intelligence and statistics, Barbados* (pp. 57–64).
Chen, K.-Y., & Wang, C.-H. (2007). Support vector regression with genetic algorithm in forecasting tourism demand. *Tourism Management, 28*(1), 215–226.
Cheng, J., & Cheng, F. (2005). Application development to virtual metrology in semiconductor industry. In *Proceedings of the 31st annual conference of IEEE industrial electronics society, Raleigh, NC, USA* (pp. 124–129).
Corduneanu, A., & Jaakkola, T. (2003). On information regularization. In *Proceedings of the 19th conference on uncertainty in artificial intelligence, Acapulco, Mexico* (pp. 151–158).
Cortes, C., & Mohri, M. (2007). On transductive regression. *Advances in Neural Information Processing Systems, 19*, 305.
Culp, M., & Michailidis, G. (2008). Graph-based semisupervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30*(1), 174–179.
Demiriz, A., Bennett, K. P., & Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. In *Proceedings of the conference on artificial neural networks in engineering (ANNIE-99), St. Louis, MO, USA* (pp. 809–814).
Fujino, A., Ueda, N., & Saito, K. (2005). A hybrid generative/discriminative approach to semi-supervised classifier design. In *Proceedings of the 20th national conference on artificial intelligence, Pittsburgh, PA, USA* (pp. 764–769).
Grandvalet, Y., & Bengio, Y. (2005). Semi-supervised learning by entropy minimization. *Advances in Neural Information Processing Systems, 17*, 529–536.
Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the 16th international conference on machine learning, Montreal, Canada* (pp. 202–209).
Kang, P., & Cho, S. (2008). Locally linear reconstruction for instance-based learning. *Pattern Recognition, 41*, 3507–3518.
Kang, P., Kim, D., Lee, H.-J., Doh, S., & Cho, S. (2011). Virtual metrology for run-to-run control in semiconductor manufacturing. *Expert Systems with Applications, 38*(3), 2508–2522.
Kim, D., & Cho, S. (2012). Pattern selection for support vector regression based response modeling. *Expert Systems with Applications, 39*(10), 8975–8985.
Kim, D., Kang, P., kyung Lee, S., Kang, S., Doh, S., & Cho, S. (2015). Improvement of virtual metrology performance by removing metrology noises in a training dataset. *Pattern Analysis and Applications, 18*(1), 173–189.
Kim, D., Lee, H.-J., & Cho, S. (2008). Response modeling with support vector regression. *Expert Systems with Applications, 34*(2), 1102–1108.
Krishnapuram, B., Williams, D., Xue, Y., Hartemink, A., Carin, L., & Fiqueiredo, M. A. (2005). On semi-supervised classification. *Advances in Neural Information Processing Systems, 17*, 721–728.
Lawrence, N. D., & Jordan, M. I. (2005). Semi-supervised learning via Gaussian processes. *Advances in Neural Information Processing Systems, 17*, 753–760.
Lee, S.-K., Kang, P., & Cho, S. (2014). Probabilistic local reconstruction in k-NN regression and its application to virtual metrology in semiconductor manufacturing. *Neurocomputing, 131*, 427–439.
Li, Y., Guan, C., Li, H., & Chin, Z. (2008). A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. *Pattern Recognition Letters, 29*(9), 1285–1294.
Li, Y.-F., Kwok, J. T., & Zhou, Z.-H. (2010). Cost-sensitive semi-supervised support vector machine. In *Proceedings of the 24th AAAI conference on artificial intelligence (AAAI-10), Atlanta, GA, USA* (pp. 500–505).
Maulik, U., & Chakraborty, D. (2011). A self-trained ensemble with semisupervised SVM: An application to pixel classification of remote sensing imagery. *Pattern Recognition, 44*(3), 615–623.
Mihalcea, R. (2004). Co-training and self-training for word sense disambiguation. In *Proceedings of the conference on computational natural language learning (CONLL-04), Boston, MA, USA* (pp. 33–40).
Mitchell, T. M. (1996). *An introduction to genetic algorithms*. Cambridge, UK: MIT Press.
Mitchell, T. M. (1999). The role of unlabeled data in supervised learning. In *Proceedings of the 6th international colloquium on cognitive science, San Sebastian, Spain* (pp. 2–11).
Nie, F., Xiang, S., Liu, Y., & Zhang, C. (2010). A general graph-based semi-supervised learning with novel class discovery. *Neural Computing and Applications, 19*(4), 549–555.
Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th international conference on information and knowledge management, McLean, VA, USA* (pp. 86–93).
Nigam, K., McCallum, A., & Mitchell, T. (2006). Semi-supervised text classification using EM. In *Semi-supervised learning*. Cambridge, UK: MIT Press.
Ortiz-García, E., Salcedo-Sanz, S., Pérez-Bellido, A., Portilla-Giqueras, J., & Prieto, L. (2010). Prediction of hourly O3 concentrations using support vector regression algorithms. *Atmospheric Environment, 44*, 4481–4488.
Pai, P.-F., & Lin, C.-S. (2005). A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega, 39*(6), 497–505.
Pozdnoukhov, A., & Bengio, S. (2006). Semi-supervised kernel methods for regression estimation. In *Procedings of 2006 IEEE international conference on acoustics, speech, and sinal processing (ICASSP-06), Toulouse, France*.
Qi, Z., Tian, Y., Shi, Y., & Yu, X. (2013). Cost-sensitive support vector machine for semi-supervised learning. *Procedia Computer Science, 18*, 1684–1689.
Reddy, I. S., Shevade, S., & Murty, M. (2011). A fast quasi-newton method for semi-supervised SVM. *Pattern Recognition, 44*, 2305–2313.
Rosenberg, C., Hebert, M., & Schneideman, H. (2005). Semi-supervised self-training of object detection models. In *The 7th IEEE workshop on applications of computer vision, Breckenridge, CO, USA* (pp. 29–36).
Seeger, M. (2000). *Labeling with labeled and unlabeled data: Technical report no. EPFL-REPORT-161327*. University of Edinburgh.
Sindhwani, V., & Keerthi, S. S. (2006). Large scale semi-supervised linear SVMs. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, New York, NY, USA* (pp. 477–484).
Sindhwani, V., Niyogi, P., & Belkin, M. (2005). Beyond the point cloud: From transductive to semi-supervised learning. In *Proceedings of the 22nd international conference on machine learning, Bonn, Germany* (pp. 824–831).
Smola, A., & Schölkopf, B. (2002). A tutorial on support vector regression:. *Technical Report NeuroCOLT NC-TR-98-030*. UK: Royal Holloway College, University of London.
Thissen, U., van Brakel, R., de Weijer, A., Melssen, W., & Buydens, L. (2003). Using support vector machines for time series prediction. *Chemometrics and Intelligent Laboratory Systems, 69*(1), 35–49.
Vapnik, V. (1995). *The natural of statistical learning theory*. New York, USA: Springer.
Vapnik, V. (1998). *Statistical learning theory*. New York, USA: Wiley.
Wang, M., Hua, X.-S., Song, Y., Dai, L.-R., & Zhang, H.-J. (2006). Semi-supervised kernel regression. In *Procedings of the 6th international conference on data mining (ICDM-06), Hong Kong* (pp. 1130–1135).
Wang, X., Fu, L., & Ma, L. (2011). Semi-supervised support vector regression model for remote sensing water quality retrieving. *Chinese Geographical Science, 21*(1), 57–64.
Wang, X., Ma, L., & Wang, X. (2010). Apply semi-supervised support vector regression for remote sensing water quality retrieving. In *Procedings of 2010 IEEE international geoscience and remote sensing symposium, Honolulu, HW, USA* (pp. 2757–2760).
Xu, S., An, X., Qiao, X., Zhu, L., & Li, L. (2011). Semi-supervised least squares support vector regression machines. *Journal of Information & Computational Science, 8*(6), 885–892.
Yu, S., Krishnapuram, B., Rosales, R., & Rao, R. (2011). Bayesian co-training. *Journal of Machine Learning Research, 12*, 2649–2680.
Zhang, C., & Wang, F. (2009). Graph-based semi-supervised learning. *Artificial Life and Robotics, 14*(4), 445–448.
Zhao, M., Chow, T. W., Zhang, Z., & Li, B. (2015). Automatic image annotation via compact graph based semi-supervised learning. *Knowledge-Based Systems, 76*, 148–165.
Zhou, Y., & Goldman, S. (2004). Democratic co-learning. In *Procedings of the 16th IEEE international conference on tools with artificial intelligence, Boca Raton, Florida, USA* (pp. 594–602).
Zhou, Z.-H., & Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering, 17*(11), 1529–1541.
Zhou, Z.-H., & Li, M. (2007). Semisupervised regression with cotraining-style algorithms. *IEEE Transactions on Knowledge and Data Engineering, 19*(11), 1479–1493.
Zhu, X. (2007). *Semi-supervised learning literature survey: Technical Report 1350*. Madison, WI, USA: Department of Computer Science, University of Wisconsin at Madison.