Incremental Support Vector Learning for Ordinal Regression

Bin Gu, Member, IEEE, Victor S. Sheng, Member, IEEE, Keng Yeow Tay, Walter Romano, and Shuo Li

Abstract—Support vector ordinal regression (SVOR) is a popular method to tackle ordinal regression problems. However, until now there were no effective algorithms proposed to address incremental SVOR learning due to the complicated formulations of SVOR. Recently, an interesting accurate on-line algorithm was proposed for training ν-support vector classification (ν-SVC), which can handle a quadratic formulation with a pair of equality constraints. In this paper, we first present a modified SVOR formulation based on a sum-of-margins strategy. The formulation has multiple constraints, and each constraint includes a mixture of an equality and an inequality. Then, we extend the accurate on-line σ-SVC algorithm to the modified formulation, and propose an effective incremental SVOR algorithm. The algorithm can handle a quadratic formulation with multiple constraints, where each constraint is constituted of an equality and an inequality. More importantly, it tackles the conflicts between the equality and inequality constraints. We also provide the finite convergence analysis for the algorithm. Numerical experiments on the several benchmark and real-world data sets show that the incremental algorithm can converge to the optimal solution in a finite number of steps, and is faster than the existing batch and incremental SVOR algorithms. Meanwhile, the modified formulation has better accuracy than the existing incremental SVOR algorithm, and is as accurate as the sum-of-margins based formulation of Shashua and Levin.

Index Terms—Incremental learning, online learning, ordinal regression (OR), support vector machine (SVM).

NOMENCLATURE

To make notations easier to follow, we give a summary of the notations in the following list.

\[ \alpha, \gamma \]

The ith element of the vector \( \alpha \) and \( \gamma \).

\[ J, J\_<, J\_\geq \]

The complement of the set \( J \), the contracted set of \( J \) by deleting \( j_c \), and the enlarged set of \( J \) by adding \( j_c \).

\[ d_j, E_j \]

The subvector of \( d' \) by extracting the elements indexed by \( J_{-} \), and a submatrix of \( E \) by extracting the columns indexed by \( J_{-} \).

\[ \beta_{d_j}^c \]

If \( j \in J_{-} \), \( d_j' \) and \( \beta_{d_j'}^c \) stands for \( d_j' \) and \( \beta_{d_j'}^c \), respectively. Otherwise, they will be ignored.

\[ Q_{SS}\]

The submatrix of \( Q \) with the rows and columns indexed by \( S_{\_} \).

\[ \hat{Q}_{\_\_}\]

The submatrix of \( Q \) after deleting the rows and columns corresponding to \( \Delta d_j' \) indexed by \( J_{-} \) and \( \Delta a_j \) indexed by \( M \).

\[ M^T \]

The transpose of the matrix \( M \).

\[ 0, O \]

A zero matrix with proper dimensions, and the \((r - 1) \times (r - 1)\) matrix with all zeroes except that \( O_{j, j} = 1 \).

\[ u_{j_c}, v_{j_c} \]

A \((r - 1)\)-dimensional column vector with all zeroes except that the \( j_c \)-th position is equal to \( 1 \) and \( -1 \), respectively.

\[ e_{S\_j}, u S\_j, v S\_j \]

A \( |S_{\_}| \)-dimensional column vector with all zeroes except that the positions corresponding to the samples \( (x_i, y_i) \) of \( S_{\_}j \) are equal to \(-1 \) and \( 1 \), respectively.

I. INTRODUCTION

In conventional machine learning and data mining research, predictive learning has become a standard inductive learning, where different subproblem formulations have been identified, for example, classification, metric regression, ordinal regression (OR), and so on. In OR problems, training samples are marked by a set of ranks, which exhibit an ordering among different categories. In contrast to metric regression problems [1], the ranks for OR are of finite types and the metric distances between the ranks are not defined; in comparison with classification problems, these ranks are also different from the labels of multiple classes due to ordering information [8]. Therefore, OR is a special case in predictive learning.
In practical OR tasks, such as information retrieval [15], collaborative filtering [6], flight delays forecasting [22], and so on, training data is usually provided one example at a time. This is a so-called online scenario. We use flight delays forecasting as an example. The given flight delay data streams are nonstationary, meaning that data distributions vary over time, and batch algorithms will generally fail if such ambiguous information is present and is erroneously integrated by the batch algorithm. Incremental learning algorithms are more capable in this case, because they allow the incorporation of additional training data without retraining from scratch [18].

Ever since Vapnik’s [26] influential work in statistical learning theory, support vector machines (SVMs) [26] have gained profound interest because of good generalization performance [2], [20]. There are also several support vector OR (SVOR) formulations proposed to tackle OR problems. For example, Herbrich et al. [15] gave a SVM formulation based on a loss function between pairs of ranks (PSVM). However, the problem size of PSVM is a quadratic function of the training data size. To address this problem, Shashua and Levin [6] proposed two SVM formulations by finding multiple parallel discrimination hyperplanes. One is a fixed-margin-based formulation, and the other is a sum-of-margins-based formulation (SMF). Chu and Keerthi [8] further improved the fixed-margin-based SVOR formulation by explicitly and implicitly keeping ordinal inequalities on the thresholds, in which the explicit constraints-based SVOR was called EXC. Cardoso and Pinto da Costa [23] proposed a data replication method and mapped it into SVM, which also implicitly used the fixed-margin strategy. The problem sizes of these SVOR formulations are all linear in the training data size. In addition, more recently, Seah et al. [17] presented a transductive SVM learning paradigm for OR, by taking advantage of the abundance of unlabeled patterns.

Although there exist several perceptron-like algorithms proposed for incremental OR learning (see [3], [9], [10]), very little work has been done on incremental learning for SVOR. Previous works mostly focus on incremental learning for standard SVM, one-class SVM, support vector regression (SVR), and so on. For example, Cauwenberghs and Poggio [7] proposed an exact incremental learning approach (the C&P algorithm) for SVM in 2001. Later, Martin [11] extended it to SVR and proposed an accurate incremental SVR algorithm. Laskov et al. [18] implemented an accurate incremental one-class SVM algorithm. Karasuyama and Takeuchi [25] gave an extended algorithm that can handle multiple data samples simultaneously. Recently, Gu et al. [28] extended the C&P algorithm to $\nu$-support vector classification ($\nu$-SVC) and proposed an effective accurate on-line $\nu$-SVC algorithm (AONSVM), which can handle the conflict between a pair of equality constraints during the process of incremental learning. Gu and Sheng [29] proved the feasibility and finite convergence of AONSVM under two assumptions (i.e., Assumptions 1 and 2 as mentioned in [29]).

To the best of our knowledge, the PSVM-based incremental algorithm (IPSVM) [12] is the only work on incremental SVOR learning. As mentioned previously, this approach is limited by the size of the problem, which is quadratic in the training data size. Therefore, it is highly desirable to design an effective incremental learning algorithm for the SVOR formulations, whose problem size is linear in the training data size. In this paper, we focus on the SMF of Shashua and Levin [6]. We first present a modified SMF (MSMF), which has multiple constraints of the mixture of an equality and an inequality. Then, we extend AONSVM to MSMF and propose an effective incremental SVOR algorithm (ISVOR). The incremental algorithm includes two steps, i.e., relaxed adiabatic incremental adjustments (RAIAs), and strict restoration adjustments (RSA). Based on the two steps, the incremental algorithm can handle inequality constraints, and can tackle the conflicts between the equality and inequality constraints. We also provide its finite convergence analysis. Numerical experiments show that ISVOR can converge to the optimal solution in a finite number of steps, and is faster than the existing batch and incremental SVOR algorithms. Meanwhile, the modified formulation has better accuracy than the existing incremental SVOR algorithm, and is as accurate as the SMF of Shashua and Levin [6].

The main contributions of this paper are summarized as follows.

1) We propose an effective ISVOR, whose problem size is linear in the training data size. We also prove the finite convergence of ISVOR. Numerical experiments show that ISVOR is faster than the existing batch and incremental SVOR algorithms.

2) The existing incremental SVM algorithms can handle a quadratic formulation with a pair of equality constraints or an equality constraint for a binary classification problem. The ISVOR can handle a quadratic formulation with multiple constraints of the mixture of an equality and an inequality for multiple binary classification problems. The ISVOR can be viewed as a generalization of the existing incremental SVM algorithms.

The rest of this paper is organized as follows. Section II gives a modified SVOR formulation, (i.e., MSMF), and its Karush–Kuhn–Tucker (KKT) conditions. The incremental SVOR algorithm is presented in Section III. The experimental setup, results and discussion are presented in Sections IV and V. The last section gives some concluding remarks.

II. MODIFIED SVOR FORMULATION

In this section, we first review SMF, then present MSMF and its dual problem. Finally, we present the KKT conditions for the solution of the dual problem.

A. Review of SMF

Without loss of generality, we consider an OR problem with $r$ ordered categories and denote these categories as consecutive integers $Y = \{1, 2, \ldots, r\}$ to keep the known ordering information. The number of training samples in the $j$th category ($j \in Y$), is denoted as $n_j$, and the $i$th training sample is denoted as $x_i^j$ ($x_i^j \in X$, where $X$ is the input space with $X \subset \mathbb{R}^d$).
Fig. 1. (a) OR based on the sum-of-margins strategy. \((w, b_j)\) and \((w, b_2)\) are the parallel discrimination hyperplanes obtained from maximizing \(2d_1 + 2d_2\) when \((w, w) = 1\). Support vectors lie on the boundaries between the neighboring categories. (b) Two cases of incremental SVOR learning. If the added sample (e.g., \(x_3\)) is sandwiched between hyperplanes \((w, b_j - d_j)\) and \((w, b_j + d_j)\), adjustments will be needed; otherwise, adjustments will be unnecessary for the added sample, such as \(x_1\) or \(x_2\).

To learn a mapping function \(r(\cdot) : X \rightarrow Y\), Shashua and Levin [6] considered \(r - 1\) parallel discrimination hyperplanes, i.e., \((w, x) - b_j\) with \(b_1 \leq \ldots \leq b_{r-1}\), where \(b_j\) is the threshold of the \(j\)th discrimination hyperplane. Supposing \(b_j = \infty\), the decision mapping function \(r(\cdot)\) can be denoted as

\[
  r(x) = \min_{j \in \{1, \ldots, r\}} \langle w, x \rangle - b_j < 0
\]

Let \(d_j \geq 0\) be the shortest distance from the \(j\)th discrimination hyperplane to the closest sample in the \(j\)th or \((j + 1)\)th category, which is the margin of the \(j\)th discrimination hyperplane (Fig. 1). Based on the sum-of-margins strategy [Fig. 1(a)], Shashua and Levin [6] tried to maximize the sum of all margins \(\sum_{j=1}^{r-1} 2d_j\) with \((w, w) = 1\), and considered all the sandwiched constraints \(b_j + d_j \leq b_{j+1} - d_{j+1}\), which derive the following primal problem (i.e., SMF): \(^1\)

\[
  \min_{w, b_j, d_j, \epsilon_j} \left\{ - \sum_{j=1}^{r-1} 2d_j + C \sum_{j=1}^{r-1} \left( \sum_{i=1}^{n_j} \epsilon_j + \sum_{i=1}^{n_j+1} \epsilon^{s+j+1}_j \right) \right\}
\]

s.t. \(\langle w, \phi(x_j^i) \rangle \leq b_j - d_j + \epsilon_j^i\), \(\epsilon_j^i \geq 0\), \(i = 1, \ldots, n_j\), \(\langle w, \phi(x^{s+j+1}_j) \rangle \geq b_j + d_j - \epsilon^{s+j+1}_j\), \(\epsilon^{s+j+1}_j \geq 0\), \(i = 1, \ldots, n_j+1\), \(\langle w, w \rangle \leq 1\), \(b_j + d_j \leq b_{j+1} - d_{j+1}\), \(d_j \geq 0\) \(\quad (2)\)

where \(j = 1, \ldots, r - 1\), training samples \(x^i_j\) are mapped into a high-dimensional reproducing kernel Hilbert space (RKHS) [19] through the transformation function \(\phi\), and we have the kernel function \(K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle\) with \(\langle \cdot, \cdot \rangle\) denoting inner product in RKHS. Furthermore, \(\epsilon_j^i\) \((\epsilon^{s+j+1}_j)\) is a non-negative slack variable measuring the degree of misclassification of the data \(x^i_j\) \((x^{s+j+1}_j)\). The parameter \(C\) controls the tradeoff between the errors in the training samples and sum-of-margins maximization.

**B. MSMF and Its Dual Problem**

According to the reduction framework of OR [20], OR learning tries to learn a rank-monotonic mapping function \(f(x, j)\), such that \(f(x, 1) \geq \cdots \geq f(x, r - 1)\). A popular approach to obtain such a function \(f(x, j)\) is to use \(r - 1\) parallel discrimination hyperplanes as mentioned in Section II-A. The key of such an approach is to keep the thresholds \(b_j\) ordered. To sum up, there are two kinds of approaches to keep such ordinal thresholds, i.e., the explicit approach [6], [8], and the implicit approach [8], [10], [15], [17], [20], [23]. It should be noted that although Shashua and Levin [6] used an explicit approach, the ordinal thresholds were achieved by the sandwiched constraints as shown in (2) because \(b_j \leq b_f + d_f \leq b_{j+1} - d_{j+1} \leq b_{j+1}\). In this paper, we use the popular implicit approach to achieve the ordinal thresholds. Thus, a modified formulation of (2) is used here by discarding the constraint of \(b_j + d_j \leq b_{j+1} - d_{j+1}\). After discarding the constraint, our proposed OR formulation, (i.e., MSMF) is more favorable to design an incremental SVOR algorithm, because the primal variables \(b_j\) and \(d_j\) can be induced directly in the KKT conditions (Section II-C).

To present the dual function of the modified formulation in a compact form, we introduce some new notations.

1) Based on the reduction framework of [20], OR can be regarded as \(r - 1\) binary classification problems. Thus, we define the two-class training sample set \(S_j = \{ (x_j^i, y_j^i = -1)\}_{i=1}^{n_j} \cup \{ (x_j^{s+j+1}_i, y_j^{s+j+1}_i = +1)\}_{i=1}^{n_j+1}\), and the extended training sample set \(S = \bigcup_{j=1}^{r-1} S_j = \{(x_1, y_1), \ldots, (x_r, y_r)\}\), where \(l = 2 \times \sum_{j=1}^{r-1} n_j = n^1 - n^l\).

2) We let \(\lambda_j = [\lambda_1^j, \ldots, \lambda_r^j]\) and \(\delta_j = [\delta_1^j, \ldots, \delta_r^j]\), where \(\lambda_j^l\) and \(\delta_j^l\) are the Lagrangian multipliers corresponding to the first and third inequality constraints in (2), respectively. Thus, \(\alpha = [\lambda^1, \delta^1, \ldots, \lambda^{l-1}, \delta^{l-1}]\) is defined to be the row vector containing all the Lagrangian multipliers \(\lambda_j^l\) and \(\delta_j^l\).

3) We define the kernel matrix \(Q\) as \(Q_{ik} = y_i y_k K(x_i, x_k)\) for all \(1 \leq i, k \leq l\).

Based on the above notations, the dual problem can be formulated as follows:

\[
  \min_{\alpha} \frac{1}{2} \alpha Q \alpha^T
\]

s.t. \(\sum_{i \in S_j} y_i a_i = 0\), \(\sum_{i \in S} a_i \geq 2\), \(j = 1, \ldots, r - 1\)

\(0 \leq a_i \leq C\), \(i = 1, \ldots, l\) \(\quad (3)\)

where \(i \in S_j\) is the abbreviated form of \((x_i, y_i) \in S_j\).

Once the optimal solution \(\alpha\) is obtained, the part \((w, \phi(x))\) of the rank-monotonic mapping function \(f(x, j)\) in RKHS can be obtained as follows:

\[
  \langle w, \phi(x) \rangle = \frac{\sum_{i=1}^l y_i a_i K(x_i, x)}{\sqrt{\alpha Q a^T}}.
\]

In addition, \(b_j\) can be obtained by solving the following linear equations:

\[
  \frac{\sum_{i=1}^l y_i a_i K(x_i, x_j)}{\sqrt{\alpha Q a^T}} - b_j + d_j = 0 \quad (5)
\]

\[
  \frac{\sum_{i=1}^l y_i a_i K(x_i, x_{j+1})}{\sqrt{\alpha Q a^T}} - b_j - d_j = 0 \quad (6)
\]
Fig. 2. Partitioning a two-class training sample set \( S^j \), which is associated with the \( j \)th binary classification, into three independent sets by KKT-conditions. (a) \( S^j_0 \) with the sample set \( S^j \). Thus, the extended training sample set \( S^{j}_{\text{new}} \) includes \( \{x_i, y_i \} \), \( (x_{i_2}, y_{i_2}) \) \( \subseteq S^j \) with \( y_{i_1} = -1 \) and \( y_{i_2} = +1 \), and \( x_{i_1}, x_{i_2} \) are also support vectors with their weights \( 0 < \alpha_{i_1} < C \), \( 0 < \alpha_{i_2} < C \).

### C. KKT Conditions

According to convex optimization theory [4], the solution of the dual problem (3) can be obtained by the following min–max problem:

\[
\begin{align*}
\min_{0 \leq \alpha_i \leq C} \max_{b'_j, d'_j \geq 0} & \quad W = \frac{1}{2} \sum_{i,k=1}^{l} \alpha_i \alpha_k Q_{ik} + \sum_{j=1}^{r-1} b'_j \left( \sum_{i \in S^j} y_i \alpha_i \right) \\
& \quad + \sum_{j=1}^{r-1} d'_j \left( 2 - \sum_{i \in S^j} \alpha_i \right) \\
\end{align*}
\]

where \( b'_j \in \mathbb{R} \) and \( d'_j \in \mathbb{R}_+ \) are Lagrangian multipliers.

From the KKT theorem [5], we obtain the following KKT conditions:

\[
\begin{align*}
\sum_{i \in S^j} y_i \alpha_i &= 0 \quad \text{(8)} \\
p_j &= \sum_{i \in S^j} \alpha_i \begin{cases} 
\geq 2 & \text{for } d'_j = 0 \\
= 2 & \text{for } d'_j > 0 
\end{cases} \quad \text{(9)} \\
g_i &= \frac{\partial W}{\partial \alpha_i} = \sum_{k=1}^{l} \alpha_k Q_{ik} + y_i b'_j - d'_j \begin{cases} 
\geq 0 & \text{for } \alpha_i = 0 \\
= 0 & \text{for } 0 < \alpha_i < C \\
\leq 0 & \text{for } \alpha_i = C. 
\end{cases} \quad \text{(10)}
\end{align*}
\]

According to the value of the function \( g_i \), a two-class training sample set \( S^j \) associated with the \( j \)th binary classification is partitioned into three independent sets (Fig. 2).

1. \( S^j_0 = \{ i \in S^j : g_i = 0, 0 < \alpha_i < C \} \), the set \( S^j_0 \) includes margin support vectors strictly on the margins.
2. \( S^j_E = \{ i \in S^j : g_i \leq 0, \alpha_i = C \} \), the set \( S^j_E \) includes error support vectors exceeding the margins.
3. \( S^j_R = \{ i \in S^j : g_i \geq 0, \alpha_i = 0 \} \), the set \( S^j_R \) includes the remaining vectors ignored by the margins.

Thus, the extended training sample set is partitioned into three independent sets, i.e., \( S^j_{\text{new}} = \bigcup_{j=1}^{r-1} S^j_0, S^j_E = \bigcup_{j=1}^{r-1} S^j_E, \) and \( S^j_R = \bigcup_{j=1}^{r-1} S^j_R. \)

In addition, according to the value of \( p_j \) in (9), we can define an active set \( J \subseteq \{1, \ldots, r-1\} \) with \( p_j = 2 \) and \( d'_j > 0 \) for all \( j \in J \).

### III. INCREMENTAL SVOR LEARNING

In this section, we consider the incremental SVOR learning algorithm for the dual problem (3). When a sample \( x_{\text{new}} \) is added into the \( j \)th category, there correspondingly exists increments in \( S \) and \( S^j \) (Table I). We define the increments as \( S^j_{\text{new}} \) and \( S^{j}_{\text{new}} \), respectively. Initially, we set the weight \( \alpha_c \) of each sample \( (x_c, y_c) \) in \( S_{\text{new}} \) to zero. If this assignment satisfies the KKT conditions, adjustments are not needed. However, if this assignment violates the KKT conditions, additional adjustments become necessary [Fig. 1(b)]. The goal of the incremental SVOR algorithm is to find an effective method for updating the weights without retraining from scratch, when a sample in \( S_{\text{new}} \) violates the KKT conditions.

Compared with the formulations of standard SVM, one-class SVM, SVR, and \( \nu \)-SVC, our SVOR formulation (3) has the following challenges, which prevent us from directly using the existing incremental SVM algorithms, including the C&P algorithm and AONSVM.

1. If \( |\sum_{i \in S^j_{\text{new}}} y_i| = |S^j_S|, \) \( d'_j > 0 \), and the label of an added sample \((x_c, y_c)\) in \( S^j_{\text{new}} \) is different from those of the margin support vectors in \( S^j_S \), there exists a conflict (referred to as Conflict-1) between (8) and (9) with a small increment of \( \alpha_c \) (Table II). Conflict-1 is different to the one in \( \nu \)-SVC, because an additional condition \( d'_j > 0 \) must be considered here.

2. The SVOR formulation (3) has multiple constraints of the mixture of an equality and an inequality, which is more complicated than a pair of equality constraints in \( \nu \)-SVC, and an equality constraint in standard SVM, one-class SVM, and SVR.

To address these challenges, we propose an incremental SVOR algorithm (i.e., ISVOR, see Algorithm 1), which includes two steps, similar to AONSVM.

The first step is RAIA. Because there may exist Conflict-1 between (8) and (9) as shown in Table II, the feasible updating path leading to the eventual satisfaction of the KKT conditions will not be guaranteed. To overcome this problem, the limitation on the enlarged \( j \)th two-class training...
Algorithm 1 Incremental SVOR Algorithm

Input: \(\alpha, d', g, p, S, R, x_{new}(\alpha, d', g)\) and \(p\) satisfy the KKT conditions of \(S, s_{new}\) is the new sample added into the \(j\)th category.

Output: \(\alpha, d', g, p, S, R\).

1: Compute \(S_{new}\) according to Table I.
2: \(\text{while} \ S_{new} \neq \emptyset \ \text{do}\)
3: \(\text{Read} (x_{c}, y_{c}) \text{from} S_{new}, \text{initial its weight} \alpha_{c} \leftarrow 0 \text{and compute} g_{c}.
4: \text{Update} S_{new} \leftarrow S_{new} - \{(x_{c}, y_{c})\}, \ S_{j}^{c} \leftarrow S_{j}^{c} \cup \{(x_{c}, y_{c})\}, \ S \leftarrow S \cup \{(x_{c}, y_{c})\}.
5: \text{while} \ g_{c} < 0 \text{and} \alpha_{c} < C \text{do}
6: \text{Compute} \beta_{b}^{c}, \beta_{d}^{c}, \beta_{s}^{c}, \beta_{f}^{c}, \text{and} \ y_{c}.
7: \text{Compute the maximal increment} \Delta \alpha_{c}^{max}.
8: \text{Update} \alpha, g, b^{'}, d^{'}, p, J_{e}, \ T_{e, j}, S_{s}, S_{E} \text{and} S_{R}.
9: \text{Update the inverse matrix} \ R.
10: \text{end while}
11: \text{Compute the inverse matrix} \ R \text{based on} \ R.
12: \text{while} \ p_{\theta} < 2 \text{or} (p_{\theta} > 2 \text{&} d_{\theta} > 0) \text{do}
13: \text{Compute} \tilde{\beta}_{b}, \tilde{\beta}_{d, \theta}, \tilde{\beta}_{s, \theta}, \tilde{\beta}_{f, \theta}, \text{and} \tilde{y}_{\theta}.
14: \text{Compute the critical adjustment quantity} \Delta \zeta_{\theta}^{*}.
15: \text{Update} \alpha, g, b^{'}, d^{'}, p, J_{e}, \ T_{e, j}, S_{s}, S_{E} \text{and} S_{R}.
16: \text{Update the inverse matrix} \ R.
17: \text{end while}
18: \text{Compute the inverse matrix} \ R \text{based on} \ R.
19: \text{end while}

samples imposed by inequality (9) is removed from this step, similar to AONSVM. In addition, our basic idea is gradually increasing \(\alpha_{c}\) under the condition of rigorously keeping all the samples satisfying the KKT conditions, except that the inequality restriction (9) should be held for the weights of the enlarged \(j\)th two-class training samples (Fig. 3). This procedure is described with pseudocode in lines 5–10 of Algorithm 1, and the details are stated in Section III-A.

The second step is SRA, whose objective is to restore the inequality restriction (9) on the enlarged \(j\)th two-class training samples. Our idea is gradually adjusting \(p_{\theta}\) under the condition of rigorously keeping all samples satisfying the KKT conditions, until all the samples satisfy the KKT conditions (Fig. 4). In addition, to avoid the recurrence of the conflict (referred to as Conflict-2) between (8) and (9) if \(\sum_{i \in S_{s}^{j}} y_{i} = |S_{s}^{j}|\) during the adjustments for \(p_{\theta}\), a trick is used in this step, similar to AONSVM. This procedure is described with pseudocode in lines 11–17 of Algorithm 1, and the details are discussed in Section III-B.

Although ISVOR and AONSVM share the similar two step procedure, ISVOR has a more general framework for processing the objective function than AONSVM, from the following two aspects.

1) ISVOR can handle multiple binary classification problems simultaneously, especially the singularity of the key matrix. However, AONSVM just manages one binary classification problem, and does not need to take account of the singularity of the key matrix.

2) ISVOR can handle multiple inequality constraints in the objective function. However, AONSVM can only handle a pair of equality constraints.

If only considering two categories in OR, and transforming the inequality constraint into an equality constraint as in [28], ISVOR degenerates to AONSVM. If further discarding the inequality constraint from the above formulation, ISVOR degenerates to the C&P algorithm, similar to AONSVM [29]. Thus, ISVOR can be viewed as a generalization of the C&P algorithm and AONSVM.

A. Relaxed Adiabatic Incremental Adjustment

During the incremental adjustment for \(\alpha_{c}\), the weights of the samples in \(S_{s}\), the Lagrange multipliers \(b_{j}^{'}\) and \(d_{j}^{'}\) should also be adjusted accordingly, to keep all the samples satisfying the KKT conditions, except that the restriction (9) should be held for the weights of the enlarged \(j\)th two-class training samples. Thus, we have the following linear system:

\[
\forall j \neq j_{\theta}: \sum_{k \in S_{s}^{j}} y_{k} \Delta a_{k} = 0 \quad (11)
\]

\[
j = j_{\theta}: \sum_{k \in S_{s}^{j}} y_{k} \Delta a_{k} + y_{c_{j}} \Delta a_{c_{j}} = 0 \quad (12)
\]

\[
\forall j \in J_{-}: \Delta p_{j} = \sum_{k \in S_{s}^{j}} \Delta a_{k} = 0 \quad (13)
\]

\[
\forall i \in S_{s}^{j}: \Delta g_{i} = \sum_{k \in S_{s}^{j}} \Delta a_{k} \cdot Q_{ik} + y_{i} \Delta b_{j}^{'} - \Delta d_{j}^{'} + \Delta a_{c_{j}} \cdot Q_{ic} = 0 \quad (14)
\]

where \(j = 1, \ldots, r - 1\). We let \(\Delta b^{'} = [\Delta b_{1}^{'}, \ldots, \Delta b_{r-1}^{'}]^{T}\), \(\Delta d^{'} = [\Delta d_{1}^{'}, \ldots, \Delta d_{r-1}^{'}]^{T}\), \(E = [e_{S_{s}^{j}}, \ldots, e_{S_{s}^{j-1}}]\), and \(U = [u_{S_{s}^{j}}^{T}, \ldots, u_{S_{s}^{j-1}}^{T}]^{T}\). Thus, the linear system (11)–(14) can be further rewritten as

\[
\begin{bmatrix}
0 & 0 & U^{T} \\
0 & 0 & E_{J_{-}}^{T} \\
U & E_{J_{-}} & Q_{S_{s}s_{s}}
\end{bmatrix}
\begin{bmatrix}
\Delta b^{'} \\
\Delta d_{j}^{'} \\
\Delta a_{s}^{*}
\end{bmatrix}
= -
\begin{bmatrix}
u_{J_{-}} \\
0 \\
Q_{S_{s}s_{s}}
\end{bmatrix}
\Delta a_{c_{j}}. \quad (15)
\]

It can be concluded that \(\tilde{Q}_{(d_{j}^{'})^2}\) becomes singular in the following two cases.

SC-1: The first singular case is that \(\sum_{i \in S_{s}^{j}} y_{i} = |S_{s}^{j}|\) for some \(j \in J_{-}\), i.e., the samples of \(S_{s}^{j}\) only have one kind of labels for some \(j \in J_{-}\). For example:

a) if \(\forall i \in S_{s}^{j}, y_{i} = +1\), we have \(e_{S_{s}^{j}}^{T} - u_{S_{s}^{j}} = 0\); 

b) if \(\forall i \in S_{s}^{j}, y_{i} = -1\), we have \(e_{S_{s}^{j}}^{T} + u_{S_{s}^{j}} = 0\).

We define the index set \(J_{-} = \{j \in J_{-} : |\sum_{i \in S_{s}^{j}} y_{i} | \neq |S_{s}^{j}|\}\). Thus, if \(|J_{-} - J_{-}^{'}| \neq 0\), \(\tilde{Q}_{(d_{j}^{'})^2}\) becomes singular.

SC-2: The second singular case is that \(|M_{j}^{'}| > 1\) for some \(j \in \{2, \ldots, r - 1\}\). The \(M_{j}^{'}\) is defined as \(M_{j}^{'} = \left\{(x_{j}^{1}, -1) \in S_{s}^{j} : (x_{j}^{1}, +1) \in S_{s}^{j-1}\right\}\).
Supposing that there exist four samples indexed by \(i_1, i_2, k_1, \) and \(k_2, \) respectively, where \(\{i_1, k_1\} \subset S_{S}^{-1}, \{i_2, k_2\} \subset S_{S}^*, x_{i_1} = x_{i_2}, \) and \(x_{k_1} = x_{k_2}.\) According to (14), we have \(\Delta g_{i_1} + \Delta g_{i_2} = \Delta g_{k_1} + \Delta g_{k_2},\) which means \(\overline{Q}_{i_1} + \overline{Q}_{i_2} = \overline{Q}_{k_1} + \overline{Q}_{k_2}\). In this case, it is easy to verify that \(\overline{Q}_{\{a_{\alpha}^c\}^2}\) is a singular matrix. When \(M'_{j} \neq \emptyset,\) we define \(M_{j}\) as the contracted set which is obtained by deleting any one sample from \(M'_{j}.\) Obviously, \(M_{j}\) is also an empty set when \(M'_{j} = \emptyset.\) Further, we let \(M = M_2 \cup \ldots \cup M_{r-1},\) and \(S_{S} = S_{S} - M.\) Thus, if \(M \neq \emptyset,\) \(\overline{Q}_{\{a_{\alpha}^c\}^2}\) is singular.

Now, we let \(\overline{Q}_{\{a_{\alpha}^c\}^2, M^2}\) denote the contracted matrix of \(\overline{Q}_{\{a_{\alpha}^c\}^2}.\) Similar to the analysis in [29, Th. 2], we can prove that \(\overline{Q}_{\{a_{\alpha}^c\}^2, M^2}\) has the inverse matrix \(R.\) Thus, the linear relationship between \(\Delta h', \Delta d'_j, \Delta a_{S_{S}^*},\) and \(\Delta a_{c}\) can be easily solved as follows:

\[
\begin{bmatrix}
\Delta h' \\
\Delta d'_j \\
\Delta a_{S_{S}^*}
\end{bmatrix} = -R \begin{bmatrix}
u_{i_{2}} \\
0 \\
Q_{S_{S}^*}
\end{bmatrix} \Delta a_{c} = \begin{bmatrix}
\beta_{h_{2}}' \\
\beta_{d_{j}}' \\
\beta_{a_{S_{S}^*}}'
\end{bmatrix} \Delta a_{c}. \tag{16}
\]

Substituting (16) into (13), we get the linear relationship between \(\Delta p_{j}\) and \(\Delta a_{c}\) as follows:

\[
\Delta p_{j} = \sum_{k \in S_{S}^*} \beta_{c_{k}}' \Delta a_{c} = \rho_{j}^c \Delta a_{c}. \tag{17}
\]

Obviously, \(\forall j \in J_-,\) we have \(\rho_{j}^c = 0.\)

Finally, substituting (16) into (14), we can get the linear relationship between \(\Delta g_{i} (\forall i \in S_{S}^*)\) and \(\Delta a_{c}\) as follows:

\[
\Delta g_{i} = \beta_{c_{k}}' Q_{S_{S}^*} + y_{i} \beta_{d_{j}}' - \beta_{d_{j}}' \Delta a_{c} \tag{18}
\]

where \(j = 1, \ldots, r - 1.\) Obviously, \(\forall i \in S_{S},\) we have \(\gamma_{i} = 0.\)

1) Some Details of RAIA: Once the linear relationships between \(\Delta h', \Delta d'_j, \Delta a_{S_{S}^*}, \Delta p, \Delta g,\) and \(\Delta a_{c}\) are available, the maximal increment \(\Delta a_{c}^\text{max}\) can be computed for each incremental adjustment (Fig. 3), such that a certain sample migrates among the sets \(S_{S}, S_{R}, \) and \(S_{E},\) or a certain index migrates between \(J_+\) and \(J'_+.\) There are four cases considered to account for such structural changes.

1. Some \(a_{i}\) in \(S_{S}\) reaches a bound. Compute the sets: \(I_{S_{S}^*} = \{i \in S_{S} : \beta_{c_{i}}' > 0\}, I_{S_{S}^-} = \{i \in S_{S} : \beta_{c_{i}}' < 0\}\). Thus, the maximum possible weight updates are

\[
\Delta a_{i}^\text{max} = \left\{ C - a_{i}, \text{ if } i \in I_{S_{S}^*} \right\} - a_{i}, \text{if } i \in I_{S_{S}^-}\n\]

and the maximal possible \(\Delta a_{c}^S\) before a certain sample in \(S_{S}\) moves to \(S_{R}\) or \(S_{E}\) is \(\Delta a_{c}^S = \min_{i \in I_{S_{S}^*} \cup I_{S_{S}^-}} (\Delta a_{i}^\text{max} / \beta_{c_{i}'})\).

2. A certain \(d_{j}\) or \(p_{j}\) reaches zero. Compute the sets: \(I_{J_{-}} = \{j \in J_{-} : \beta_{d_{j}}' < 0\}, I_{J_{+}} = \{j \in J_{+} : \beta_{p_{j}}' < 0\}\).

Thus, the maximal possible \(\Delta a_{c}^{J_{-}, J_{+}}\) before a certain index in \(J_{-} (J_{+})\) migrates to \(J_{+} (J_{-})\) is \(\Delta a_{c}^{J_{-}, J_{+}} = \min_{j \in I_{J_{-}}} (\Delta a_{c}^{J_{-}, J_{+}} (-d_{j}' / \beta_{d_{j}}'), \min_{j \in I_{J_{+}}} (\Delta a_{c}^{J_{-}, J_{+}} (-p_{j}' / \beta_{p_{j}}'))).

3. A certain \(g_{i}\) corresponding to a sample in \(S_{R}\) or \(S_{E}\) reaches zero. Compute the sets: \(I_{S_{R}^*} = \{i \in S_{R} : \gamma_{i} > 0\}, I_{S_{E}^*} = \{i \in S_{E} : \gamma_{i} < 0\}\). Thus, the maximal possible \(\Delta a_{S_{R}^*, S_{E}^*}\) before a certain sample in \(S_{R}\) or \(S_{E}\) migrates to \(S_{S}\) is \(\Delta a_{S_{R}^*, S_{E}^*} = \min_{i \in I_{S_{R}^*} \cup I_{S_{E}^*}} (-g_{i} / \gamma_{i}).\)

4. \(a_{c}\) reaches the upper bound or \(g_{c}\) reaches zero. The maximal possible \(\Delta a_{c}\), before the new candidate sample \((x_{c}, y_{c})\) satisfies the restriction (7) of the KKT conditions, is \(\Delta a_{c}^{} = \min \{C - a_{c}, -g_{c} / \gamma_{c}\}.

Finally, the smallest of the four values

\[
\Delta a_{c}^\text{max} = \min \left\{ \Delta a_{S_{S}^*}, \Delta a_{c}^{J_{-}, J_{+}}, \Delta a_{S_{R}^*, S_{E}^*}, \Delta a_{c}^{} \right\} \tag{19}
\]

constitutes the maximal increment of \(\Delta a_{c}\).

Based on the maximal increment \(\Delta a_{c}^\text{max}\) we can update \(a, g, h', d', p\) according to (16)–(18), and \(J_{+}, J_{-}, S_{S}, S_{E}\) and \(S_{R}\) according to (19).

Once the components of the set \(S_{S}^*\) or \(J_{+}\) are changed, i.e., a sample is either added to or removed from the set \(S_{S}^*\) or an index is added into or removed from the set \(J_{+}\), the changes of the inverse matrix can be found in Lemma 5 and 6 of.[29]. In addition, after SRA, the inverse matrix \(\hat{R}\) for the next round of RAIA can also be computed from \(\hat{R}\) using the same contracted rule.

2) Finite Convergence of RAIA: Obviously, RAIA is an iterative procedure. Thus, we are concerned about its finite convergence, which is the foundation of the usefulness of RAIA. Specifically, the finite convergence of RAIA means that a new candidate sample \((x_{c}, y_{c})\) will satisfy the KKT conditions in a finite number of steps, except that the restriction (9) does not need to hold for all the weights of the enlarged \(j_{i}\)th two-class training samples. In this section, we will prove it.

To prove the finite convergence of RAIA, we first prove that the objective function \(W\) is strictly monotonically decreasing during RAIA (Theorem 1).

**Theorem 1:** During RAIA, the objective function \(W\) is strictly monotonically decreasing.

**Proof:** During RAIA, suppose that the previous adjustment is indexed by \(k\), the immediate next is indexed by \(k + 1,\)
and let \( \beta_0^c = 0, \beta_S^c = 0, \beta^c = 1 \), then we have

\[
W_{k+1}^{[q]} = \frac{1}{2} \sum_{i, j \in S} \left( a_{ij}^{[q]} + \beta_{ij}^{[q]} \Delta a_{ij}^{[q]} \right) \left( a_{ij}^{[q]} + \beta_{ij}^{[q]} \Delta a_{ij}^{[q]} \right) Q_{i j} + \sum_{j=1}^{r-1} \left( b_j^{[q]} + \beta_{bj}^{[q]} \Delta a_{bj}^{[q]} \right) y_i \left( a_i^{[q]} + \beta_i^{[q]} \Delta a_i^{[q]} \right) + \sum_{j=1}^{r} \left( d_j^{[q]} + \beta_{d}^{[q]} \Delta a_{d}^{[q]} \right) \left( \sum_i a_i^{[q]} + \beta_i^{[q]} \Delta a_i^{[q]} - 2 \right) = W_k^{[q]} + \sum_{i \in S} \gamma_i^{[q]} \Delta a_i^{[q]} + \frac{1}{2} \sum_{i \in S} \gamma_i^{[q]} \Delta a_i^{[q]} - 2 = W_k^{[q]} + \frac{1}{2} \sum_{i \in S} \gamma_i^{[q]} \Delta a_i^{[q]}
\]

In other words, \( W_{k+1}^{[q]} - W_k^{[q]} = \left( g_k^{[q]} + 1/2 r_k^{[q]} \Delta a_k^{[q]} \right) \Delta a_k^{[q]} \). Similar to [29, Corollary 8], we can prove that the maximal increment \( \Delta a_k^{\text{max}} > 0 \) for each RAIA. In addition, it is easy to verify that \( g_k^{[q]} + 1/2 r_k^{[q]} \Delta a_k^{[q]} < 0 \), so we have \( W_{k+1}^{[q]} - W_k^{[q]} < 0 \). This completes the proof.

Let \( (W_1^{[q]}, W_2^{[q]}, W_3^{[q]}, \ldots) \) be the sequence generated during RAIA. Based on Theorem 1, we know that \((W_1^{[q]}, W_2^{[q]}, W_3^{[q]}, \ldots)\) is a monotonically decreasing sequence. To further prove the finite convergence of RAIA, we can show that the sequence is finite and converges to the KKT conditions, except that the restriction (9) does not need to hold for all the weights of the enlarged \( j \)-th two-class training samples, which is similar to [29, Th. 14].

### B. SRA

After RAIA, the KKT conditions are satisfied by all the samples, except that the inequality restriction (9) is satisfied by the enlarged \( j \)-th two-class training samples. In the SRA step, we gradually adjust \( p_{j^*} \) to restore the inequality restriction (9), so that the KKT conditions are satisfied by all the samples.

For each adjustment of \( p_{j^*} \), the weights of the samples in \( S_j \), the Lagrange multipliers \( b^* \) and \( d^* \) should also be adjusted accordingly, to keep all the samples satisfying the KKT conditions. Thus, we have the following linear system:

\[
\forall j: \sum_{k \in S_j} y_{kj} \Delta a_{kj} = 0 \quad (20)
\]

\[
\forall j \in J': \sum_{k \in S_j} \Delta a_{kj} = 0 \quad (21)
\]

\[
j = j_k: \sum_{k \in S_j} \Delta a_{kj} + \varepsilon \Delta d'_{j_k} + \Delta \zeta_{j_k} = 0 \quad (22)
\]

\[
\forall i \in S_j^I: \Delta g_i = \sum_{k \in S_j} \Delta a_{ik} Q_{ik} + y_i \Delta b'_{k} - \Delta d'_{k} = 0 \quad (23)
\]

where \( j = 1, \ldots, r - 1 \), \( \Delta \zeta_{j_k} \) is the introduced variable for adjusting \( p_{j^*} \), \( \varepsilon \) is any negative number, and \( \varepsilon \Delta d'_{j_k} \) in (22) is an extra term. The trick of using the extra term can prevent the reoccurrence of Conflict-2, similar to [28].

Then, the linear system (20)–(23) can be further rewritten as

\[
egin{bmatrix} 0 & 0 & UT \\ 0 & O & ET \\ U & E & Q_{S_j S_j} \end{bmatrix} \begin{bmatrix} \Delta b' \\ \Delta d' \\ \Delta a_{S_j} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v_{j_k} \end{bmatrix} \begin{bmatrix} \beta_{b'} \\ \beta_{d'} \\ \beta_{a_{S_j}} \end{bmatrix} \Delta \zeta_{j_k} \quad (24)
\]

\[
\hat{Q}
\]

We let \( \hat{Q}_{\Delta M} \) denote the contracted matrix of \( \hat{Q} \). Similar to the analysis of [29, Th. 7], we can prove that \( \hat{Q}_{\Delta M} \) has the inverse matrix \( \hat{R} \). Then, the linear relationship between \( \Delta b', \Delta d'_{j_k}, \Delta a_{S_j}, \) and \( \Delta \zeta_{j_k} \) can be obtained as follows:

\[
\begin{bmatrix} \Delta b' \\ \Delta d'_{j_k} \\ \Delta a_{S_j} \end{bmatrix} = \hat{R} \begin{bmatrix} 0 \\ v_{j_k} \end{bmatrix} \Delta \zeta_{j_k} \defeq \begin{bmatrix} \hat{\beta}_{b'} \\ \hat{\beta}_{d'} \\ \hat{\beta}_{a_{S_j}} \end{bmatrix} \Delta \zeta_{j_k} \quad (25)
\]

From (25), we have \( \sum_{i \in S_j} \gamma_i \Delta a_{ij} = - (1 + \varepsilon \hat{\beta}_{d'}) \Delta \zeta_{j_k} \), which implies that the control of the adjustment of \( p_{j^*} \) can be achieved by \( \Delta \zeta_{j_k} \).

Substituting (25) into (21), we get the linear relationship between \( \Delta p_{j} \) and \( \Delta \zeta_{j_k} \) as follows:

\[
\Delta p_{j} = \sum_{k \in S_j} \hat{\beta}_k \Delta \zeta_{j_k} \defeq \hat{\rho}_j \Delta \zeta_{j_k} \quad (26)
\]

Obviously, \( \forall j \in J' \), we have \( \hat{\rho}_j = 0 \).

Finally, substituting (25) into (23), we can get the linear relationship between \( \Delta g_i \) (\( \forall i \in S_j^I \)) and \( \Delta \zeta_{j_k} \) as follows:

\[
\Delta g_i = \left( \sum_{k \in S_j} \hat{\beta}_k Q_{ik} + y_i \hat{\beta}_{d'} - \hat{\beta}_d' \right) \Delta \zeta_{j_k} \defeq \hat{\gamma}_i \Delta \zeta_{j_k} \quad (27)
\]

where \( j = 1, \ldots, r - 1 \). Obviously, \( \forall i \in S_j \), we have \( \hat{\gamma}_i = 0 \).

**1) Some Details of SRA:** Similar to RAIA, we need to compute the critical adjustment quantity \( \Delta \zeta_{j_k}^{\text{max}} \) for each restoration adjustment (Fig. 4), such that a certain sample migrates among the sets \( S_j, S_{j^*}, \) and \( S_{j^*} \), or a certain index migrates between \( j_j \) and \( j_{j_k} \). If \( p_{j^*} > 2 \), we will compute the maximal adjustment quantity \( \Delta \zeta_{j_k}^{\text{max}} \), and let \( \Delta \zeta_{j_k} = \Delta \zeta_{j_k}^{\text{max}} \). Otherwise, we will compute the minimal adjustment quantity \( \Delta \zeta_{j_k}^{\text{min}} \), and

\[2\] Similar to the analysis of [29, Th. 9], we can prove that \( (1 + \varepsilon \hat{\beta}_{d'}) \geq 0 \) under the condition that \( \varepsilon < 0 \).
let $\Delta_{jk}^{*} = \Delta_{jk}^{\min}$. Four scenarios are considered to account for such structural changes.

1) A certain $a_i$ in $S_B$ reaches a bound. First, compute the sets: $I^P_B = \{i \in S_B : \beta_i > 0\}$, $I^N_B = \{i \in S_B : \beta_i < 0\}$. Two possible cases are considered for the critical adjustment quantity $\Delta_{jk}^{S_B}$ before a certain sample in $S_B$ moves to $S_R$ or $S_E$:
   a) when $p_{jk} > 2$, the possible weight updates are
   \[
   \Delta a_i^\max = \begin{cases} 
   C - a_i, & \text{if } i \in I_B^P \\
   -a_i, & \text{if } i \in I_B^N 
   \end{cases}
   \]
   then the maximal possible $\Delta_{jk}^{S_B} = \min_{i \in I_B^P \cup I_B^N} (\Delta a_i^\max / \beta_i)$;
   b) when $p_{jk} < 2$, the possible weight updates are
   \[
   \Delta a_i^\max = \begin{cases} 
   -a_i, & \text{if } i \in I_B^P \\
   C - a_i, & \text{if } i \in I_B^N 
   \end{cases}
   \]
   then the minimal possible $\Delta_{jk}^{S_B} = \max_{i \in I_B^P \cup I_B^N} (\Delta a_i^\max / \beta_i)$.

2) A certain $d_j^p$ or $p_j$ reaches zero. We consider two cases for the maximal possible $\Delta a_{j^p}^{J_{j^r}^{\prime}}$ before a certain index in $J_{j^r}^{\prime}$ ($J_R$) migrates to $J_{j^r}^{\prime}$ ($J_R$):
   a) when $p_{jk} > 2$, we compute the sets: $I_{J_{j^r}^{\prime}}^C = \{j \in J_{j^r}^{\prime} : \beta_{d_j^p} < 0\}$, $I_{J_{j^r}^{\prime}}^C = \{j \in J_{j^r}^{\prime} : \beta_{d_j^p} > 0\}, \Delta a_{j^p}^{J_{j^r}^{\prime}} = \min_{j \in I_{J_{j^r}^{\prime}}^C} (\Delta a_{j^p}^{J_{j^r}^{\prime}} / \beta_{d_j^p})$;
   b) when $p_{jk} < 2$, we compute the sets: $I_{J_{j^r}^{\prime}}^C = \{j \in J_{j^r}^{\prime} : \beta_{d_j^p} < 0\}$, $I_{J_{j^r}^{\prime}}^C = \{j \in J_{j^r}^{\prime} : \beta_{d_j^p} > 0\}, \Delta a_{j^p}^{J_{j^r}^{\prime}} = \max_{j \in I_{J_{j^r}^{\prime}}} (\Delta a_{j^p}^{J_{j^r}^{\prime}} / \beta_{d_j^p})$.

3) A certain $g_i$ in $S_R$ or $S_E$ reaches zero. Two cases are considered for the critical adjustment quantity $\Delta a_{j^p}^{S_R, S_E}$ before a certain sample in $S_R$ or $S_E$ moves to $S_S$:
   a) when $p_{jk} > 2$, we compute the sets: $I_{SEP}^C = \{i \in S_E : \gamma_i > 0\}$, $I_{SEP}^C = \{i \in S_R : \gamma_i < 0\}, \Delta a_{j^p}^{S_R, S_E} = \min_{i \in I_{SEP}^C} (\Delta a_{j^p}^{S_R, S_E} / g_i)$;
   b) when $p_{jk} < 2$, we compute the sets: $I_{SEP}^C = \{i \in S_E : \gamma_i < 0\}$, $I_{SEP}^C = \{i \in S_R : \gamma_i > 0\}, \Delta a_{j^p}^{S_R, S_E} = \max_{i \in I_{SEP}^C} (\Delta a_{j^p}^{S_R, S_E} / g_i)$.

4) The restriction (9) on the enlarged $j_k$th two-class training samples is restored. Two cases must be considered for the critical adjustment quantity $\Delta a_{j^p}^{*}$, before $p_{jk}$ and $d_{jk}$ satisfy the inequality restriction (9):
   a) when $p_{jk} > 2$, the maximal possible $\Delta a_{j^p}^{*} = \min (p_{jk} - 2/1 + \epsilon \tilde{\beta}_{d_j^p}^{*}, (d_j^p / \tilde{\beta}_{d_j^p}^{*})^3)$;
   b) when $p_{jk} < 2$, the minimal possible $\Delta a_{j^p}^{*} = (p_{jk} - 2/1 + \epsilon \tilde{\beta}_{d_j^p}^{*})$.

Finally, if $p_{jk} > 2$, the smallest of the four values
\[
\Delta_{jk}^{\max} = \min \left\{ \Delta_{jk}^{S_B} \right\}
\]
constitutes the maximal increment of $\Delta a_{j^p}^{*}$.

\textbf{Theorem 2:} During SRA, the objective function $W$ is strictly monotonically increasing during SRA.

Proof: During SRA, let $\tilde{\beta}_{S_R} = 0$, $\tilde{\beta}_{S_E} = 0$, the superscript $[k]$ denotes the $k$th adjustment, then we have
\[
W^{[k+1]} = \frac{1}{2} \sum_{i_1, i_2 \in S} \left( a_{i_1}^{[k]} + \tilde{\beta}_{i_1}^{[k]} \Delta a_{i_1}^{[k]} \right) \left( a_{i_2}^{[k]} + \tilde{\beta}_{i_2}^{[k]} \Delta a_{i_2}^{[k]} \right) Q_{i_1 i_2}
\]
\[
+ \sum_{j=1}^{r-1} \left( \tilde{\beta}_{d_j}^{[k]} + \tilde{\beta}_{d_{j}}^{[k]} \Delta a_{d_{j}}^{[k]} \right) \sum_{i \in S_j} y_i \left( a_{i}^{[k]} + \tilde{\beta}_{i}^{[k]} \Delta a_{i}^{[k]} \right)
\]
\[
+ \sum_{j=1}^{r-1} \left( \sum_{i \in S_j} a_{i}^{[k]} + \tilde{\beta}_{i}^{[k]} \Delta a_{i}^{[k]} - 2 \right) \left( d_j^{[k]} + \tilde{\beta}_{d_j}^{[k]} \Delta a_{d_{j}}^{[k]} \right)
\]
\[
= W^{[k]} + \sum_{i \in S_j} a_{i}^{[k]} \tilde{\beta}_{i}^{[k]} \Delta a_{i}^{[k]} + \left( \sum_{i \in S_j} y_i \left( a_{i}^{[k]} + \tilde{\beta}_{i}^{[k]} \Delta a_{i}^{[k]} \right) - 2 \right) \tilde{\beta}_{d_j}^{[k]} \Delta a_{d_{j}}^{[k]}
\]
\[
+ \frac{1}{2} \tilde{\beta}_{d_{j}}^{[k]} \left( \Delta a_{d_{j}}^{[k]} \right)^2 \sum_{i \in S_j} \tilde{\beta}_{i}^{[k]} \left( \Delta a_{i}^{[k]} \right)^2
\]
prove that \( \Delta \zeta \) converges to the optimal solution to \( \min_0 W \). We can further prove that the sequence is finite and increasing. Similar to the analysis of [29, Th. 15], we can know that the singularities of \( \hat{Q}_{\omega_c} \) and \( \hat{Q} \), and the finite convergence of ISVOR. To validate the existence of the conflicts, we count the events of Conflict-1 and Conflict-2 during RAIA and SRA, over 50 trials. To investigate the singularities of \( \hat{Q}_{\omega_c} \) and \( \hat{Q} \), we consider the occurrences of SC-1 and SC-2 (Section III-A), during ISVOR over 50 trials. To illustrate the fast convergence of ISVOR empirically, we investigate the average numbers of the iterations of RAIA, and SRA, over 20 trials.

To demonstrate the usefulness of ISVOR, we compare the running time of ISVOR with the batch algorithm (i.e., the SMO algorithm [8]) of SMF and EXC, which are called SMO-SMF and SMO-EXC, respectively, and the incremental algorithm of PSVM (i.e., IPSVM). We also compare the generalization performance of SMF, EXC, PSVM, and MSMF, which correspond to the above four algorithms (i.e., SMO-SMF, SMO-EXC, IPSVM, and ISVOR, respectively). Specifically, to have a better comparison of the generalization performance, two evaluation metrics are utilized to quantify the accuracy and time of ISVOR with the batch algorithm (i.e., the SMO time of ISVOR with the batch algorithm (i.e., the SMO

\[
W[1] = \frac{1}{2} \beta_{d, j}^k \sum_{i \in S^k} \tilde{p}_i^k (\Delta \zeta_{s, [k]}^j)^2 + \tilde{p}_{d, j}^k \Delta \zeta_{s, [k]}^j (\sum_{i \in S^k} a_i^j - 2) = W[1] + \left( \sum_{i \in S^k} a_i^j - 2 - \frac{1}{2} (1 + \tilde{p}_{d, j}^k) \Delta \zeta_{s, [k]}^j \right) \cdot \tilde{p}_{d, j}^k \Delta \zeta_{s, [k]}^j.
\]

In other words, \( W[k+1] - W[k] = (\sum_{i \in S^k} a_i^j - 2 - 1/2(1 + \tilde{p}_{d, j}^k) \Delta \zeta_{s, [k]}^j) \tilde{p}_{d, j}^k \Delta \zeta_{s, [k]}^j \). Similar to the analysis in [29, Th. 9], we can prove that \( (1 + \tilde{p}_{d, j}^k) \Delta \zeta_{s, [k]}^j \geq 0 \) under the condition that \( \varepsilon < 0 \). Similar to [29, Corollary 8], we can prove that \( \Delta \zeta_{s, [k]}^j > 0 \) if \( p_j > 2 \), and \( \Delta \zeta_{s, [k]}^j < 0 \) if \( p_j < 2 \), for each SRA. Similar to the analysis in [29, Th. 7], we can prove \( \tilde{p}_{d, j}^k \Delta \zeta_{s, [k]}^j > 0 \). Furthermore, it is easy to verify that \( (\sum_{i \in S^k} a_i^j - 2 - (1/2)(1 + \tilde{p}_{d, j}^k) \Delta \zeta_{s, [k]}^j) \Delta \zeta_{s, [k]}^j > 0 \). So, \( W[k+1] - W[k] > 0 \). This completes the proof.

Similar to RAIA, we let \( W[1], W[2], W[3], \ldots \) be the sequence generated during SRA. Based on Theorem 2, we can know that \( W[1], W[2], W[3], \ldots \) is a monotonically increasing sequence. Similar to the analysis of [29, Th. 15], we can further prove that the sequence is finite and converges to the optimal solution to \( \min_{0 \leq u_i \leq C} \max_{d, j} W \). Combined with the finite convergence of RAIA, it can be concluded that ISVOR converges to the optimal solution to \( \min_{0 \leq u_i \leq C} \max_{d, j} W \) in a finite number of steps.

### IV. Experimental Setup

#### A. Design of Experiments

To demonstrate the usefulness of ISVOR, and show its advantage in terms of computation efficiency, we conduct a detailed experimental study.

#### B. Implementation

As mentioned in Section III, our incremental scenario is processing one added sample at a time. When a sample is added into the original \( \sum_{j=1}^{n_1} n_j \) training samples, IPSVM and ISVOR update the weights without retraining from scratch based on their corresponding methods. However, the batch algorithms SMO-SMF and SMO-EXC retrain the weights from scratch for the original samples with the added one.

---

**TABLE IV**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Conflict-1</th>
<th>Conflict-2</th>
<th>SC-1</th>
<th>SC-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bank</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Computer Activity</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Friedman</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Census</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>23</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Conflict-1</th>
<th>Conflict-2</th>
<th>SC-1</th>
<th>SC-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Winequality-red</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Winequality-white</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Notes:**

L, P, and G are the abbreviations of Linear, Polynomial, and Gaussian kernels, respectively.

---

4The boolean test \([\cdot]\) is 1 if the inner condition is true, and 0 otherwise.

---

**This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.**

---

**GU et al.: INCREMENTAL SUPPORT VECTOR LEARNING FOR OR**
We implemented ISVOR in MATLAB, and used the MATLAB code of [7] to implement IPSVM directly. We also implemented SMO-SMF in MATLAB. Specially, SMO-EXC was implemented in [8] in C, and this C implementation is used in our experiments. Generally speaking, a program written in C always runs much faster than the same one written in MATLAB, and hence, it is inappropriate to compare the running time of the MATLAB and C programs directly. However, the running time of SMO-EXC is still reported to compare with the other three algorithms, to a certain extent.

Experiments were performed on a 2.5-GHz Intel Core i5 machine with 8-GB RAM. For kernels, the linear kernel $K(x_1, x_2) = x_1 \cdot x_2$, polynomial kernel $K(x_1, x_2) = (x_1 \cdot x_2 + 1)^d$, and Gaussian kernel $K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/2\sigma^2)$ are used in our experiments, where the parameters $d$ and $\sigma$ are set to 2, and 2.2361, respectively, unless otherwise specified. In addition, the regularization parameter $C$ is fixed to 10 unless otherwise specified. The parameter, $\varepsilon$ of SRA, is fixed to $-1$ throughout all the experiments. The tolerance parameters of SMO-SMF and SMO-EXC are all fixed to $10^{-10}$.

C. Data Sets

Table III summarizes the characteristics of eight data sets used in our experiments, where five data sets are from regression problems, and the other three are real OR data sets. The detailed description of each data set is stated as follows.

1) Regression Data Sets: The data sets Bank, Computer Activity, Census, and Abalone are available at http://www.gatsby.ucl.ac.uk/~chuwei/ordinalregression.html, but is independent with the structural changes of the sets $S_S$, $S_R$, $S_E$, $J'$, and $J'$.  

Fig. 5. Average numbers of iterations of RAIA and SRA on the different data sets. (a) Bank. (b) Computer Activity. (c) Friedman. (d) Census. (e) Abalone. (f) Winequality-red. (g) Winequality-white. (h) Spine Image.
the data set Friedman is available at http://mldata.org/repository/data/viewslug/friedman-datasets-fri_c2_250_5/.

Originally, these benchmark data sets are used for metric regression problems. To make them suitable for OR problems, we discretized the target values of the samples into five ordinal quantities using equal interval binning.

2) Real OR Data Sets: Winequality-red and Winequality-white are from the UCI machine learning repository [30]. They are real OR data sets.

The spine image data set was collected by us from the London, Canada. The data set is to diagnose a degenerative disc disease based on Pfirrmann et al. [31] grading system (Grade 1 healthy and Grade 5 advanced), depending on five image texture features (including contrast, correlation, energy, homogeneity, and mean signal intensity) quantified from magnetic resonance imaging. The data set contains 350 records, where 20, 137, 90, 82, and 21 records were marked Grade 1, 2, 3, 4, and 5, respectively, by an experienced radiologist.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Usefulness of ISVOR

1) Existence of the Conflicts and the Singularities: When the size of the original training set is 10, 15, 20, 25, and 30, for each data set, Table IV presents the corresponding numbers of occurrences of Conflict-1 and Conflict-2. From this table, we find that the two kinds of conflicts happen with a high probability on the linear and polynomial kernels, and especially on the Spine Image data set. This is because the lower the dimension of the data or the RKHS, the higher the probability that all the margin support vectors in \( S_{jc} \) have one and the same label. Thus, it is essential to handle the conflicts during the incremental SVOR learning. Our ISVOR can avoid these conflicts effectively.

Table IV also presents the numbers of occurrences of SC-1 and SC-2 on the eight data sets, where the original training sample size of each data set is also set as 10, 15, 20, 25, and 30, respectively. From this table, we find that SC-2 happens with a higher probability than SC-1 does.
Although SC-1 happens with a low probability, the possibility of the occurrences still cannot be excluded. Thus, it is very significant that ISVOR handles the two singular cases. Our ISVOR can handle the singularities of $\hat{Q}((\alpha^r)^2$ and $\hat{\alpha}$ effectively.

2) Finite Convergence: We randomly select the samples with the data size shown in the horizontal axis of Fig. 5 for each data set, such as 150, 300, 450, 600, 750, 900, 1050, 1200, and so on, as the original training set, and try to demonstrate the average numbers of the iterations of RAIA, and SRA, when incorporating a new sample into the original training set. Specifically, the average number is obtained by counting the iterations for the increased extended training sample in $S_{\text{new}}$, over 20 trials, regardless of whether whose initial value of the function $g$ is less than 0, or not. Fig. 5 shows the average numbers of iterations of RAIA, and SRA, with different kernels on the different data sets. It is obvious that RAIA and SRA exhibit quick convergence for all data sets and kernels, especially with the Gaussian kernel. Based on Fig. 5, we can conclude that ISVOR avoids the infeasible updating paths as far as possible, and successfully converges to the optimal solution with a fast convergence speed.

B. Comparison With Other Methods

1) Running Time: We randomly select the samples with the data size shown in the horizontal axis of Fig. 6 for each data set as the original training set, similar to the setting in the experiments of finite convergence, and record the running time of SMO-SMF, SMO-EXC, IPSVM, and ISVOR, when incorporating a new sample. Fig. 6 shows the average running time of SMO-SMF, IPSVM, SMO-EXC, and ISVOR with different kernels on the different data sets, over 20 trails. The results clearly show that our ISVOR is generally much faster than SMO-SMF, and IPSVM, on all the data sets and kernels. It should be noted that SMO-EXC is implemented by C. It is inappropriate to compare the running time of the two implementations in different languages directly, as mentioned in Section IV-B. Even so, we still observe that ISVOR is faster than SMO-EXC on the Bank, Friedman, Census, Abalone, Winequality-red, and Winequality-white data sets. If the time of SMO-EXC is multiplied by the ratio between the MATLAB implementation and the C implementation, it would be obvious that ISVOR is much faster than SMO-EXC. To sum up, we can conclude that ISVOR is much faster than SMO-SMF, IPSVM, and SMO-EXC.

2) Generalization Performance: Gaussian kernel is used for comparing the generalization performance. A 5-fold cross validation with a two-step grid search strategy is used to determine the optimal values of model parameters (the Gaussian kernel parameter $\sigma$ and the regularization factor $C$) involved in the problem formulations: the initial search is done on a $3 \times 7$ coarse grid linearly spaced in the region $\{(\log_{10} C, \log_{10} 2\sigma^2)|1 \leq \log_{10} C \leq 3, \,-3 \leq \log_{10} 2\sigma^2 \leq 3\}$, followed by a fine search on a $9 \times 9$ uniform grid linearly spaced by 0.2 in the $(\log_{10} C, \log_{10} 2\sigma^2)$ space.

We randomly select the samples with the data size shown in Table V for each data set as the validation set. The optimal
parameter values of each formulation for MAE and MZE are computed by the above 5-fold cross validation procedure on the validation set. The remaining samples of each data set are further randomly split into a training set, and a test set, with the data sizes shown in Table V, over 20 trials. For each trial, both MAE and MZE on the test set are computed by a model learned from the training set with the corresponding optimal parameter values computed by the 5-fold cross validation procedure. Fig. 7(a) and (b) shows the MAEs and MZEs for each formulation for MAE and MZE are the parameter values of each formulation for MAE and MZE.

VI. CONCLUSION

To extend AONSV to the SVOR formulation (3), we first presented a modified formulation of SVOR based on maximizing sum-of-margins which has multiple constraints of the mixture of an equality and an inequality, then proposed its incremental algorithm. We also provided the finite convergence analysis for it. Numerical experiments showed that the incremental algorithm can converge to the optimal solution in a finite number of steps, and is faster than the existing batch and incremental SVOR algorithms. Meanwhile, the modified formulation has better accuracy than the existing incremental SVOR algorithm, and is as accurate as the SMF of Shashua and Levin [6].

Theoretically, the decremental SVOR learning can also be designed in a similar manner. Because the proposed incremental SVOR algorithm can handle multiple inequality constraints, and can tackle the conflicts between the equality and inequality constraints, we believe that it can be extended to other SVOR formulations of [6] and [8] based on the framework of parametric quadratic programming [32]. In the future, we hope to provide the feasibility analysis for the incremental SVOR algorithm, similar to the work of [29], and extend the incremental SVOR algorithm to multiple kernel learning [33].

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments on the early versions of this paper, and Dr. J. Qin for assistance with the data analytics Cloud at Shared Hierarchical Academic Research Computing Network provided through the Southern Ontario Smart Computing Innovation Platform.

REFERENCES


Bin Gu (M’08) received the B.S. and Ph.D. degrees in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005 and 2011, respectively.

He joined the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, in 2010, as a Lecturer, where he is currently an Associate Professor. He is also currently a Post-Doctoral Fellow with the University of Western Ontario, London, ON, Canada. His current research interests include machine learning, data mining, and medical image analysis.

Victor S. Sheng (M’11) received the Ph.D. degree in computer science from the University of Western Ontario, London, ON, Canada, in 2007.

He was an Associate Research Scientist and NSERC Post-Doctoral Fellow in Information Systems with Stern Business School, New York University, New York, NY, USA. He is an Assistant Professor with the Department of Computer Science, University of Central Arkansas, Conway, AR, USA, and the founding Director of the Data Analytics Laboratory. His current research interests include data mining, machine learning, and related applications.

Prof. Sheng is a member of the IEEE Computer Society. He is a PC Member for a number of international conferences and a reviewer for several international journals. He was a recipient of the Best Paper Runner-Up Award from the 2008 International Conference on Knowledge Discovery and Data Mining, and the Best Paper Award from the 2011 Industrial Conference on Data Mining.

Keng Yeow Tay received the Medical degree from the University of New South Wales, Sydney, NSW, Australia.

He had his subspecialty fellowship training in diagnostic neuroradiology at Addenbrooke’s Hospital, Cambridge, U.K., and the University of Toronto, Toronto, ON, Canada. He is currently an Assistant Professor of Radiology with the Schulich School of Medicine and Dentistry, University of Western Ontario, London, ON. His current research interests include diagnostic neuroradiology, and head and neck radiology.

Walter Romano received the Medical degree from the University of Western Ontario (UWO), London, ON, Canada, and the Fellowship in high-risk obstetrical ultrasound and MRI from the University of Michigan, Ann Arbor, MI, USA.

He is currently an Adjunct Professor of Radiology with the Schulich School of Medicine at UWO. He is the Medical Director of Imaging with the Saint Thomas Elgin General Hospital, Thomas, ON.

Shuo Li received the bachelor’s and master’s degrees in computer science from Anhui University, Hefei, China, and the Ph.D. degree in computer science from Concordia University, Montreal, QC, Canada.

He is a Research Scientist and Project Manager with General Electric (GE) Healthcare, Mississauga, ON, Canada. He is also an Adjunct Research Professor with the University of Western Ontario, London, ON, Canada, and an Adjunct Scientist with the Lawson Health Research Institute, London. He is currently leading the Digital Imaging Group of London as the Scientific Director. He serves as a Guest Editor and an Associate Editor in prestigious journals in the field. His current research interests include intelligent medical imaging systems, with a main focus on automated medical image analysis and visualization.

Dr. Li’s Ph.D. thesis received the Doctoral Prize from Concordia University, which gives to the most deserving graduating student in the faculty of engineering and computer science. He was a recipient of several GE internal awards.