**Abstract**

Incremental and decremental learning (IDL) deals with the tasks where new data arrives sequentially as a stream or old data turns unavailable continually due to the privacy protection. Existing IDL methods mainly focus on support vector machine and its variants with linear-type loss. There are few studies about the quadratic-type loss, whose Lagrange multipliers are unbounded and much more difficult to track. In this paper, we take the latest statistical learning framework optimal margin distribution machine (ODM) which involves a quadratic-type loss due to the optimization of margin variance, for example, and equip it with the ability to handle IDL tasks. Our proposed ID-ODM can avoid updating the Lagrange multipliers in an infinite range by determining their optimal values beforehand so as to enjoy much more efficiency. Moreover, ID-ODM is also applicable when multiple instances come and leave simultaneously. Extensive empirical studies show that ID-ODM can achieve 9.1x speedup on average with almost no generalization lost compared to retraining ODM on new data set from scratch.

**1 Introduction**

For machine learning tasks, once data changes, the models should be refined. However, retraining models from scratch usually incurs unbearable cost or is even impossible in some situations, which gives rise to the incremental and decremental learning (IDL) [Schlimmer and Fisher, 1986; Cauwenberghs and Poggio, 2000]. This paradigm can efficiently update the model and be applied in many learning problems, just to name a few, anomaly detection [Laxhammar and Falkman, 2014; Nguyen et al., 2019], concept drifting [Syed et al., 1999; Gâlmeanu and Andonie, 2022], active learning [Ertekin et al., 2007; Huang et al., 2020], privacy protection [Nguyen et al., 2020; Sekhari et al., 2021], and model estimation [Cherubin et al., 2021].

Existing IDL methods mainly focus on support vector machine (SVM) [Cortes and Vapnik, 1995] and its variants.

We summarize the representative works of this line in Table 1. Based on the path following method, IDSVM [Cauwenberghs and Poggio, 2000] takes both incremental learning and decremental learning into account to update SVM. MID-SVM [Karasuyama and Takeuchi, 2009] is a batch version of IDSVM, AONSVM [Gu et al., 2012] and FISVDD [Jiang et al., 2019] extends the ν-SVM [Schölkopf et al., 2000] and SVDD [Tax and Duin, 2004] to incremental learning, respectively. All these models use hinge loss. Besides, IL-S3VM [Gu et al., 2018] and ILTSVM [Chen et al., 2023] are incremental semi-supervised SVMs with symmetric hinge loss or its variant ramp loss.

On the one hand, the aforementioned methods mainly focus on large margin models, but the latest theoretical studies [Gao and Zhou, 2013] disclose that the margin distribution is much more critical to the generalization performance. On the other hand, these methods are limited to linear-type loss, in which the hinge loss is sensitive to outliers and symmetric hinge loss leads to non-convex optimization problems. In contrast, the quadratic-type loss is smoother and numerically easier to deal with. It results in a strictly diagonally dominant matrix in dual quadratic form which is always invertible, thus relaxes the restriction of linear-type loss that data should be linearly independent. As far as we know, there are few IDL methods concerning models with quadratic-type loss.

In this paper, we take the latest statistical learning framework optimal margin distribution machine (ODM) [Zhang and Zhou, 2019] which involves a quadratic-type loss due to the optimization of margin variance as an example and equip it with the ability to handle IDL tasks. The drawback of quadratic-type loss lies in the optimization aspect, that is the Lagrange multipliers are unbounded and much more difficult to track when data varies, and existing IDL methods with linear-type loss can hardly be adapted to these circumstances. To overcome this difficulty, we estimate the optimal value of the Lagrange multiplier of the varying data in advance to avoid the optimization in an infinite range, and then update along the fastest direction to make the interruption by breakpoints less frequent. We also theoretically analyze the time complexity and convergence of our proposed incremental and decremental ODM (ID-ODM). The remarkable advantages are as follows:

- It is the first attempt to implement IDL for kernel methods optimizing the margin distribution.

**Incremental and Decremental Optimal Margin Distribution Learning**

Li-Jun Chen, Teng Zhang*, Xuanhua Shi and Hai Jin

National Engineering Research Center for Big Data Technology and System Services Computing Technology and System Lab, Cluster and Grid Computing Lab School of Computer Science and Technology, Huazhong University of Science and Technology, China

{ljchen, tengzhang, xhshi, hjin}@hust.edu.cn

*Corresponding Author
Table 1: An overview of the existing literature on incremental and decremental learning, in which BC, RG, OR, OCC, and SSC are the abbreviations of binary classification, regression, ordinary regression, one-class classification, and semi-supervised classification, respectively. "single" and "multiple" represent the number of varying instances.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Models</th>
<th>Tasks</th>
<th>Loss</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISVM [Rüping, 2001; Liang and Li, 2009]</td>
<td>SVM</td>
<td>BC</td>
<td>hinge loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>ID SVM [Cawuhenberghs and Poggio, 2000]</td>
<td>SVM</td>
<td>BC</td>
<td>hinge loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>INSVR [Gu et al., 2015b]</td>
<td>ν-SVR</td>
<td>RG</td>
<td>hinge loss, linear</td>
<td>multiple, inc. &amp; dec.</td>
</tr>
<tr>
<td>ISVOR [Gu et al., 2015a]</td>
<td>SVOR</td>
<td>OR</td>
<td>hinge loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>FISVDD [Jiang et al., 2019]</td>
<td>SVDD</td>
<td>OCC</td>
<td>hinge loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>oSVMR [Wang and Vucetic, 2009]</td>
<td>SVMR</td>
<td>BC</td>
<td>ramp loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>IL-S SVM [Gu et al., 2018]</td>
<td>S SVM</td>
<td>SSC</td>
<td>symmetric hinge loss, linear</td>
<td>single, incremental</td>
</tr>
<tr>
<td>ILTSVM [Chen et al., 2023]</td>
<td>TSVM</td>
<td>SSC</td>
<td>ODM loss, quadratic</td>
<td>multiple, inc. &amp; dec.</td>
</tr>
</tbody>
</table>

2 Preliminaries

We first introduce some conventions and notations used throughout the paper. The normal font letters (e.g., \( y \)) denote scalars. The boldface letters (e.g., \( w \) and \( \mathbf{W} \)) denote vectors and matrices, respectively. The upper case letters with mathematical font (e.g., \( S \)) indicate the sets. Particularly, \( |m| \) is defined as the integer set \( \{1, 2, \ldots, m\} \). \( m_S \) indicates the vector consisting of the entries of \( w \) specified by index set \( S \). Similarly, \( \mathbf{W}_{S,S'} \) is the submatrix of \( \mathbf{W} \) specified by row index set \( S \) and column index set \( S' \).

For a binary classification problem, let \( \mathcal{X} \subseteq \mathbb{R}^d \) and \( \mathcal{Y} = \{1, -1\} \) denote the instance space and label set, respectively. The training set \( \{(x_i, y_i)\}_{i=1}^m \subseteq (\mathcal{X} \times \mathcal{Y})^m \) is drawn i.i.d from some underlying distribution on \( \mathcal{X} \times \mathcal{Y} \). Let \( \phi : \mathcal{X} \mapsto \mathbb{H} \) be a feature mapping associated with some positive definite kernel \( \kappa \) where \( \mathbb{H} \) is the corresponding reproducing kernel Hilbert space (RKHS). The hypothesis is defined as the linear function \( f(x) = \langle w, \phi(x) \rangle_{\mathbb{H}} \).

This linear decision function naturally leads to the definition of margin, i.e., \( \gamma(x, y) = y\langle w, \phi(x) \rangle_{\mathbb{H}} \). Notice that \( f \) misclassifies \( (x, y) \) if and only if it produces a negative margin, thus it represents the confidence of the prediction results. Recent studies on margin theory reveal that margin distribution is more important to generalization than a single margin, which gives rise to the ODM:

\[
\min_{w, \xi_i} \frac{1}{2} \|w\|_H^2 + \frac{\lambda}{2} \sum_{i \in [m]} (\xi_i^2 + \epsilon_i^2),
\]

s.t. \( 1 - \theta - \xi_i \leq \gamma_i \leq 1 + \theta + \epsilon_i \), \( \forall i \in [m] \),

where \( \gamma_i = y_i f(x_i) \) and \( \lambda \) is the trade-off hyperparameter. The margin mean has been fixed as 1 since scaling \( w \) does not affect the decision boundary. The slack variables \( \xi_i \) and \( \epsilon_i \) represent the deviation from the strip centered at margin mean with width \( 2\theta \). The hyperparameter \( \theta \) is to tolerate the tiny deviations smaller than \( \theta \). When it is set as 0, the third term is exactly the margin variance.

Introducing the Lagrange multipliers \( \zeta_i \geq 0 \) and \( \beta_i \geq 0 \) for constraints leads to the Lagrangian of Eqn. (1):

\[
L = \frac{1}{2} \|w\|_H^2 - \sum_{i \in [m]} \zeta_i(\gamma_i - (1 - \theta - \xi_i))
\]

\[
+ \frac{\lambda}{2} \sum_{i \in [m]} (\xi_i^2 + \epsilon_i^2) + \sum_{i \in [m]} \beta_i(\gamma_i - (1 + \theta + \epsilon_i)).
\]

The KKT conditions [Boyd and Vandenberge, 2004] are

\[
\zeta_i(\gamma_i - (1 - \theta + \xi_i)) = 0,
\]

\[
\beta_i(\gamma_i - (1 + \theta + \epsilon_i)) = 0.
\]

The analysis on complementary slackness conditions, i.e., Eqn. (2)-(3), leads to the partition of data into three sets:

\[ L = \{i \mid \gamma_i < 1 - \theta, \xi_i > 0, \beta_i = 0\}, \]

\[ C = \{i \mid \gamma_i \in [1 - \theta, 1 + \theta], \xi_i = 0, \beta_i = 0\}, \]

\[ R = \{i \mid \gamma_i > 1 + \theta, \xi_i = 0, \beta_i > 0\}, \]

which are shown in Figure 1(a), respectively. Besides, let \( \alpha_i = \zeta_i - \beta_i \) and \( S = L \cup R \) is the index set of instances whose \( \alpha_i \neq 0 \), the decision function can be rewritten as

\[
f(x_i) = \langle w, \phi(x_i) \rangle_{\mathbb{H}} = \sum_{j \in S} \alpha_j y_j \kappa_{ij}
\]

with \( \kappa_{ij} = \kappa(x_i, x_j) \).
we have

The relationship between $\alpha_S$ and $\alpha_k$. Notice that $1 \pm \theta$ are constants, to make the KKT conditions hold, i.e.,

$$
\gamma_i + \xi_i = 1 - \theta, \quad \forall i \in \mathcal{L},
$$

$$
\gamma_i - \epsilon_i = 1 + \theta, \quad \forall i \in \mathcal{R},
$$

the Lagrange multiplier $\alpha_S$ should vary with $\alpha_k$ such that

$$
\Delta \gamma_i + \Delta \epsilon_i = 0, \quad \forall i \in \mathcal{L},
$$

$$
\Delta \gamma_i - \Delta \epsilon_i = 0, \quad \forall i \in \mathcal{R},
$$

where $\Delta$ denotes the variations of variables. With Eqn. (4), we have

$$
\Delta \gamma_i = \gamma_i \Delta f(x_i) = \sum_{j \in S} \Delta \alpha_j y_i y_j \kappa_{ij} + \Delta \alpha_k y_i y_k \kappa_{ik}. \tag{7}
$$

Notice that $\alpha_i = \lambda \xi_i$ for $\forall i \in \mathcal{L}$ and $\alpha_i = -\lambda \epsilon_i$ for $\forall i \in \mathcal{R}$, we can derive the following linear system

$$
\sum_{j \in S} \Delta \alpha_j h_{ij} + \Delta \alpha_k h_{ik} + \Delta \alpha_i / \lambda = 0, \quad \forall i \in S, \tag{8}
$$

where $h_{ij} = y_i y_j \kappa_{ij}$. Denote $H = (h_{ij})_{i,j \in [k]}$, then Eqn. (8) can be rewritten in a matrix form

$$
(H_{SS} + I_{[S]} / \lambda) \Delta \alpha_S = -H_{Sk} \Delta \alpha_k,
$$

which implies that $\Delta \alpha_S$ varies linearly with $\Delta \alpha_k$:

$$
\Delta \alpha_S = -Q_S^{-1} H_{Sk} \Delta \alpha_k = t \Delta \alpha_k, \tag{9}
$$

where $Q_S = H_{SS} + I_{[S]} / \lambda$ and $t = -Q_S^{-1} H_{Sk}$.

Update $\alpha_k$. By substituting Eqn. (9) into Eqn. (7) to eliminate $\Delta \alpha_S$, the remaining variable is only $\Delta \alpha_k$. We can monitor the variation of margin with $\Delta \alpha_k$ for all instances.

Different from previous works like [Cauwenberghs and Poggio, 2000] in which $\alpha_k$ belongs to $[0, C]$ and can be simply updated by increasing gradually from 0 to $C$, here $\alpha_k \in \mathbb{R}$ is unbounded, and we do not even know whether to increase or decrease $\alpha_k$. Fortunately, if $(x_k, y_k)$ is successfully added, by KKT conditions, we have

$$
\gamma_k + \Delta \gamma_k^* + (\alpha_k + \Delta \alpha_k^*) / \lambda = d, \tag{10}
$$

where

$$
d = \begin{cases} 
1 - \theta, & \gamma_k < 1 - \theta, \\
1 + \theta, & \gamma_k > 1 + \theta.
\end{cases} \tag{11}
$$

With Eqn. (7) and Eqn. (9), we have a closed-form solution

$$
\Delta \alpha_k^* = \frac{d - \gamma_k - \alpha_k / \lambda}{1 + 1 / \lambda + H_{ks} t}. \tag{12}
$$

The above analysis is based on the assumption that the three index sets $\mathcal{L}, \mathcal{C},$ and $\mathcal{R}$ remain unchanged when $\alpha_k$ is varying, thus Eqn. (12) is not an actual optimal solution, but it can provide a credible update direction. In practice, we can let $\Delta \alpha_k = \eta \Delta \alpha_k^*$ and increases $\eta$ from 0 to 1 gradually. At the same time, we monitor the three index sets. When one of the following events occurs

1. $i \in S$ migrates to $\mathcal{C}$, i.e., nonzero $\alpha_i$ turns to 0,
2. $i \in \mathcal{C}$ migrates to $S$, i.e., $f(x_i)$ reaches the boundary of the strip as shown in Figure 1, a breakpoint is detected and sequentially perform the following four steps
3. Stop the increase of $\eta$.
4. Update the index sets $\mathcal{L}, \mathcal{C},$ and $\mathcal{R}$.
5. Record the current $\alpha_k$ and $\alpha_S$.
6. Update $H, t$, and $\Delta \alpha_k$.

The above procedure is repeated until no breakpoint is detected during the increase of $\eta$ from 0 to 1, then $(x_k, y_k)$ can be added to $\mathcal{L}$ or $\mathcal{R}$ based on current $\gamma_k$.

3.2 Decremental Case

Suppose the invalid instance is $(x_k, y_k)$. If $k \in \mathcal{C}$, i.e., $\alpha_k = 0$, it has no contribution to the model and can be removed directly. Thus without loss of generality, we assume $k \in S$. Now we need to turn $\alpha_k$ to 0 to totally eliminate its effect on the model. Analogous to the incremental case, we first update $S = S \setminus \{k\}$ and derive the relationship between $\alpha_S$ and $\alpha_k$, and then show how to update $\alpha_S$ by varying $\alpha_k$.

The relationship between $\alpha_S$ and $\alpha_k$. Since we should make KKT conditions hold at any time, the deletion of $\alpha_k$ from $S$ can be viewed as the reverse process of addition of $\alpha_k$ to $S$. Again, $\Delta \alpha_S$ and $\Delta \alpha_k$ satisfy

$$
(H_{SS} + I_{[S]} / \lambda) \Delta \alpha_S = -H_{Sk} \Delta \alpha_k.
$$

Since $\Delta \alpha_S$ changes linearly with $\Delta \alpha_k$, we can trace the variation of the margin with $\Delta \alpha_k$. 

3 ID-ODM for Single Instance Varying

In this section, we investigate the case when single instance comes or leaves.

3.1 Incremental Case

Suppose the new instance is $(x_k, y_k)$. If $\gamma_k \in [1 - \theta, 1 + \theta]$, according to Eqn. (4), this instance can be directly added to $\mathcal{C}$ without any update. Thus without loss of generality, we assume $\gamma_k \notin [1 - \theta, 1 + \theta]$ and initialize the Lagrange multiplier $\alpha_k$ as 0. Next, we first derive the relationship between $\alpha_S$ and $\alpha_k$, and then detail how to update $\alpha_S$ by varying $\alpha_k$ gradually.

(a) Add multiple new instances (red points). (b) Remove multiple invalid instances (black points). (c) Add new instances and remove invalid instances simultaneously.

Figure 1: Description of different tasks.
Update $\alpha_k$. The update is now reduced to determine $\Delta \alpha_k$. Distinct from the incremental case, the invalid data vanishes and the algorithm terminates with $\Delta \alpha_k^* = -\alpha_k$.

As a result, we can update $\alpha_k$ with $\Delta \alpha_k = -\eta \alpha_k$. Like the incremental case, each update is comprised of four steps when we keep watch on breakpoints. The whole procedure stops when $\eta = 1$.

4 ID-ODM

In this section, we first present the general ID-ODM for multiple instances varying and then analyze the time complexity and convergence.

4.1 Multiple Instances Varying

Without loss of generality, we assume that there are $p$ instances $\{(x_i, y_i)\}_{i \in I}$ becoming available and $q$ instances $\{(x_j, y_j)\}_{j \in J}$ turning invalid where

$E = \{m + 1, m + 2, \ldots, m + p\}, \ J \subseteq S.$

We denote $A = E \cup J$ to represent all varying instances for convenience. The most straightforward way is to view each single instance individually and invoke the method in Section 3.1 and 3.2 for $p$ times and $q$ times, respectively. Nevertheless, this scheme is computationally inefficient as shown in Figure 2(a).

![Figure 2: Different schemes to process multiple instances. (a) Update $\alpha_k$, $k \in A$ individually. (b) Update $\alpha_{\mathcal{E}}$ and $\alpha_{\mathcal{T}}$ simultaneously.](image)

One smarter way is to update $\alpha_{\mathcal{E}}$ and $\alpha_{\mathcal{T}}$ simultaneously as shown in Figure 2(b). Under this circumstance, multiple new instances satisfy the optimal conditions, and multiple invalid instances are removed at the same time. In the same manner as before, we first update $S = S \setminus J$ and derive the relationship between $\alpha_{\mathcal{S}}$ and $\alpha_{\mathcal{A}}$, and then show how to update $\alpha_{\mathcal{E}}$ and $\alpha_{\mathcal{T}}$ simultaneously.

The relationship between $\alpha_{\mathcal{S}}$ and $\alpha_{\mathcal{A}}$. Different from the single instance case where $\Delta \alpha_k$ is a scalar, now $\Delta \alpha_S$ varies with the vector $\Delta \alpha_{\mathcal{A}}$. The margin change of $(x_i, y_i)$ is described as

$$
\Delta \gamma_i = \sum_{j \in S} \Delta \alpha_j y_i y_j \kappa_{ij} + \sum_{k \in A} \Delta \alpha_k y_i y_k \kappa_{ik} = H_{iS} \Delta \alpha_S + H_{iA} \Delta \alpha_{\mathcal{A}}
$$

In order to make KKT conditions hold, $\Delta \alpha_S$, $\Delta \alpha_{\mathcal{E}}$, and $\Delta \alpha_{\mathcal{T}}$ should satisfy

$$(H_{SS} + I/\lambda) \Delta \alpha_S + H_{SA} \Delta \alpha_{\mathcal{A}} = 0.$$ 

Thus, $\Delta \alpha_S$ can be traced by $\Delta \alpha_{\mathcal{E}}$ and $\Delta \alpha_{\mathcal{T}}$ according to

$$\Delta \alpha_S = -Q_S^{-1} H_{SA} \Delta \alpha_{\mathcal{A}} = [T_{SE} \ T_{ST}] \Delta \alpha_{\mathcal{E}} \Delta \alpha_{\mathcal{T}}$$

(14)

where $T_{SE} = -Q_S^{-1} H_{SE}$ and $T_{ST} = -Q_S^{-1} H_{ST}$.

Update $\alpha_{\mathcal{A}}$. The critical point to update the model lies in identifying $\Delta \alpha_{\mathcal{A}}$. Intuitively, $\alpha_{\mathcal{T}}$ is supposed to update along the opposite direction of $\alpha_{\mathcal{T}}$ to remove the invalid data. In other words, $\Delta \alpha_{\mathcal{T}} = -\eta \alpha_{\mathcal{T}}$. As for $\alpha_{\mathcal{E}}$, we determine the update direction $\Delta \alpha_{\mathcal{E}}$ by keeping KKT conditions w.r.t. the new instances. Since multiple instances arrive, Eqn. (10) can be recast in

$$d_{\mathcal{E}} = \gamma_{\mathcal{E}} + H_{\mathcal{ES}} \Delta \alpha_{\mathcal{S}} + H_{\mathcal{EA}} \Delta \alpha_{\mathcal{A}} \Delta \alpha_{\mathcal{E}} + (\alpha_{\mathcal{E}} + \Delta \alpha_{\mathcal{E}})/\lambda$$

$$= (Q_{\mathcal{E}} + H_{\mathcal{ES}} T_{\mathcal{SE}})^{-1} (d_{\mathcal{E}} - \gamma_{\mathcal{E}} - \alpha_{\mathcal{E}})$$

$$- (Q_{\mathcal{E}} + H_{\mathcal{ES}} T_{\mathcal{SE}})^{-1} (H_{\mathcal{ES}} T_{\mathcal{ST}} + H_{\mathcal{ET}}) \Delta \alpha_{\mathcal{T}}.$$ 

It seems that $\Delta \alpha_{\mathcal{E}}^*$ is relevant to $\Delta \alpha_{\mathcal{T}}^*$, but when no breakpoint is produced, $\Delta \alpha_{\mathcal{T}}^* = -\alpha_{\mathcal{T}}$ is fixed. Consequently, we update $\alpha_{\mathcal{A}}$ via

$$\Delta \alpha_{\mathcal{A}} = [\Delta \alpha_{\mathcal{E}} \Delta \alpha_{\mathcal{T}}] = \frac{\Delta \alpha_{\mathcal{E}}}{\Delta \alpha_{\mathcal{T}}}$$

and track $\Delta \alpha_S$ and $\Delta f(x)$ by searching for the minimal $\eta$ which will lead to a breakpoint or the termination of ID-ODM. Algorithm 1 summarizes the pseudo-code for ID-ODM.

4.2 Time Complexity

According to the Algorithm 1, the main time cost is caused by computing the inverse of the matrix $Q_S$. It is well accepted that the time complexity of computing the inverse of a $n$-order square matrix is $O(n^3)$. If we calculate $Q_S^{-1}$ at each iteration, the time complexity is $O(c(m + p - q)^3)$, where $c$ is the number of breakpoints. Hence it is very time-consuming. However, by resorting to Sherman-Morrison-Woodbury formula [Golub and Van Loan, 1983], we can reduce the time cost to $O(c(m + p - q)^2)$.

Suppose a new instance $(x_c, y_c)$ is added to $S$, let $S' = S \cup \{c\}$, then we have

$$Q_{S'}^{-1} = Q_S^{-1} - Q_S^{-1} H_{Sc} [H_{cc} + (1/\lambda) - H_{Sc} Q_S^{-1} H_{Sc}]^{-1} Q_S^{-1}.$$ 

The Schur’s complement [Zhang, 2006] of $Q_S$ is $S = H_{cc} + 1/\lambda - H_{Sc} Q_S^{-1} H_{Sc}$. According to the Sherman-Morrison-Woodbury formula,

$$Q_{S'}^{-1} = \begin{bmatrix} Q_S^{-1} & \Gamma S & -Q_S^{-1} H_{Sc} Q_S^{-1} S^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} Q_S^{-1} & 0 \ 0 & 1 \ v' & 1 \ \end{bmatrix} S^{-1} \begin{bmatrix} v' & 1 \ \end{bmatrix},$$

(15)

where $v = -Q_S^{-1} H_{Sc}$.

As for the decremental case, we can exchange the roles of $S$ and $S'$, and calculate $Q_{S'}^{-1}$ in the same manner.
Algorithm 1 ID-ODM

1: Input: The data set \(\{(x_i, y_i)\}_{i=1}^m\), the hyperparameters \(\{\lambda, \theta, \sigma\}\), the given model \(\alpha_S\), the varying instances \(\{(x_i, y_i)\}_{i \in \Xi, X}\).
2: Calculate the margin \(\gamma\) and the matrix \(Q^{-1}\), \(H\), \(T\).
3: for \(k \in \mathcal{E}\) do
4: if \(\gamma_k \in [1 - \theta, 1 + \theta]\) then
5: \(C = C \cup \{k\}, \mathcal{E} = \mathcal{E} \setminus \{k\}\).
6: end if
7: end for
8: for \(k \in \mathcal{I}\) do
9: if \(k \in C\) then
10: \(C = C \setminus \{k\}, \mathcal{I} = \mathcal{I} \setminus \{k\}\).
11: end if
12: end for
13: \(S = S \setminus \mathcal{I}\).
14: while true do
15: Calculate the update direction \(\Delta\alpha_A\).
16: Calculate the minimal \(\eta\) leading to breakpoints.
17: if \(\eta < \eta_{\text{th}}\) then
18: Update the index sets \(S\) and \(C\).
19: Update \(Q^{-1}_S\), \(T\), \(\gamma\), \(\alpha_S\) and \(\alpha_A\).
20: else
21: Break.
22: end if
23: end while
24: Output: The updated model \(\alpha_S\).

4.3 Convergence

According to Eqn. (9), ID-ODM depends on the existence of \(Q^{-1}_S\). Fortunately, for quadratic-type loss, \(Q\) is strictly diagonally dominant and positive definite, thus its inverse always exists.

When an instance \((x_k, y_k)\) comes or leaves, \(\alpha_k\) varies, the classification hyperplane evolves, and \(\alpha_S\) monotonically increases or decreases [Laskov et al., 2006]. To guarantee the convergence of Algorithm 1, we should further prove the monotonic update of \(\alpha_S\) is strict, that is no instance migrates between index sets back and forth.

Theorem 1 (Immediate cycling will not occur). If a breakpoint \((x_c, y_c)\) migrates from \(C\) to \(S\) or vice versa in round \(t\), it will not come back in the next round.

Proof. \(C \to S\): Suppose \((x_c, y_c)\) migrates from \(C\) to \(S\). Before it enters \(C\), we always have \(\alpha_c^{(t)} = 0\). If it comes back from \(C\) to \(C\) in the next round, its margin will increase, i.e., \(\Delta\gamma_c^{(t+1)} > 0\). Notice that \(\Delta\gamma_c = -\Delta\alpha_c / \lambda\), \(\forall i \in C\) according to Eqn. (5), which implies \(\Delta\alpha_c^{(t+1)} > 0\). Together with \(\alpha_c^{(t)} = 0\), we have \(\alpha_c^{(t+1)} < 0\). A contradiction occurs since \(\alpha_c > 0\), \(\forall i \in \mathcal{L}\).

The proof of migration from \(C\) to \(R\) is similar. If \((x_c, y_c)\) comes back from \(R\) to \(C\) in the next round, its margin will decrease, and we have \(\Delta\gamma_c^{(t+1)} < 0\) and \(\Delta\alpha_c^{(t+1)} > 0\) by noticing that \(\Delta\gamma_c = -\Delta\alpha_c / \lambda\), \(\forall i \in \mathcal{R}\) according to Eqn. (6).

Together with \(\alpha_c^{(t)} = 0\), we have \(\alpha_c^{(t+1)} > 0\), which contradicts with \(\alpha_c < 0\), \(\forall i \in \mathcal{R}\).

\(S \to C\): It is equivalent to prove that the margin varies monotonically, i.e., \(\Delta\gamma_c^{(t+1)}\) and \(\Delta\gamma_c^{(t)}\) have the same sign.

Before a breakpoint turns out in round \(t\), the index sets are \(S^{(t-1)}\) and \(C^{(t-1)}\). Obviously the migration of \((x_c, y_c)\) from \(S^{(t-1)}\) to \(C^{(t)}\) can be viewed as the reverse process of migration from \(C^{(t)}\) to \(S^{(t-1)}\). Thus, we can calculate \(t^{(t)}\) as

\[
t^{(t)} = -Q^{-1}_{S^{(t-1)}}H_{S^{(t-1)}}
\]

\[
= -\left(\begin{bmatrix} Q^{-1}_{S^{(t-1)}} & 0 \\ 0 & 0 \end{bmatrix} + \frac{\gamma}{\beta}S^{-1}\begin{bmatrix} v & 1 \end{bmatrix}\right)H_{S^{(t-1)}}k
\]

\[
= \begin{bmatrix} t^{(t+1)} \\ 0 \end{bmatrix} - \begin{bmatrix} v & 1 \end{bmatrix}S^{-1}\begin{bmatrix} v^T \end{bmatrix} + H_{ek}\right),
\]

where \(S\) is the Shur’s complement of \(Q_{S^{(t-1)}}\), and the first and last equality are according to the definition of \(t\), \(v\), and Eqn. (15), respectively. With Eqn. (16), we have

\[
\Delta\alpha_c^{(t)} = -S^{-1}\begin{bmatrix} v^T \end{bmatrix}H_{S^{(t)}}k + H_{ek}\Delta\alpha_k^{(t)}
\]

\[
= -S^{-1}\left(-H_{cS^{(t)}}Q^{-1}_{S^{(t)}}H_{S^{(t)}}k + H_{ek}\right)\Delta\alpha_k^{(t)}.
\]

Then we can further calculate the variation of margin for the instance \((x_c, y_c)\) in round \(t + 1\) as

\[
\Delta\gamma_c^{(t+1)} = \sum_{j \in S^{(t)}}\Delta\alpha_j^{(t+1)}h_{cj} + \Delta\alpha_k^{(t+1)}h_{ek}
\]

\[
= H_{cS^{(t)}}\Delta\alpha_k^{(t+1)} + H_{ek}\Delta\alpha_k^{(t+1)}
\]

\[
= \left(-H_{cS^{(t)}}Q^{-1}_{S^{(t)}}H_{S^{(t)}}k + H_{ek}\right)\Delta\alpha_k^{(t+1)},
\]

where the first and third equality are according to Eqn. (7) and Eqn. (9), respectively. With Eqn. (17), we have \(\Delta\gamma_c^{(t+1)} = -S\Delta\alpha_c^{(t)}\Delta\alpha_k^{(t+1)} / \Delta\alpha_k^{(t)}\).

Since the update of \(\alpha_k\) is monotonic, \(\Delta\alpha_k^{(t)}\) and \(\Delta\alpha_k^{(t+1)}\) have the same sign. Notice that \(\Delta\alpha_k^{(t)}\) has the opposite sign to \(\Delta\gamma_c^{(t)}\), thus \(\Delta\gamma_c^{(t+1)}\) has the same sign with \(\Delta\gamma_c^{(t)}\) if \(S > 0\). Since \(Q_S\) is a positive definite matrix, the corresponding Shur’s complement \(S > 0\) [Ouellette, 1981], which concludes the proof. For multiple instance changes, the proof is similar.

5 Experiments

In this section, we empirically analyze the effectiveness and efficiency of our proposed methods.

5.1 Setup

Data sets. We perform experiments on nine real-world data sets available on the LIBSVM website. The features of each data set are normalized into \([0, 1]\). The characteristics of data sets are summarized in Table 2. We randomly divide all data sets into training and test sets with a ratio of 4:1. Besides, we randomly select 75% of the training data as historical data and the rest are varying data.

1https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

3527
Methods. To make an overall comparison, we implement single incremental and decremental ODM (SID-ODM) and batch incremental and decremental ODM (BID-ODM). We use historical data to train the initial ODM and all varying instances will be added to the model and then deleted. SID-ODM adds/deletes an instance at once while BID-ODM processes a batch of data. The batch size is set to 5. Apart from our proposed methods, we retrain the model for comparison, which is denoted as SR-ODM and BR-ODM corresponding to single and multiple varying instances, respectively. All methods are performed on a machine with the 11th Gen Intel(R) Core(TM) i5-11320H@3.20GHz CPUs and 16GB main memory. ODM is optimized by dual coordinate gradient descent.

Hyperparameters. The hyperparameters in our experiments contain three model hyperparameters $\{\lambda, \theta, \sigma\}$. $\lambda$ and $\theta$ are tuned from $\{10^{-3}, \ldots, 10^{3}\}$ and $\{0.1, 0.2, \ldots, 0.9\}$, respectively by grid search. The kernel hyperparameter $\sigma$ is fixed to $1/d$ where $d$ is the number of features. All experiments are performed five times according to different data partitions.

Evaluation Measures. To verify the efficiency of our proposed methods, we compare the running time of each method. Besides, we also compare the accuracy on the test set to validate the effectiveness. We record the average accuracy as well as the standard deviation.
5.2 Results

Efficiency Comparison. Figure 3 demonstrates the cumulative running time of conducting SID-ODM, BID-ODM, SR-ODM, and BR-ODM on each data set. It can be seen that both SID-ODM and BID-ODM cost much less time than retraining the model from scratch, which reveals the efficiency of our proposed method. Besides, it is worth noting that the running time of SR-ODM and BR-ODM varies significantly with the data size while SID-ODM and BID-ODM behave steadily, which benefits from the Sherman-Morrison-Woodbury formula.

Table 3 further shows the speedup ratio of our proposed methods compared to retraining the model. Since data sets have various characteristics, the boost of the performance differs slightly, but on all data sets, we can gain more than 4 times acceleration and even can reach $9.1 \times$ speedup on average. Besides, there is no doubt that BID-ODM is more efficient than SID-ODM since it processes multiple instances at one time but with a closed-time cost to SID-ODM. The speedup ratio of BID-ODM compared to SID-ODM is also recorded in Table 3 to make an intuitive comparison.

![Diagram of efficiency comparison](image)

Table 3: The average running time of SR-ODM, BR-ODM, SID-ODM, and BID-ODM on each data set with the speedup ratio

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Time (in seconds)</th>
<th>Speedup Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SR-ODM</td>
<td>SID-ODM</td>
</tr>
<tr>
<td>heart</td>
<td>0.007±0.001</td>
<td>0.001±0.001</td>
</tr>
<tr>
<td>ionosphere</td>
<td>0.012±0.002</td>
<td>0.001±0.001</td>
</tr>
<tr>
<td>breast</td>
<td>0.033±0.005</td>
<td>0.004±0.003</td>
</tr>
<tr>
<td>diabetes</td>
<td>0.040±0.006</td>
<td>0.008±0.004</td>
</tr>
<tr>
<td>fourclass</td>
<td>0.047±0.007</td>
<td>0.009±0.004</td>
</tr>
<tr>
<td>splice</td>
<td>0.197±0.031</td>
<td>0.022±0.015</td>
</tr>
<tr>
<td>svmage3</td>
<td>0.127±0.020</td>
<td>0.024±0.012</td>
</tr>
<tr>
<td>madelon</td>
<td>3.001±0.470</td>
<td>0.078±0.079</td>
</tr>
<tr>
<td>a2a</td>
<td>0.443±0.071</td>
<td>0.082±0.054</td>
</tr>
</tbody>
</table>

Generalization Performance Comparison. Figure 4 shows the accuracy over 5 trials with a box plot. We also plot the mean accuracy of retraining the model in the box. Intuitively, our proposed methods accelerate the update of ODM nearly without sacrificing accuracy, which validates the effectiveness of our proposed methods.

In order to further validate the performance of ID-ODM, we compare it with MID-SVM [Karasuyama and Takeuchi, 2009], which is another batch IDL method based on SVM. We incrementally add 100 instances to the training set and record the accuracy. As shown in Table 4, ID-ODM performs better than MID-SVM on most datasets, which benefits from optimizing margin distribution.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>ID-ODM</th>
<th>MID-SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>heart</td>
<td>81.5 ± 4.7</td>
<td>81.1 ± 4.1</td>
</tr>
<tr>
<td>ionosphere</td>
<td>87.4 ± 6.6</td>
<td>90.6 ± 3.8</td>
</tr>
<tr>
<td>breast</td>
<td>97.2 ± 0.9</td>
<td>96.7 ± 1.2</td>
</tr>
<tr>
<td>diabetes</td>
<td>76.9 ± 4.8</td>
<td>76.6 ± 4.9</td>
</tr>
<tr>
<td>fourclass</td>
<td>78.7 ± 2.8</td>
<td>77.1 ± 1.9</td>
</tr>
<tr>
<td>splice</td>
<td>84.4 ± 3.1</td>
<td>85.5 ± 3.6</td>
</tr>
<tr>
<td>svmage3</td>
<td>80.5 ± 1.5</td>
<td>79.9 ± 1.3</td>
</tr>
<tr>
<td>madelon</td>
<td>58.0 ± 1.1</td>
<td>56.7 ± 1.6</td>
</tr>
<tr>
<td>a2a</td>
<td>83.0 ± 0.8</td>
<td>83.0 ± 0.8</td>
</tr>
</tbody>
</table>

Avg. Acc. | 81.1 | 80.8

Table 4: The accuracy of ID-ODM and MID-SVM

6 Conclusions

In this paper, we propose the ID-ODM to efficiently update ODM when new data is available or old data turns invalid. Specifically, we avoid updating the unbounded Lagrange multipliers in an infinite range by estimating the optimal value beforehand, thus iteratively updating the model. Moreover, ID-ODM is effective no matter how data varies, thus it is applicable to various real-world tasks. Besides, we provide some theoretical analysis on ID-ODM. Extensive experiments validate the effectiveness and efficiency of our method. In the future, we will extend our method to other models with quadratic-type loss.
Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0108501, the National Natural Science Foundation of China under Grant 62006088, and the Key R&D Program of Hubei under Grant 2020BAA020.

References


