Weighted Support Vector Machine for Extremely Imbalanced Data*

Jongmin Mun[a], Sungwan Bang[b,*], Jaeoh Kim[c,**]

[a]*Data Sciences and Operations Department, Marshall School of Business, University of Southern California,*
[b]*Department of Mathematics, Korea Military Academy,*
[c]*Department of Data Science, Inha University,*

Abstract

Based on an asymptotically optimal weighted support vector machine (SVM) that introduces label shift, a systematic procedure is derived for applying oversampling and weighted SVM to extremely imbalanced datasets with a cluster-structured positive class. This method formalizes three intuitions: (i) oversampling should reflect the structure of the positive class; (ii) weights should account for both the imbalance and oversampling ratios; (iii) synthetic samples should carry less weight than the original samples. The proposed method generates synthetic samples from the estimated positive class distribution using a Gaussian mixture model. To prevent overfitting to excessive synthetic samples, different misclassification penalties are assigned to the original positive class, synthetic positive class, and negative class. The proposed method is numerically validated through simulations and an analysis of Republic of Korea Army artillery training data.

Keywords: Bayes rule, Cost-sensitive learning, Gaussian mixture, Imbalanced classification, Label shift, Oversampling, Weighted support vector machine

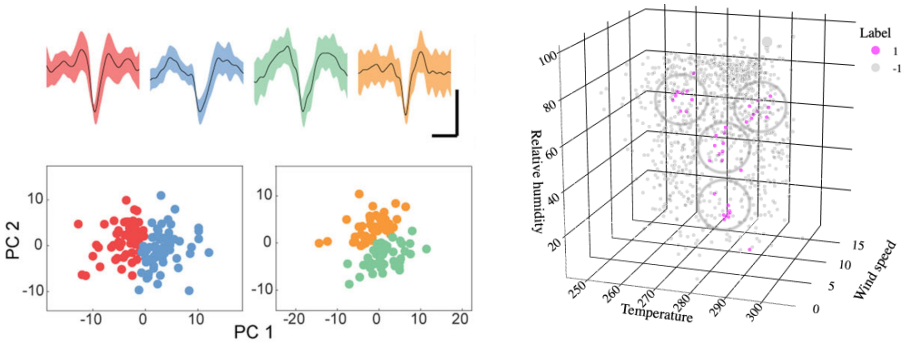
*This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2022-00165709).

*Co-corresponding author.
**Corresponding author.
Email addresses: jongmin.mun@marshall.usc.edu (Jongmin Mun), wan1365@gmail.com (Sungwan Bang), jaeoh.k@inha.ac.kr (Jaeoh Kim)

1. Introduction

Extremely imbalanced datasets present significant challenges in binary classification, as discussed in Krawczyk (2016). Formally, let $\mathcal{X} \subset \mathbb{R}^d$ represent the feature space and $\mathcal{Y} := \{-1, 1\}$ the label space. Let $\mathcal{P}$ be a family of joint distributions over $\mathcal{X} \times \mathcal{Y}$. For a random pair $(\boldsymbol{X}, Y)$ drawn from some $\mathbb{P} \in \mathcal{P}$, denote the marginal probabilities of the positive and negative classes as $\pi^+(\mathbb{P}) := \mathbb{P}(Y = 1)$ and $\pi^-(\mathbb{P}) := \mathbb{P}(Y = -1)$, respectively. Define the population imbalance ratio as $\mathrm{IR}(\mathbb{P}) := \pi^-(\mathbb{P})/\pi^+(\mathbb{P})$. An i.i.d. random sample $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$ drawn from $\mathbb{P}$ is considered extremely imbalanced if $\mathrm{IR}(\mathbb{P})$, or its empirical counterpart $\widehat{\mathrm{IR}}(\mathbb{P}) := (n - n^+)/n^+$, is very large, where $n^+ := \sum_{i=1}^n \mathbb{1}(Y_i = 1)$. Let c^- and c^+ be the decision-theoretic costs of false negatives and false positives, respectively. For extremely imbalanced datasets, it is usually the case that $c^- \gg c^+$. For example, failing to predict a wildfire has far more severe consequences than the cost of a false alarm. Conventional classification methods, however, tend to bias toward the negative class, leading to significant losses in high-stakes scenarios.

Datasets exhibiting subgroup structures have long posed challenges in statistics, dating back to Simpson's paradox (Simpson, 1951). Before defining



(a) Spike-sorted neural signals.

(b) Meteorological conditions during Army artillery training.

Figure 1: (a) Principal components of neural signals for two movement states (left and right subplots), each with two subgroups (corresponding to different neurons, colored differently). The upper plot shows the averaged signals per subgroup. (b) Army training sessions for wildfire (red, $Y = 1$) and no wildfire (grey, $Y = -1$). Axes represent meteorological conditions, with marker size indicating precipitation. Grey circles highlight subgroups of the positive class identified visually.

these structures, we illustrate two modern examples:

- **Neural signals.** Neural signals recorded via multi-electrode arrays are essential for brain research. Each electrode captures signals from multiple neurons, forming signal subgroups corresponding to individual neurons. Identifying these subgroups is a critical step in neural signal analysis (Chung et al., 2017; Chaure et al., 2018), as illustrated in Figure 1a, taken from Park et al. (2024).

- **Demographic subgroups.** Datasets often misrepresent the population distribution of gender, racial, or nationality demographics. For instance, genomic data overrepresents individuals of European descent (Fatumo et al., 2022). In response, fairness-aware machine learning aims to mitigate performance disparities across subgroups (Duchi and Namkoong, 2021; Chi et al., 2021), which involves identification of subgroup patterns within each class. For example, Fonseca (2013) employs hierarchical clustering methods and latent class models to detect subgroups in social science data.

We now formally define the subgroup structure of the positive class as follows.

Definition 1.1 (Subgroup structure)**.** *Let $g^+(\boldsymbol{x})$ denote the conditional density of the features of $\mathbb{P}$ given $Y = 1$. The positive class of $\mathbb{P}$ has a subgroup structure if $g^+(\boldsymbol{x}) = \sum_{k=1}^{K^*} w_k g_k(\boldsymbol{x})$, where for each k, $g_k(\boldsymbol{x})$ is any valid density over $\mathcal{X}$ and $0 < w_k < 1$ is a mixing weight such that $\sum_{k=1}^{K^*} w_k = 1$.*

This paper addresses binary classification in extremely imbalanced datasets with subgroup structures in the positive class. Figure 1b illustrates this with a wildfire dataset from Republic of Korea Army training sessions (Nam et al., 2024), where wildfire instances are far fewer than non-wildfire ones, and the meteorological variables reveal distinct subgroups within the wildfire cases. A detailed analysis is provided in Section 4.2.

Support vector machine (SVM; Cortes and Vapnik (1995)), despite its popularity in binary classification and proven asymptotic optimality (Lin, 2002), often performs poorly on extremely imbalanced datasets (Wu and Chang, 2003). Refinements typically focus on two approaches: cost-sensitive learning and oversampling. Cost-sensitive learning approaches (Veropoulos et al., 1999; Yang et al., 2005; Wang et al., 2014) assign higher misclassification penalties to positive class samples. In practice, the oversampling

approach is often preferred, which increases the positive class sample size by generating synthetic data. SMOTE (Chawla et al., 2002), a popular oversampling technique based on linear interpolation, has widely used variations such as ADASYN (He et al., 2008), Borderline-SMOTE (Han et al., 2005), and Safe-level-SMOTE (Bunkhumpornpat et al., 2009). However, when subgroup structures are present, these methods risk generating synthetic samples outside the density support. Additionally, a large volume of synthetic data can introduce significant noise, severely compromising classifier generalizability.

To address these limitations, we combine oversampling with cost-sensitive learning. Synthetic positive class samples are generated from the estimated positive class distribution using a flexible Gaussian mixture model (GMM), capturing the subgroup structure (Bang and Kim, 2020). To minimize noise from synthetic samples, we assign higher misclassification penalties to original positive samples than to synthetic ones. Additionally, following the weighted SVM approach (Veropoulos et al., 1999), we apply larger misclassification penalties to positive samples than to negative ones, resulting in three distinct penalties for negative, original positive, and synthetic positive samples.

Intuitively, the three misclassification costs should reflect both the original imbalance and oversampling ratios. We formalize this intuition by considering the oracle case where oversampling utilizes the true positive class density, specifying penalties that ensure the classifier's performance asymptotically approaches that of the optimal Bayes classifier. Building on this, we propose a combined oversampling and weighted SVM approach, using sample estimates of these oracle penalties. Simulation studies and real data analysis demonstrate that our method compares favorably against conventional SVM-based approaches for imbalanced data across various classification metrics.

Our study aims to provide a practical method for applying SVM to extremely imbalanced datasets with a cluster-structured positive class, guided by theory in the oracle setting. Our approach differs from previous studies that apply oversampling and/or weighted SVM in the following ways:

- In the presence of positive class subgroup structures, our method leverages them to generate more accurate synthetic samples.
- Our method prevents overfitting to noise from synthetic samples by assigning them lower misclassification penalties than original positive samples.

- Our method systematically determines the three misclassification penalties using a formula based on statistical decision theory.

The rest of the paper is organized as follows. Section 2 provides an overview of statistical decision theory, weighted SVM, and the GMM. Section 3 presents the asymptotically optimal oracle procedure and derives our proposed methodology from it. Section 4 numerically validates our method through simulations and a real data study. Finally, Section 5 provides the conclusion and discusses further considerations.

2. Background

This section briefly introduces statistical decision theory for binary classification. We relate this theory to weighted SVM, forming the foundation of our proposed method. We demonstrate the optimality of weighted SVM under label shift (defined in Section 2.2), which motivates the misclassification penalties in our method. Finally, we provide a concise introduction to the GMM and EM algorithm, both of which are used in our proposed method.

2.1. Statistical decision theory for binary classification

Let $g^-(\boldsymbol{x})$ denote the conditional density of the features of $\mathbb{P}$ given $Y = -1$. The posterior probability $p(\boldsymbol{x}) := \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$ is expressed as:

$$p(\boldsymbol{x}) = \frac{\pi^+(\mathbb{P})g^+(\boldsymbol{x})}{\pi^+(\mathbb{P})g^+(\boldsymbol{x}) + \pi^-(\mathbb{P})g^-(\boldsymbol{x})}, \tag{1}$$

by the Bayes formula. The classification risk of a classifier ϕ is defined as:

$$R_{\mathbb{P}}(\phi) := \mathbb{E}\left[c^-\mathbb{1}\left(\phi(\boldsymbol{X}) = -1\right)p(\boldsymbol{X}) + c^+\mathbb{1}\left(\phi(\boldsymbol{X}) = 1\right)\left(1 - p(\boldsymbol{X})\right)\right]. \tag{2}$$

The minimizer of (2), known as the Bayes rule, is written as:

$$\phi^*_{\mathbb{P}}(\boldsymbol{x}) = \begin{cases} +1 & \text{if } \dfrac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})} > \dfrac{c^+}{c^-}, \\ -1 & \text{otherwise,} \end{cases} \tag{3}$$

and its classification risk, known as the Bayes optimal risk, is denoted as $R^*_{\mathbb{P}} := R_{\mathbb{P}}(\phi^*_{\mathbb{P}})$. Since $p(\boldsymbol{x})$ is unknown, the Bayes rule (3) is also unknown, and we must learn a classifier ϕ_n from a random sample $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$. We

now introduce the notion of consistent classifiers (Lin, 2004), an asymptotic optimality concept based on the Bayes optimal risk, which is the best possible risk that a classifier can achieve.

Definition 2.1 (Consistent classifier)**.** *For $\mathbb{P} \in \mathcal{P}$, a sequence of classifiers ϕ_n based on $\{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n$ is consistent if its classification risk sequence*

$$R_{\mathbb{P}}(\phi_n) = \mathbb{E}\left[c^- \mathbb{1}\left(\phi_n(\boldsymbol{X}) = -1\right) p(\boldsymbol{X}) + c^+ \mathbb{1}\left(\phi_n(\boldsymbol{X}) = 1\right)\left(1 - p(\boldsymbol{X})\right) | \{(\boldsymbol{X}_i, Y_i)\}_{i=1}^n\right] \tag{4}$$

converges in probability to the Bayes optimal risk $R^*_{\mathbb{P}}$, as $n \to \infty$.

2.2. Weighted SVM and its optimality

It is computationally intractable to derive a classifier by minimizing the empirical version of the classification risk (4), because the (modified) 0-1 loss in (4) is not convex in ϕ_n. A common alternative is to use a convex surrogate for the 0-1 loss with appropriate regularization, which yields a consistent classifier in many settings (Bartlett et al., 2006). One way to implement this idea is through the reproducing kernel Hilbert space (RKHS) approach.

Let H_K be an RKHS of real-valued functions, equipped with the norm $\|\cdot\|_{H_K}$, induced by the Gaussian kernel $K(\cdot,\cdot)$ with bandwidth γ. With a random sample of size n, we train a classifier $\phi_n = \operatorname{sign}\big(f_n(\boldsymbol{x}) + b_n\big)$ by solving:

$$(f_n, b_n) = \underset{f \in H_K, b \in \mathbb{R}}{\arg\min}\, \Omega(\lambda_n, f) + \frac{1}{n}\sum_{i=1}^n \ell\big(y_i, f(\boldsymbol{x}_i) + b\big), \tag{5}$$

where Ω is a regularization function, $\lambda_n > 0$ is a regularization parameter, and ℓ is a convex loss function that bounds the 0-1 loss from above. We present two examples of (5) that form the foundation of our proposed method: kernel SVM (Cortes and Vapnik, 1995) and weighted kernel SVM (Veropoulos et al., 1999).

Example 2.1 (Kernel SVM)**.** *Let $\Omega(\lambda_n, \cdot) = \lambda_n \|\cdot\|^2_{H_K}$ and $\ell\big(y_i, f(\boldsymbol{x}_i) + b\big) = \big(1 - y_i\big(f(\boldsymbol{x}_i) + b\big)\big)_+$, where $(u)_+ = \max\{0, u\}$. Then the classifier based on (5) is the soft margin Gaussian kernel SVM (Hastie et al., 2009).*

The kernel SVM in Example 2.1 is consistent when $c^+ = c^- = 1$ (Lin, 2002). The weighted kernel SVM improves upon this by assigning different misclassification penalties to each class:

Example 2.2 (Weighted kernel SVM). *Let* $\Omega(\lambda_n, \cdot) = \lambda_n \|\cdot\|_{H_K}^2$ *and*

$$\ell\big(y_i, f(\boldsymbol{x}_i)\big) = L(y_i)\big(1 - y_i(f(\boldsymbol{x}_i) + b)\big)_+. \tag{6}$$

Then the classifier based on (5) *is the weighted Gaussian kernel SVM, with the misclassification penalty set to* $L(1)$ *for the positive class and* $L(-1)$ *for the negative class.*

Lin et al. (2002) prove the consistency of the weighted kernel SVM in Example 2.2 under label shift (Garg et al., 2020). Under label shift, the training sample is drawn from the *source distribution* $\mathbb{P}_s$, but the unlabeled data for evaluation or deployment is drawn from the *target distribution* $\mathbb{P}$, where $\mathbb{P}_s(Y_s = y) \neq \mathbb{P}(Y = y)$ but $\mathbb{P}_s(\boldsymbol{X}_s = \boldsymbol{x}|Y_s = y) = \mathbb{P}(\boldsymbol{X} = \boldsymbol{x}|Y = y)$. Since their result motivates our proposed method, we present it as a theorem.

Theorem 2.1 (Consistency of weighted kernel SVM). *Assume label shift with a target distribution* $\mathbb{P}$ *and a source distribution* $\mathbb{P}_s$. *Let* $\phi_n(\mathbb{P}_s)$ *be a sequence of weighted SVM classifiers trained with a random sample of size* n *drawn from* $\mathbb{P}_s$, *with the loss and regularization function defined as in Example 2.2, and the misclassification penalties in* (6) *specifically defined as:*

$$L(-1) = c^+\pi^+(\mathbb{P}_s)\pi^-(\mathbb{P}) \text{ and } L(1) = c^-\pi^-(\mathbb{P}_s)\pi^+(\mathbb{P}).$$

Then $\phi_n(\mathbb{P}_s)$ *is consistent with respect to* $\mathbb{P}$, *i.e.* $R_{\mathbb{P}}\big(\phi_n(\mathbb{P}_s)\big) \xrightarrow{P} R^*_{\mathbb{P}}$.

The proof of Theorem 2.1 rewrites the Bayes rule $\phi^*_{\mathbb{P}}$ (3) in terms of the posterior probability of $\mathbb{P}_s$, denoted as $p_s(\boldsymbol{x})$, instead of $p(\boldsymbol{x})$. We define $p_s(\boldsymbol{x})$ as follows:

$$p_s(\boldsymbol{x}) := \mathbb{P}_s(Y_s = 1|\boldsymbol{X}_s = \boldsymbol{x}) = \frac{\pi^+(\mathbb{P}_s)g^+(\boldsymbol{x})}{\pi^+(\mathbb{P}_s)g^+(\boldsymbol{x}) + \pi^-(\mathbb{P}_s)g^-(\boldsymbol{x})}. \tag{7}$$

Using (1) and (7), the Bayes rule (3) is rephrased as follows:

$$\phi_{\mathbb{P}^*}(\boldsymbol{x}) = \begin{cases} +1 & \text{if} \quad \dfrac{p_s(\boldsymbol{x})}{1 - p_s(\boldsymbol{x})} > \dfrac{c^+}{c^-}\dfrac{\pi^+(\mathbb{P}_s)}{\pi^-(\mathbb{P}_s)}\dfrac{\pi^-(\mathbb{P})}{\pi^+(\mathbb{P})} \\ -1 & \text{otherswise,} \end{cases}$$

which can be further rephrased as:

$$\phi^*(\boldsymbol{x}) = \text{sign}\left[p_s(\boldsymbol{x}) - \frac{L(-1)}{L(-1)+L(1)}\right]. \tag{8}$$

Now we briefly explain the GMM. The GMM assumes that each component density $f_k(\boldsymbol{x})$ in Definition 1.1 is a multivariate Gaussian density $\mathcal{N}(\cdot \mid \boldsymbol{\mu}_k, \Sigma_k)$, where $\boldsymbol{\mu}_k$'s are mean vectors and Σ_k's are covariance matrices. The EM algorithm (Dempster et al., 1977) estimates the parameters $\{(\boldsymbol{\mu}_k, \Sigma_k, w_k)\}_{k=1}^{K^*}$ by iteratively maximizing the log-likelihood $\mathcal{L}(\{\boldsymbol{\mu}_k, \Sigma_k, w_k\}_{k=1}^{K^*} \mid \boldsymbol{x}_1, \ldots \boldsymbol{x_n}) = \sum_{i=1}^{n} \log\big(\sum_{k=1}^{K^*} w_k \mathcal{N}(\boldsymbol{x}_i \mid \boldsymbol{\mu}_k, \Sigma_k)\big)$.

3. Proposed classification method

Starting our reasoning from the oracle case where we have full information about the data-generating distribution, we present an optimal oracle classifier that intentionally introduces label shift. Then we propose a practical method inspired by this oracle classifier.

3.1. Motivation if the underlying distribution were known

Assume an oracle setting with full knowledge of the positive class conditional density $g^+(\boldsymbol{x})$ and the population imbalance ratio $\text{IR}(\mathbb{P})$. Then we can intentionally introduce label shift by creating a source distribution that can be interpreted as generating the synthetic-data-augmented dataset.

Definition 3.1 (Oracle oversampler)**.** *Fix a target distribution $\mathbb{P}$, with positive and negative class densities $g^+(\boldsymbol{x})$ and $g^-(\boldsymbol{x})$, respectively. Choose a desired imbalance ratio $\text{IR}^\dagger > 0$. We define a source distribution $\mathbb{P}_{\text{IR}^\dagger}$ and the corresponding random pair $(\boldsymbol{X}_{\text{IR}^\dagger}, Y_{\text{IR}^\dagger}, Z)$ over $\mathcal{X} \times \mathcal{Y} \times \{0,1\}$, with the following properties:*

1. *Conditional densities of features $\boldsymbol{X}_{\text{IR}^\dagger}$ given $Y_{\text{IR}^\dagger} = 1$ and $Y_{\text{IR}^\dagger} = -1$ are the same as $g^+(\boldsymbol{x})$ and $g^-(\boldsymbol{x})$, respectively.*
2. $\text{IR}(\mathbb{P}_{\text{IR}^\dagger}) = \pi^-(\mathbb{P}_{\text{IR}^\dagger})/\pi^+(\mathbb{P}_{\text{IR}^\dagger}) = \text{IR}^\dagger$.
3. $\mathbb{P}_{\text{IR}^\dagger}(Z = 1 | Y_{\text{IR}^\dagger} = -1) = 0$.
4. $\dfrac{\pi^-(\mathbb{P}_{\text{IR}^\dagger})}{\mathbb{P}_{\text{IR}^\dagger}(Y_{\text{IR}^\dagger} = 1, Z = 0)} = \text{IR}(\mathbb{P})$.

5. *Z and $\boldsymbol{X}_{\mathrm{IR}^\dagger}$ are conditionally independent, i.e. for any $\boldsymbol{x} \in \mathcal{X}$,*

$$\mathbb{P}_{\mathrm{IR}^\dagger}(Z = z | Y_{\mathrm{IR}^\dagger} = 1, \boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) = \mathbb{P}_{\mathrm{IR}^\dagger}(Z = z | Y_{\mathrm{IR}^\dagger} = 1), \text{ and}$$
$$\mathbb{P}_{\mathrm{IR}^\dagger}(Z = z | Y_{\mathrm{IR}^\dagger} = -1, \boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) = \mathbb{P}_{\mathrm{IR}^\dagger}(Z = z | Y_{\mathrm{IR}^\dagger} = -1).$$

The properties of Definition 3.1 imlply that a random sample of inflated size from drawn $\mathbb{P}_{\mathrm{IR}^\dagger}$ can be interpreted as the original dataset augmented with synthetic data from oracle oversampling, which has full access to the original conditional density (property 1) and imbalance ratio (property 2). Additionally, no synthetic negative samples are generated (property 3), and while each sample is randomly labeled as synthetic ($Z = 1$) or original ($Z = 0$) (property 5), the imbalance ratio within the original samples is preserved (property 4). We now present an optimal weighted SVM under label shift from $\mathbb{P}$ to $\mathbb{P}_{\mathrm{IR}^\dagger}$.

Definition 3.2 (Oracle oversampling weighted SVM)**.** *For a target distribution $\mathbb{P}$ and a source distribution $\mathbb{P}_{\mathrm{IR}^\dagger}$, define the oversampling ratio $\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) := \mathrm{IR}(\mathbb{P})/\mathrm{IR}^\dagger$. Extend the L function (misclassification penalties) in Theorem 2.1 into $L_{\mathrm{IR}^\dagger}(-1, 0) = L(-1)$ and*

$$L_{\mathrm{IR}^\dagger}(1, 0) = \frac{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})\, L(1)}{2} \text{ and } L_{\mathrm{IR}^\dagger}(1, 1) = \frac{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})\, L(1)}{2(\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1)}, \tag{9}$$

Then we define an extended loss function:

$$\ell_{\mathrm{IR}^\dagger}\big(y_i, z_i, f(\boldsymbol{x}_i)\big) := L_{\mathrm{IR}^\dagger}(y_i, z_i)\big(1 - y_i\big(f(\boldsymbol{x}_i) + b\big)\big)_+.$$

For given inflated sample size n, observe $\{\tilde{\boldsymbol{x}}_i, \tilde{y}_i, \tilde{z}_i\}_{i=1}^n$ from i.i.d. copies of $(\boldsymbol{X}_{\mathrm{IR}^\dagger}, Y_{\mathrm{IR}^\dagger}, Z)$. The oracle oversampling weighted SVM $\phi_{\mathrm{IR}^\dagger,n}(\boldsymbol{x}) := \mathrm{sign}\big(f_n(\boldsymbol{x}) + b_n\big)$ is a weighted SVM in Example 2.2 trained using this sample, i.e.

$$(f_n, b_n) = \underset{f \in H_K, b \in \mathbb{R}}{\arg\min}\ \lambda_n \|f\|_{H_K}^2 + \frac{1}{n}\sum_{i=1}^n L_{\mathrm{IR}^\dagger}(\tilde{y}_i, \tilde{z}_i)\big(1 - \tilde{y}_i\big(f(\tilde{\boldsymbol{x}}_i) + b\big)\big)_+. \tag{10}$$

Note that $\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})$ can be interpreted as the ratio of the number of positive

class samples after oversampling and before oversampling, since

$$\begin{aligned}\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) = \frac{\mathrm{IR}(\mathbb{P})}{\mathrm{IR}^\dagger} &= \frac{\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})/\mathbb{P}_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}=1, Z=0)}{\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})/\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})} \\ &= \frac{\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})}{\mathbb{P}_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}=1, Z=0)}. \end{aligned} \tag{11}$$

Additionally, the function $L_{\mathrm{IR}^\dagger}$ ensures that

$$L_{\mathrm{IR}^\dagger}(1,0)/L_{\mathrm{IR}^\dagger}(1,1) = \big(\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1\big),$$

which indicates that the original positive sample has $(\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})-1)$ times the relative importance of the synthetic positive sample. The following theorem, with proof in Appendix A, shows the asymptotic optimality of $\phi_{\mathrm{IR}^\dagger,n}$.

Theorem 3.1 (Optimality of oracle oversampling weighted SVM)**.** *The sequence of classifiers $\phi_{\mathrm{IR}^\dagger,n}$ is consistent with respect to $\mathbb{P}$, i.e.*

$$R_{\mathbb{P}}\big(\phi_{\mathrm{IR}^\dagger,n}\big) \xrightarrow{P} R^*_{\mathbb{P}}.$$

3.2. Proposed classification methodology: GSWSVM

The oracle procedure in Definition 3.2 assumes knowledge of both $\mathrm{IR}(\mathbb{P})$ and $g^+(\boldsymbol{x})$. In practice, while we can choose $\mathrm{IR}^\dagger$, we only observe $(\boldsymbol{x}_i, y_i)_{i=1}^n$ and do not have access to $\mathrm{IR}(\mathbb{P})$ or $g^+(\boldsymbol{x})$. We now introduce our method, the **G**aussian mixture **s**ynthetic samples **w**eighted **s**upport **v**ector **m**achine (GSWSVM), which replaces the parameters in Definition 3.2 with their corresponding sample versions. Using the empirical imbalance ratio $\widehat{\mathrm{IR}}(\mathbb{P}) = (n-n^+)/n^+$, the sample version oversampling ratio is defined as:

$$\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) := \frac{\widehat{\mathrm{IR}}(\mathbb{P})}{\mathrm{IR}^\dagger} = \frac{n-n^+}{\mathrm{IR}^\dagger n^+}. \tag{12}$$

Then the required number of synthetic positive class samples is defined as:

$$n_s^+ := \Big\lfloor (\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1)n^+ \Big\rfloor = \left\lfloor \left(\frac{n-n^+}{\mathrm{IR}^\dagger n^+} - 1\right) n^+ \right\rfloor = \left\lfloor \frac{n-n^+}{\mathrm{IR}^\dagger} - n^+ \right\rfloor. \tag{13}$$

Then the sample version $L_{\mathrm{IR}^\dagger}$ function is defined as:

$$\begin{aligned}
\widehat{L}_{\mathrm{IR}^\dagger}(-1,0) &:= c^+ \frac{n^+ + n_s^+}{n+n_s^+} \frac{n-n^+}{n}, \\
\widehat{L}_{\mathrm{IR}^\dagger}(1,0) &:= \frac{\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})}{2} c^- \frac{n-n^+}{n+n_s^+} \frac{n^+}{n}, \text{ and} \\
\widehat{L}_{\mathrm{IR}^\dagger}(1,1) &:= \frac{\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})}{2\big(\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})-1\big)} c^- \frac{n-n^+}{n+n_s^+} \frac{n^+}{n}.
\end{aligned} \tag{14}$$

Our proposed method is a two-step procedure:

1. Let $\hat{g}^+(\boldsymbol{x})$ be the estimated positive class density through Gaussian mixture estimation via EM algorithm. Generate n_s^+ synthetic samples from $\hat{g}^+(\boldsymbol{x})$ with indices $i = n+1, \ldots, n+n_s^+$. Set $z_i = 1$ for $i = n+1, \ldots, n+n_s^+$ and $z_i = 0$ for $i = 1, \ldots, n$. Note that no synthetic negative class samples are generated, and $g^-(\boldsymbol{x})$ is not estimated.

2. Apply the weighted SVM in Definition 3.2 to the augmented dataset $\{(\boldsymbol{x}_i, y_i, z_i)\}_{i=1}^{n+n_s^+}$, using the penalties $\widehat{L}_{\mathrm{IR}^\dagger}(-1,0)$, $\widehat{L}_{\mathrm{IR}^\dagger}(1,0)$, and $\widehat{L}_{\mathrm{IR}^\dagger}(1,1)$ in (14). The classifier is then fitted as: $\hat{\phi}_{\mathrm{IR}^\dagger,n}(\boldsymbol{x}) = \mathrm{sign}\big(f_n(\boldsymbol{x}) + b_n\big)$, where

$$(f_n, b_n) = \underset{f \in H_K, b \in \mathbb{R}}{\arg\min}\ \lambda_n \|f\|_{H_K}^2 + \frac{1}{n+n_s^+} \sum_{i=1}^{n+n_s^+} \widehat{L}_{\mathrm{IR}^\dagger}(y_i, z_i)\big(1 - y_i\big(f(\boldsymbol{x}_i)+b\big)\big)_+. \tag{15}$$

Note that the Lagrangian dual problem of (15) is as follows:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha} \in \mathbb{R}^{n+n_s^+}} \sum_{i=1}^{n+n_s^+} \alpha_i - \sum_{i=1}^{n+n_s^+} \sum_{j=1}^{n+n_s^+} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\
&\text{subject to} \quad \sum_{i=1}^{n+n_s^+} \alpha_i y_i = 0, \quad 0 \le \alpha_i \le \widehat{L}_{\mathrm{IR}^\dagger}(y_i, z_i)/\lambda_n.
\end{aligned}$$

In practice, this dual problem can be solved using the off-the-shelf packages for instance-wise weighted SVM, such as the `R` package `WeightSVM`. The misclassification penalties (14) account for the decision-theoretic costs c^+ and c^-,

the empirical imbalance ratio $\widehat{\mathrm{IR}}(\mathbb{P})$, and the oversampling ratio $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})$. For clarity and implementation, Algorithm 1 provides a summary of the method. Theorem 3.2 justifies the penalties in our method, demonstrating that the sample versions converge to the $L_{\mathrm{IR}^\dagger}$ function (9).

Algorithm 1 Classification method for imbalanced dataset with positive class subgroup structure: GSWSVM

Input: Observed dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$; Cost of false negative c^-; Cost of false positive c^+; Desired imbalance ratio $\mathrm{IR}^\dagger$; SVM tuning hyperparameter λ; Gaussian kernel hyperparameter γ

Output: classification rule $\hat{\phi}_{\mathrm{IR}^\dagger,n}(\boldsymbol{x})$

$\{(\boldsymbol{x}_i, y_i, 0)\}_{i=1}^n \leftarrow$ observed dataset with additional label $z_i = 0$

$n^+ \leftarrow \sum_{i=1}^n \mathbb{1}(Y_i = 1)$

$\widehat{\mathrm{IR}}(\mathbb{P}) \leftarrow (n - n^+)/n^+$

$\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) \leftarrow \widehat{\mathrm{IR}}(\mathbb{P})/\mathrm{IR}^\dagger$

$n_s^+ \leftarrow \left\lfloor (\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1)n^+ \right\rfloor$

$\hat{g}^+ \leftarrow$ an estimated positive class density learned by GMM via EM algorithm on the positive class samples of the training dataset

$\{(\boldsymbol{x}_i, y_i, 1)\}_{i=n+1}^{n+n_s^+} \leftarrow$ synthetic positive class samples generated from $\hat{g}^+$

$L_{\mathrm{IR}^\dagger}(-1, 0) \leftarrow c^+ \frac{n^+ + n_s^+}{n + n_s^+} \frac{1 - n^+}{n}$

$L_{\mathrm{IR}^\dagger}(1, 0) \leftarrow \frac{\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})}{2} c^- \frac{n - n^+}{n + n_s^+} \frac{n^+}{n}$

$L_{\mathrm{IR}^\dagger}(1, 1) \leftarrow \frac{\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})}{2(\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1)} c^- \frac{n - n^+}{n + n_s^+} \frac{n^+}{n}$

$\hat{\phi}_{\mathrm{IR}^\dagger,n}(\boldsymbol{x}) \leftarrow$ Solution of the weighted SVM with weights $L_{\mathrm{IR}^\dagger}(-1, 0)$, $L_{\mathrm{IR}^\dagger}(1, 0)$, and $L_{\mathrm{IR}^\dagger}(1, 1)$ for negative class, original positive class, and synthetic positive class samples, respectively, with Gaussian kernel bandwidth γ and SVM cost parameter $1/\lambda$

return $\hat{\phi}_{\mathrm{IR}^\dagger,n}(\boldsymbol{x})$

Theorem 3.2. *Consider a large sample regime where $n \to \infty$ observations are drawn from $\mathbb{P}$. In this regime, $n^+ = \sum_{i=1}^n \mathbb{1}(Y_i = 1)$ and its transformation n_s^+ (13) are sequences of random variables indexed by n. Then the following sequences of random variables, which are transformations of n^+*

and n_s^+*, have the following convergence in probability:*

$$\frac{n^+}{n} \xrightarrow{P} \pi^+(\mathbb{P}),\ \frac{n-n^+}{n} \xrightarrow{P} \pi^-(\mathbb{P}),\ \widehat{\mathrm{IR}}(\mathbb{P}) \xrightarrow{P} \mathrm{IR}(\mathbb{P}),$$
$$\frac{n-n^+}{n+n_s^+} \xrightarrow{P} \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger}),\ \frac{n^+ + n_s^+}{n+n_s^+} \xrightarrow{P} \pi^+(\mathbb{P}_{\mathrm{IR}^\dagger}),\ \textit{and}\ \widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) \xrightarrow{P} \mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}),$$

which imply:

$$\widehat{L}_{\mathrm{IR}^\dagger}(-1,0) \xrightarrow{P} L_{\mathrm{IR}^\dagger}(-1,0),\ \widehat{L}_{\mathrm{IR}^\dagger}(1,0) \xrightarrow{P} L_{\mathrm{IR}^\dagger}(1,0),\ \textit{and}\ \widehat{L}_{\mathrm{IR}^\dagger}(1,1) \xrightarrow{P} L_{\mathrm{IR}^\dagger}(1,1).$$

The proof of Theorem 3.2 is given in Appendix B. Suppose that $g^+(\boldsymbol{x})$ is really a Gaussian mixture. Under suitable conditions for the convergence of the EM algorithm to a global optimum (Balakrishnan et al., 2017; Kwon and Caramanis, 2020; Segol and Nadler, 2021), as $n \to \infty$, the estimated positive class density $\hat{g}^+(\boldsymbol{x})$ converges to the true density $g^+(\boldsymbol{x})$. This result, along with Theorem 3.2, implies that asymptotically, the augmented dataset after oversampling can be viewed as drawn from $\mathbb{P}_{\mathrm{IR}^\dagger}$, making our method solve the same optimization problem as the optimal procedure in Theorem 3.1. In other words, the classifier learned by our method approaches the Bayes rule asymptotically, providing an intuitive motivation for our method. Even if $g^+(\boldsymbol{x})$ is not really a Gaussian mixture, this argument still provides some intuition because Gaussian mixture is an universal approximater that can theoretically approximate all smooth densities, if sufficiently many components are used (Goodfellow et al., 2016).

In practical finite-sample regime, on the other hand, despite positive instances in a training dataset may represent an extremely small sample size, it remains possible to construct certain types of density functions, as exemplified by the artillery training data (Section 4.2) that inspired this study, and the simulations in Section 4.1.2. This underscores that an appropriate Gaussian mixture density function can indeed be learned from minority instances within the original dataset.

Finally, it is worth pointing out that validating subgroup structures within the positive class through exploratory data analysis is crucial before applying our method. We suggest beginning with visual inspections using PCA or dendrograms from hierarchical clustering. If they hint a subgroup structure, we estimate the number of clusters, K^*, to determine whether the positive class comprises subgroups ($K^* > 1$) or not ($K^* = 1$). We recommend Gap statis-

tic (Tibshirani et al., 2001) and Jump statistic (Sugar and James, 2003), as they effectively capture the absence of subgroups, unlike many conventional methods (Hastie et al., 2009, Chapter 14.3.11). If these methods indicate that $K^* > 1$ and the estimated K^* aligns with our visual assessments, we proceed with our approach GSWSVM.

4. Numerical Results

This section numerically validates the performance of the proposed method on extremely imbalanced synthetic and real datasets, comparing it to SVM-based baseline methods: SVM, ClusterSVM, SMOTE-SVM, BLSMOTE-SVM, and DBSMOTE-SVM (details in Appendix C). Our real dataset consists of military artillery training sessions of the Republic of Korea Army (ROKA), with labels indicating wildfire occurrences and features representing meteorological conditions. We also assess the robustness of our method to data dimensionality and oversampling ratios.

The hyperparameters for GSWSVM are set as follows. The oversampling ratio is fixed at $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) = 5$ in synthetic data analysis, based on our observation in Section 4.1.1 that the method is not sensitive to this parameter. In real data analysis, where the imbalance ratio is fixed, we try different oversampling ratios and again confirm the insensitivity of GSWSVM to these variations. Misclassification penalties are set as $c^+ = 1$ and $c^- = \mathrm{IR}(\mathbb{P})$ ($c^- = \widehat{\mathrm{IR}}(\mathbb{P})$ for real data), reflecting the relative importance of detecting the positive class. For the Gaussian mixture estimation, we apply the EM algorithm, using a spherical covariance structure with equal volume for efficiency in small sample size settings. The number of components is selected via Bayesian information criteria (Schwarz, 1978). For the oversampling-based baseline methods (SMOTE-SVM, BLSMOTE-SVM, and DBSMOTE-SVM), the oversampling ratio is tuned from $\{2, 4, 6\}$ in synthetic data analysis, and is pre-specified from $\{1.57, 2.36, 4.72, 11.55, 23.6\}$ in real data analysis. These methods all use the Gaussian kernel $K(\boldsymbol{u}, \boldsymbol{v}) = \exp(-\gamma\|\boldsymbol{u} - \boldsymbol{v}\|_2^2)$, and tune the SVM regularization parameter C (inverse of λ_n in (15)) and Gaussian kernel bandwidth γ, both from the grid $\{4^{(i-5)}\}_{i=0}^{10}$. For ClusterSVM, the regularization and global regularization parameters are both tuned from $\{1, 5, 10, 20, 50, 100\}$, following Gu and Han (2013). All tuning parameters are selected via 5-fold cross-validation, using G_{mean} as the evaluation metric.

Classification performance is often evaluated using metrics derived from the confusion matrix (Table 1), with accuracy being a common metric: $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$.

In extremely imbalanced datasets, prioritizing true positives is crucial, often by placing more weight on the true positive rate (TPR, or recall) $\frac{\text{TP}}{\text{TP+FN}}$ over the true negative rate (TNR, or specificity) $\frac{\text{TN}}{\text{TN+FP}}$. Specialized metrics that focus on true positives include the geometric mean $G_{\text{mean}} = \frac{\text{TP}\cdot\text{TN}}{(\text{TP+FN})(\text{TN+FP})}$ and the F_β score: $\frac{(1+\beta^2)\text{TP}}{(1+\beta^2)\text{TP}+\beta^2\text{TN}+\text{FP}}$. This paper uses $\beta = 2$ to place more weight on true positives. In conclusion, this paper reports accuracy, TPR, TNR, G_{mean}, and F_2 scores from 5-fold cross-validation.

Table 1: Confusion matrix for two class classification.

Predicted Label	True label	
	1	-1
1	True Positive (TP)	False Positive (FP)
-1	False Negative (FN)	True Negative (TN)

All metrics are reported as averages (with standard errors) over 200 Monte Carlo simulations. All numerical studies were conducted using `R` version 4.2.3 on a 2.50 GHz Intel Xeon E5-2640 processor with 64 GB RAM. The following `R` libraries were used: `e1071` (Meyer et al., 2024) for standard SVM, `WeightSVM` (Xu et al., 2011) for misclassification penalties, `smotefamily` (Siriseriwan, 2024) for SMOTE and variants, `SwarmSVM` (He, 2016) for clusterSVM, `mclust` (Scrucca et al., 2023) for Gaussian mixture models, and `caret` (Kuhn, 2008) for cross-validation.

4.1. Synthetic data analysis

We validate our method on synthetic multivariate Gaussian data, comparing it to baselines across two distributional settings with varying imbalance ratios, and examine its robustness to oversampling and sensitivity to data dimensions. The data and codes that support the findings of this study are openly available in GitHub at `https://github.com/Jong-Min-Moon/Gaussian_Mixture_Syntheticsamples_Weighted_Support_Vector_Machine.git`.

4.1.1. Comparison with baseline methods

We employ two modified versions of the multivariate Gaussian mixture data taken from Hastie and Tibshirani (1996), which proposes a method that is similar to ours—a combination of Fisher's linear discriminant analysis and

Gaussian mixture. The features are 21-dimensional, which is suitable to demonstrate the performance in moderate dimensions.

First, we define the basic form of the j-th entry of the mean vector using the following function:

$$h(j) = \max\big\{0.6 - 0.3|j - 11|, 0\big\}, \tag{16}$$

where $j = 1, \ldots, 21$. Define a random vector $\boldsymbol{X}_s \in \mathbb{R}^{21}$ as follows:

$$\boldsymbol{X}_s = \big(\boldsymbol{X}_s(1), \boldsymbol{X}_s(2), \ldots, \boldsymbol{X}_s(21)\big)^\top, \text{ where } \boldsymbol{X}_s(j) = h(j - s) + 0.2\epsilon_j, \tag{17}$$

where ϵ_j are i.i.d. univariate standard Gaussian. Note that $\boldsymbol{X}_s$ has a multivariate Gaussian distribution with an isotropic covariance matrix.

We consider two settings. In Setting 1, the positive and negative classes have the same level of variation, potentially simplifying the classification problem. In Setting 2, only the positive class has subgroup structure, while the negative class does not. More formally, the settings are stated as follows:

- Setting 1: The positive class sample and negative class sample are i.i.d. copies of the mixture of the following components, with uniform mixing probabilities:

$$\begin{aligned} &\text{Positive class: } \boldsymbol{X}_{-8}, \boldsymbol{X}_{-5}, \boldsymbol{X}_{-2}, \boldsymbol{X}_1, \boldsymbol{X}_4, \boldsymbol{X}_7, \boldsymbol{X}_{10} \\ &\text{Negative class: } \boldsymbol{X}_{-10}, \boldsymbol{X}_{-7}, \boldsymbol{X}_{-4}, \boldsymbol{X}_{-1}, \boldsymbol{X}_2, \boldsymbol{X}_5, \boldsymbol{X}_8, \end{aligned}$$

- Setting 2: The positive class is the same as Setting 1. The negative class sample follows a 21-dimensional multivariate Gaussian distribution, where the j-th component of the mean vector is given by $\frac{1}{7}\sum_{s\in S} h(j - s)$, with $S = \{-10, -7, -4, -1, 2, 5, 8\}$ and h defined in (16). The covariance matrix is $0.2I_{21}$, where I_{21} denotes the identity matrix of size 21×21.

Figure 2 visualizes the subgroup structure for the case of $\mathrm{IR}(\mathbb{P}) = 5.67$ by dimension reduction through t-Distributed Stochastic Neighbor Embedding (t-SNE; Maaten and Hinton (2008)) technique. In Setting 1, both the positive and negative classes contain subgroups, a common scenario seen in demographic and neuron subgroups (Section 1). This setting is frequently used to evaluate the performance of nonlinear kernel methods and generative

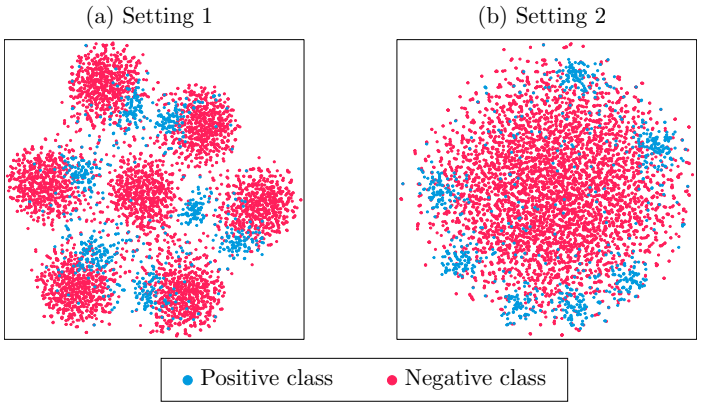


Figure 2: Scatterplot of the simulated dataset from Section 4.1.1 (15% positive class, $\mathrm{IR}(\mathbb{P}) = 5.67$), with dimensionality reduced via t-SNE, for (a) Setting 1 and (b) Setting 2. Note that while t-SNE captures the subgroup structure, it may not accurately reflect distances between subgroups.

models (Bang and Jhun, 2014; Mangasarian and Musicant, 2001; Bao et al., 2021).

For both of Setting 1 and Setting 2, we consider four simulated scenarios where the original imbalance ratios are $\mathrm{IR}(\mathbb{P}) = 49, 19, 9$, and 5.67, corresponding to proportion of positive class samples 2%, 5%, 10%, and 15%, respectively. The total sample size is fixed at $n = 3000$. The results for Setting 1, summarized in Table 2, demonstrate that our proposed method outperforms the baseline methods in terms of TPR, G_{mean}, and F_2. This superiority becomes more pronounced as $\mathrm{IR}(\mathbb{P})$ increases. The baseline methods achieve higher accuracy and TNR but lower TPR, resulting in reduced G_{mean} and F_2, indicating poor detection of the positive class. The results for Setting 2, summarized in Table 3, show a similar pattern. While the assumptions of Theorem 3.1 and Theorem 3.2 do not hold for finite samples ($n = 3000$), they offer insight into why GSWSVM outperforms baseline methods on this dataset.

4.1.2. Sensitivity to oversampling ratio and data dimension

We numerically show that our method is insensitive to the oversampling ratio $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})$ when it exceeds a moderate level. This finding eliminates the need for additional hyperparameter tuning, providing a significant computational advantage, and justifies our decision to fix the $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger})$ at 5 in Section 4.1.1. We also numerically demonstrate that the EM algorithm re-

Table 2: The means and standard deviations (in parentheses) of the assessment metrics for each method and proportion of positive class samples, evaluated on the imbalanced isotropic Gaussian mixture data from Setting 1 of Section 4.1, with mean vectors defined by the h function (16).

IR($\mathbb{P}$)	Method	Accuracy	TPR	TNR	G_{mean}	F_2
49	GSW$_{\text{SVM}}$	0.8630(0.0249)	0.6120(0.0608)	0.8681(0.0255)	0.7224(0.0386)	0.2854(0.0372)
	SVM	0.9748(0.0029)	0.2145(0.0846)	0.9903(0.0035)	0.4079(0.1337)	0.2627(0.0608)
	ClusterSVM	0.9663(0.0044)	0.3227(0.0636)	0.9795(0.0040)	0.5438(0.0634)	0.3044(0.0597)
	SMOTE-SVM	0.9734(0.0032)	0.2395(0.0651)	0.9884(0.0027)	0.4563(0.0864)	0.2556(0.0652)
	BLSMOTE-SVM	0.9738(0.0028)	0.2255(0.0756)	0.9891(0.0028)	0.4310(0.1071)	0.2455(0.0621)
	DBSMOTE-SVM	0.9724(0.0031)	0.2277(0.0610)	0.9876(0.0026)	0.4408(0.0790)	0.2414(0.0621)
19	GSW$_{\text{SVM}}$	0.8993(0.0141)	0.7481(0.0327)	0.9073(0.0146)	0.8223(0.0201)	0.5830(0.0354)
	SVM	0.9519(0.0040)	0.3550(0.1392)	0.9833(0.0066)	0.5280(0.2013)	0.4532(0.0372)
	ClusterSVM	0.9450(0.0052)	0.4686(0.0519)	0.9701(0.0049)	0.6698(0.0383)	0.4666(0.0470)
	SMOTE-SVM	0.9434(0.0056)	0.4963(0.0416)	0.9669(0.0049)	0.6885(0.0331)	0.4859(0.0391)
	BLSMOTE-SVM	0.9462(0.0043)	0.4696(0.0513)	0.9713(0.0040)	0.6658(0.0541)	0.4432(0.0415)
	DBSMOTE-SVM	0.9460(0.0047)	0.4270(0.0494)	0.9733(0.0038)	0.6359(0.0492)	0.4396(0.0422)
9	GSW$_{\text{SVM}}$	0.9062(0.0084)	0.8272(0.0219)	0.9150(0.0086)	0.8695(0.0130)	0.7404(0.0203)
	SVM	0.9266(0.0065)	0.5640(0.0787)	0.9669(0.0064)	0.7257(0.0959)	0.5953(0.0276)
	ClusterSVM	0.9269(0.0059)	0.6010(0.0364)	0.9631(0.0050)	0.7592(0.0237)	0.6099(0.0331)
	SMOTE-SVM	0.9060(0.0079)	0.6956(0.0281)	0.9294(0.0083)	0.8032(0.0166)	0.6532(0.0244)
	BLSMOTE-SVM	0.9060(0.0080)	0.6957(0.0276)	0.9294(0.0079)	0.8033(0.0168)	0.6105(0.0279)
	DBSMOTE-SVM	0.9182(0.0068)	0.5999(0.0265)	0.9535(0.0062)	0.7552(0.0175)	0.5984(0.0258)
5.67	GSW$_{\text{SVM}}$	0.8993(0.0073)	0.8493(0.0515)	0.9081(0.0098)	0.8741(0.0504)	0.7959(0.0127)
	SVM	0.9091(0.0103)	0.6330(0.0859)	0.9579(0.0086)	0.7616(0.1001)	0.6713(0.0203)
	ClusterSVM	0.9193(0.0057)	0.6910(0.0259)	0.9573(0.0043)	0.8127(0.0157)	0.6987(0.0234)
	SMOTE-SVM	0.8847(0.0111)	0.7829(0.0222)	0.9017(0.0124)	0.8397(0.0132)	0.7292(0.0203)
	BLSMOTE-SVM	0.8857(0.0117)	0.7806(0.0229)	0.9032(0.0134)	0.8392(0.0133)	0.7012(0.0264)
	DBSMOTE-SVM	0.9028(0.0091)	0.6920(0.0257)	0.9380(0.0086)	0.8050(0.0165)	0.6836(0.0248)

liably approximates Gaussian mixtures in moderate dimensions, confirming the usability of our method in these settings.

Robustness to oversampling ratio. Under Setting 1 of Section 4.1.1 with IR($\mathbb{P}$) = 19, we test oversampling ratios of $\widehat{\text{OR}}(\mathbb{P}_{\text{IR}^\dagger}) = 2, 3, 4, 5, 6, 7, 8$, and 9. Table 4 shows that for $\widehat{\text{OR}}(\mathbb{P}_{\text{IR}^\dagger}) \geq 5$, all assessment metrics show no significant improvement.

Reliable Gaussian mixture distribution estimation in moderate dimensions. As our method targets extremely imbalanced datasets, the performance of Gaussian mixture estimation on small positive class samples could be a concern. To address this, we conduct a numerical investigation.

We begin by explaining the role of Gaussian mixture estimation in our method. There are two main approaches to Gaussian mixture distribution

Table 3: The means and standard deviations (in parentheses) of the assessment metrics for each method and proportion of positive class samples, evaluated on the imbalanced isotropic Gaussian mixture data from Setting 2 of Section 4.1, with mean vectors defined by the h function (16).

IR($\mathbb{P}$)	Method	Accuracy	TPR	TNR	G_{mean}	F_2
49	GSW$_{\text{SVM}}$	0.7954(0.0433)	0.8087(0.0613)	0.7951(0.0449)	0.7974(0.0282)	0.2799(0.0401)
	SVM	0.9757(0.0033)	0.2679(0.0606)	0.9902(0.0029)	0.4859(0.0779)	0.2927(0.0578)
	ClusterSVM	0.9794(0.0014)	0.0061(0.0139)	0.9993(0.0015)	0.0178(0.0381)	0.0069(0.0155)
	SMOTE-SVM	0.9749(0.0032)	0.2483(0.0607)	0.9897(0.0028)	0.4723(0.0708)	0.2685(0.0593)
	BLSMOTE-SVM	0.9743(0.0032)	0.2323(0.0571)	0.9895(0.0030)	0.4541(0.0719)	0.2552(0.0542)
	DBSMOTE-SVM	0.9730(0.0031)	0.2060(0.0543)	0.9886(0.0028)	0.4218(0.0713)	0.2280(0.0532)
19	GSW$_{\text{SVM}}$	0.8793(0.0287)	0.7871(0.0379)	0.8842(0.0309)	0.8325(0.0199)	0.5683(0.0428)
	SVM	0.9532(0.0054)	0.4334(0.0783)	0.9805(0.0058)	0.6311(0.1026)	0.4701(0.0403)
	ClusterSVM	0.9456(0.0051)	0.0309(0.0263)	0.9937(0.0063)	0.1016(0.0649)	0.0342(0.0269)
	SMOTE-SVM	0.9482(0.0057)	0.5206(0.0475)	0.9707(0.0047)	0.7078(0.0336)	0.5117(0.0458)
	BLSMOTE-SVM	0.9466(0.0067)	0.4058(0.0478)	0.9750(0.0063)	0.6233(0.0448)	0.4159(0.0452)
	DBSMOTE-SVM	0.9442(0.0056)	0.3885(0.0462)	0.9735(0.0050)	0.6101(0.0372)	0.3955(0.0452)
9	GSW$_{\text{SVM}}$	0.9021(0.0101)	0.8178(0.0195)	0.9115(0.0109)	0.8628(0.0118)	0.7283(0.0198)
	SVM	0.9295(0.0072)	0.5579(0.0932)	0.9708(0.0066)	0.7178(0.1163)	0.5996(0.0273)
	ClusterSVM	0.8831(0.0110)	0.1072(0.0413)	0.9693(0.0154)	0.2814(0.0688)	0.1164(0.0410)
	SMOTE-SVM	0.9079(0.0101)	0.7113(0.0279)	0.9297(0.0107)	0.8124(0.0169)	0.6653(0.0261)
	BLSMOTE-SVM	0.9035(0.0137)	0.5822(0.0353)	0.9392(0.0152)	0.7380(0.0226)	0.5672(0.0323)
	DBSMOTE-SVM	0.9038(0.0124)	0.5541(0.0355)	0.9427(0.0132)	0.7212(0.0239)	0.5461(0.0337)
5.67	GSW$_{\text{SVM}}$	0.8958(0.0088)	0.8350(0.0160)	0.9065(0.0096)	0.8697(0.0101)	0.7782(0.0158)
	SVM	0.9135(0.0093)	0.6345(0.0669)	0.9628(0.0071)	0.7719(0.0765)	0.6670(0.0221)
	ClusterSVM	0.8985(0.0154)	0.6038(0.0739)	0.9505(0.0104)	0.7508(0.0549)	0.6146(0.0711)
	SMOTE-SVM	0.8552(0.0139)	0.8525(0.0208)	0.8557(0.0178)	0.8535(0.0100)	0.7517(0.0148)
	BLSMOTE-SVM	0.8767(0.0171)	0.6785(0.0352)	0.9116(0.0228)	0.7854(0.0174)	0.6546(0.0249)
	DBSMOTE-SVM	0.8770(0.0147)	0.6457(0.0325)	0.9178(0.0188)	0.7688(0.0179)	0.6312(0.0255)

estimation: estimating each mixture component individually, and estimating the overall mixture distribution. The first focuses on recovering the number of components and their parameters, while the second aims to approximate the true underlying distribution $\mathbb{P}$ without strictly determining the number of components.

Table 4: The means and standard deviations (in the parenthesis) of the assessment metrics for GSWSVM with IR($\mathbb{P}$) = 19, evaluated on the imbalanced Gaussian mixture data in Setting 1 of Section 4.1.1.

$\widehat{\text{OR}}(\mathbb{P}_{\text{IR}^\dagger})$	Accuracy	TPR	TNR	G_{mean}	F_2
2	0.8977(0.0140)	0.7241(0.0367)	0.9068(0.0144)	0.8087(0.0225)	0.5617 (0.0364)
3	0.9000(0.0138)	0.7393(0.0346)	0.9085(0.0142)	0.8181(0.0211)	0.5748 (0.0352)
4	0.9005(0.0147)	0.7459(0.0344)	0.9087(0.0152)	0.8218(0.0212)	0.5801 (0.0367)
5	0.8993(0.0141)	0.7481(0.0327)	0.9073(0.0146)	0.8223(0.0201)	0.5830 (0.0354)
6	0.8997(0.0148)	0.7516(0.0349)	0.9075(0.0155)	0.8244(0.0209)	0.5818 (0.0362)
7	0.8990(0.0150)	0.7531(0.0349)	0.9066(0.0155)	0.8249(0.0215)	0.5814 (0.0379)
8	0.8983(0.0156)	0.7542(0.0336)	0.9058(0.0163)	0.8252(0.0204)	0.5810 (0.0368)
9	0.8972(0.0156)	0.7546(0.0339)	0.9047(0.0162)	0.8248(0.0206)	0.5797 (0.0363)

While the first approach is sensitive to data dimension and the distances between cluster means, the second approach exhibits a milder dependency on the data dimension and is independent of cluster separations (Suresh et al., 2014). This difference can be quantified by comparing the minimax rates: $\sqrt{d/n}/\lambda^2$ (Azizyan et al., 2013) for the first approach and $(\log n)^{d/4}/n^{1/2}$ (Kim and Guntuboyina, 2022) for the second approach, where λ is the minimum ℓ_1 distance between clusters. Our method lies between these two approaches. While it effectively captures the clustered structure of the positive class by estimating the number of clusters, its theoretical justification rests on reliable estimation of the overall distribution. Given that our focus is not exclusively on estimating each individual mixture component, the dependency on d should not be overly severe.

We numerically validate this conjecture under the following setting. To maintain a consistent distance between mean vectors across different d, we adopt a setup different from Section 4.1.1. The positive class distribution is a four-component Gaussian mixture on $\mathbb{R}^d$ with uniform mixing probabilities, which is defined as follows. Following Doss et al. (2023) where cluster means are contained in a unit ball, we set the cluster means as vertices of a square centered at $\mathbf{0}$. Let $\boldsymbol{\mu}_0 := (1, 1, \ldots, 1) \in \mathbb{R}^d$. Also, let $\boldsymbol{\nu}_0 := (1, \ldots, 1, -1, \ldots, -1) \in \mathbb{R}^d$, with the first half as 1 and the second half as -1 (assuming even d). By setting

$$\boldsymbol{\mu}_1 = (2\sqrt{d})^{-1}\boldsymbol{\mu}_0, \boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1, \boldsymbol{\mu}_3 = (2\sqrt{d})^{-1}\boldsymbol{\nu}_0, \text{ and } \boldsymbol{\mu}_4 = -\boldsymbol{\mu}_3,$$

all cluster means are contained in the 1-ball. The covariance matrix is set to

$0.04I_d$. We try $d = 4$ to 36 in steps of 2, and $n^+ \in \{50, 100, 200, 300, 400, 500\}$. We assess the estimation performance of EM algorithm for the distribution and the number of clusters. The distribution estimation performance is measured by the widely used 2-Wasserstein distance between Gaussian mixtures (Delon and Desolneux, 2020), calculated via R package `LOMAR` (Heriche, 2021).

The results in Figure 3 indicate that with a sample size of around 200, the EM algorithm effectively estimates both the distribution and number of clusters for dimensions up to 20. This aligns with Table 2 where $d = 21$, which demonstrates significant improvements in the classification performance of GSWSVM as n^+ increases from 60 to 150.

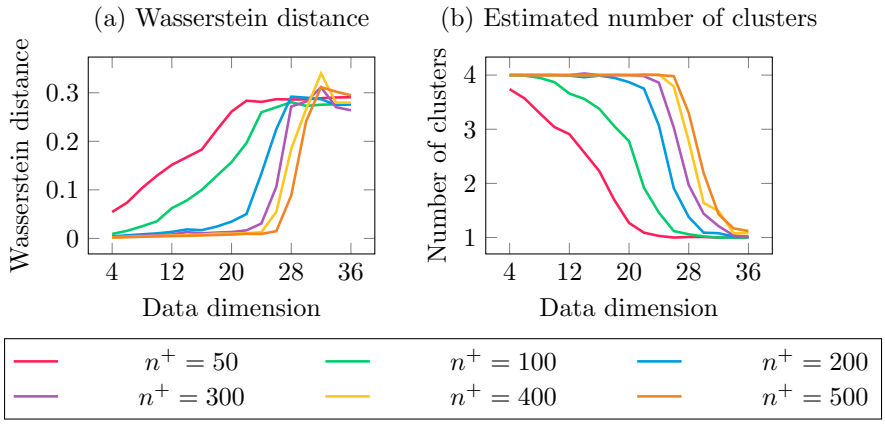


Figure 3: Performances of Gaussian mixture distribution estimation (measured by 2-Wasserstein disantce between the true and estimated Gaussian mixture) and cluster number estimation via the EM algorithm across various data dimensions and sample sizes, averaged over 200 independent repetitions.

4.2. Real data analysis

We now shift our focus to a real-world dataset consisting of 984 records from artillery military training sessions of ROKA. Each record consists of a label and four features. The label indicates whether a wildfire occurred during the session ($Y = 1$) or not ($Y = -1$). Only 40 observations are labeled as wildfire, resulting in an extremely imbalanced dataset with $\widehat{\mathrm{IR}}(\mathbb{P}) = 23.6$. The features consist of four meteorological variables (precipitation, wind speed, temperature, and relative humidity). Table 4.2 provides detailed descriptions and units for these features.

Table 5: Meteorological features in the ROKA training wildfire dataset.

Variable	Detailed variable description	Unit
Temperature	Instantaneous temperature measured at a height of 2 m above the ground	Degree Celsius
Precipitation	Accumulative precipitation over last 60 min	mm
Wind speed	Average of 10 min at a height of 10 m above the ground	m/s
Relative humidity	-	%

The boxplot (Figure 4) and the pairwise scatterplot (Figure 5) show the difference in feature distributions between classes, suggesting their usefulness for classification and potential subgroup structures in the positive class. To validate the subgroup assumption for the positive class, we begin with a visual inspection. The 3D scatter plot (Figure 1b), the pairwise scatter plot (Figure 5), scatter plot of the first two principal components (Figure 6a), and the dendrogram from hierarchical clustering (Figure 6b) suggest the presence of 3-4 subgroups. To further substantiate our visual observations, we employ formal statistical methods for estimating the number of clusters: Gap statistic (Tibshirani et al., 2001) and Jump statistic (Sugar and James, 2003). These statistics increase the number of cluster and identify the point where within-cluster dissimilarity starts to decrease more slowly. They are effective across a wide range of distributions, even without a clear subgroup structure, making them valuable for identifying whether subgroups exist or not, i.e. $K^* = 1$ versus $K^* > 1$. Appendix D explains these methods in detail. The results summarized in Table 4.2 provide strong evidence that the positive class possesses a subgroup structure, confirming our conjecture.

Our analysis consider the following settings. We try the oversample ratios of $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) = 23.6, 11.55, 4.72, 2.36$ and 1.57, corresponding to the desired imbalance ratio of $\mathrm{IR}^\dagger = 1, 2, 5, 10$, and 15, respectively. We set $c^- = 23.6$ $(= \widehat{\mathrm{IR}}(\mathbb{P}))$ and $c^+ = 1$. Given the adverse effect of heterogeneous scale on SVM and K-nearest neighbor (Hastie et al., 2009), which SMOTE variants rely on, we normalize features to a range between zero and one prior to analysis.

The result given in Table 7 indicates that on the real dataset, GSWSVM demonstrates superior performance compared to other baseline methods in terms of G_{mean}. This superiority is consistent over all desired imbalance ratio we considered. Although the assumptions of the asymptotic results in Theorem 3.1 and Theorem 3.2 do not hold in the finite-sample regime with

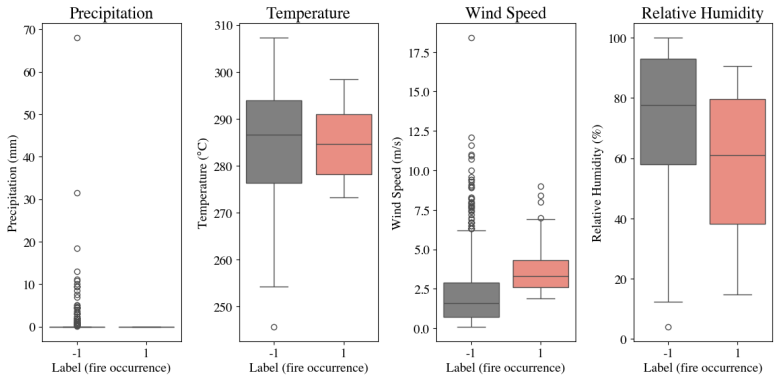


Figure 4: Boxplots of the features for the artillery training sessions in which wildfire occured ($Y = 1$) and not occured ($Y = -1$).

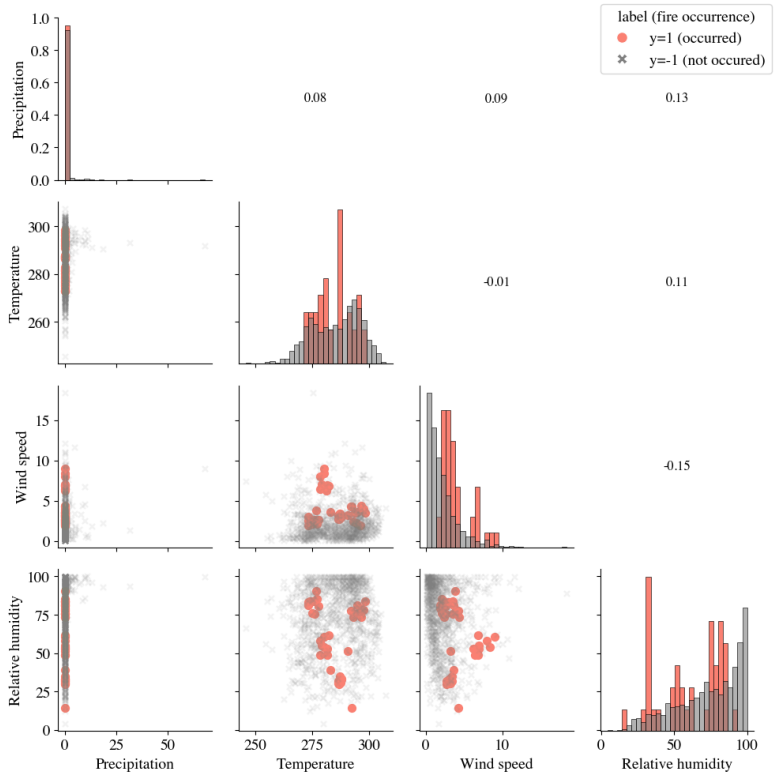
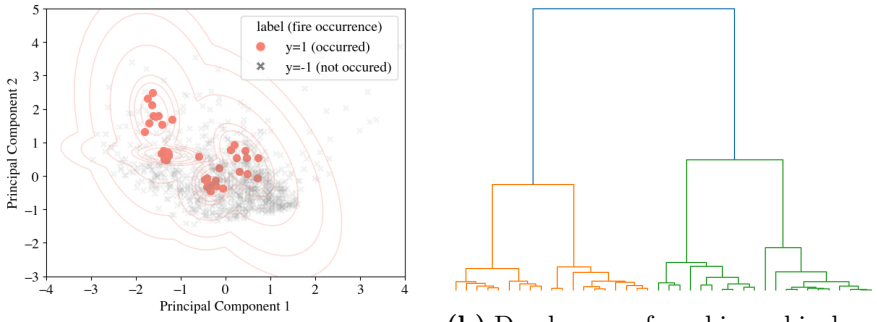


Figure 5: Feature distributions in the ROKA wildfire dataset. Lower triangular panes: pairwise scatterplot. Diagonal panes: class-wise histograms ($Y = 1$ and $Y = -1$). Upper triangular panes: correlation coefficients.



(a) First two principal components

(b) Dendrogram from hierarchical agglomerative clustering

Figure 6: Visual inspection on the subgroup structure of the positive class in the ROKA wildfire dataset: (a) First two principal components PCs, with red contour lines showing the positive class density, estimated via Gaussian mixture model on the first two PC scores, suggesting the subgroup structure. (b) Dendrogram for hierarchical agglomerative clustering (Euclidean distance dissimilarity and Ward linkage) on the positive class, suggesting a four-cluster structure.

Table 6: Estimated cluster number of the positive class by Gap statistic (Tibshirani et al., 2001) and Jump statistic Sugar and James (2003). For the Gap statistic, the number of resampling is set to $B = 100$. For the Jump statistic, R represents negative power transform of rate distortion, which is recommended to be set as $R < d/2$ for non-Gaussian distributions; Details are given in Appendix D.

Method	Estimated number of clusters
Gap statistic	4
Jump statistic ($R = 0.5$)	4
Jump statistic ($R = 0.75$)	4
Jump statistic ($R = 1.0$)	10
Jump statistic ($R = 1.5$)	10
Jump statistic ($R = 2.0$)	10

Table 7: The means and standard errors (in parentheses) of assessment metrics for each method and desired imbalance ratio, evaluated on real dataset presented in Section 4.2. For the real dataset, SVM and ClusterSVM produced zero true positives, resulting in non-meaningful values of G_{mean}. Therefore their results are omitted in this table.

$\text{IR}^\dagger$	$\widehat{\text{OR}}(\mathbb{P}_{\text{IR}^\dagger})$	Method	Accuracy	TPR	G_{mean}
15	1.57	GSWSVM	0.945(0.005)	0.761(0.043)	0.847(0.026)
		SMOTE-SVM	0.966(0.004)	0.589(0.047)	0.752(0.034)
		BLSMOTE-SVM	0.967(0.004)	0.568(0.046)	0.737(0.034)
		DBSMOTE-SVM	0.966(0.004)	0.581(0.051)	0.746(0.037)
10	2.36	GSWSVM	0.936(0.005)	0.809(0.039)	0.869(0.022)
		SMOTE-SVM	0.963(0.004)	0.620(0.044)	0.771(0.030)
		BLSMOTE-SVM	0.966(0.004)	0.586(0.047)	0.750(0.034)
		DBSMOTE-SVM	0.962(0.004)	0.622(0.053)	0.770(0.038)
5	4.72	GSWSVM	0.925(0.009)	0.839(0.041)	0.879(0.023)
		SMOTE-SVM	0.955(0.004)	0.711(0.049)	0.822(0.031)
		BLSMOTE-SVM	0.963(0.004)	0.630(0.049)	0.777(0.033)
		DBSMOTE-SVM	0.957(0.004)	0.713(0.055)	0.824(0.036)
2	11.55	GSWSVM	0.917(0.012)	0.857(0.037)	0.885(0.020)
		SMOTE-SVM	0.937(0.009)	0.801(0.044)	0.865(0.026)
		BLSMOTE-SVM	0.955(0.004)	0.708(0.050)	0.820(0.033)
		DBSMOTE-SVM	0.937(0.011)	0.8(0.044)	0.864(0.028)
1	23.6	GSWSVM	0.915(0.013)	0.857(0.041)	0.884(0.022)
		SMOTE-SVM	0.920(0.013)	0.857(0.035)	0.886(0.020)
		BLSMOTE-SVM	0.948(0.006)	0.744(0.053)	0.838(0.033)
		DBSMOTE-SVM	0.921(0.015)	0.841(0.041)	0.878(0.025)

$n = 944$, these theorems provide an intuition for why GSWSVM demonstrates superior performance compared to other baseline methods on this dataset.

The dataset used in this analysis was created by Nam et al. (2024), preprocessing and combining the data from the ROKA and Korea Meteorological Administration. While partial data is accessible at the following URL: https://github.com/jihyun-nam/Prediction-of-Forest-Fire-Risk/tree/main, access to full dataset requires permission from the ROKA. Please contact the author to acquire the necessary permissions.

5. Conclusion

To improve the SVM performance on extremely imbalanced datasets with subgroup structures in the positive class, we integrate oversampling and cost-sensitive approaches. We capture the subgroup structure by first learning the positive class conditional distribution using a Gaussian mixture model. Given the extreme imbalance, a substantial number of synthetic samples should be generated from this learned distribution. To control the variability introduced by this synthetic sample generation, we apply different misclassification penalties for the original positive, synthetic positive, and negative class. Our proposed scheme for determining these misclassification penalties is motivated by the asymptotic optimality of the weighted SVM under label shift (Theorem 3.1 and Theorem 3.2), and reflects the imbalance ratio, oversampling ratio and decision-theoretical misclassification costs. We numerically validate our method through synthetic and real data analysis.

Our method requires exploratory data analysis (EDA) for finding the subgroup structure. While formal guidelines for EDA are not provided, practitioners can adapt techniques creatively based on data characteristics. In extreme situations where the sample size is too small to detect subgroup structures, or when it cannot be reasonably concluded that the positive class has a subgroup structure, our method, which relies on subgroup assumptions, may not be applicable. In such cases, the estimated conditional distribution would be a single Gaussian, potentially deviating significantly from the real distribution and leading to compromised performance.

This poses an important direction for future research that involves exploring alternatives to the GMM for learning the positive class distribution. Specifically, modern deep neural network models, such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), hold promise but also present challenges, including overfitting, mode collapse, and heightened sensitivity to hyperparameters, particularly due to the limited sample size typical in imbalanced data studies. Addressing these challenges necessitates careful methodological considerations, and thus, the application of various advanced techniques for learning the probability distribution to this problem is a promising avenue for future research.

Appendix A. Proof of Lemma 3.1

The proof follows the approach in Lemma 3.1 of Lin (2002) and the explanation under equation (16) of Lin (2002), which are generalized into

Theorem 3.1 and Lemma 4.1 of their later work (Lin, 2004). We first rephrase Lemma 4.1 of Lin (2004) that outlines the statistical assumptions required for the asymptotic result in Lemma 3.1.

Lemma Appendix A.1. *Let $\bar{f}, \bar{b}$ be the global minimizer of the population-level risk $\mathbb{E}\big(\ell(Y, X, f, b)\big)$. Suppose that $\bar{f}, \bar{b}$ satisfies the condition*

$$|\phi^*(\boldsymbol{x})| \leq c|\bar{f}(\boldsymbol{x}) + \bar{b}| \text{ for all } \boldsymbol{x}, \tag{A.1}$$

where $c > 0$ does not depend on $\boldsymbol{x}$, and we recall that ϕ^* is the Bayes rule (2). Then for any function f and $b \in \mathbb{R}$,

$$R_{\mathbb{P}}\big(\text{sign}(f(\boldsymbol{x}) + b)\big) - R^*_{\mathbb{P}} \leq c\bigg\{\int |f(\boldsymbol{x}) + b - \bar{f}(\boldsymbol{x}) - \bar{b}|^2 d(\boldsymbol{x}) d\boldsymbol{x}\bigg\}^{1/2},$$

where $d(\boldsymbol{x})$ is the density of $\boldsymbol{X}$ and $R^*_{\mathbb{P}}$ is the Bayes optimal risk.

We will show that the condition (A.1) holds for Theorem 3.1 by showing that the function $\bar{f}(\boldsymbol{x}) + \bar{b}$ where

$$(\bar{f}, \bar{b}) \in \underset{f \in H_K, b \in \mathbb{R}}{\arg\min}\, \mathbb{E}\big[L_{\text{IR}^\dagger}(Y_{\text{IR}^\dagger}, Z)\big(1 - Y_{\text{IR}^\dagger}\big(f(\boldsymbol{X}_{\text{IR}^\dagger}) + b\big)\big)_+\big] \tag{A.2}$$

is the same as the Bayes rule (3). Then lemma Appendix A.1 implies that the statistical assumption for Theorem 3.1 is that the solution of the regularized ERM problem (10):

$$(f_n, b_n) = \underset{f \in H_K, b \in \mathbb{R}}{\arg\min}\, \lambda_n \|f\|^2_{H_K} + \frac{1}{n}\sum_{i=1}^{n} L_{\text{IR}^\dagger}(Y_{\text{IR}^\dagger,i}, Z_i)\big(1 - Y_{\text{IR}^\dagger,i}\big(f(\boldsymbol{X}_{\text{IR}^\dagger,i}) + b\big)\big)_+$$

converges in L_2 to $\bar{f}(\boldsymbol{x}) + \bar{b}$. This L_2 convergence is implied by the richness of the RKHS H_K, and is stated in Lin (2002) in our setting.

Now we show the equivalent of the Bayes rule and $\bar{f}(\boldsymbol{x}) + \bar{b}$. First, for simplicity, denote $g(\boldsymbol{X}_{\text{IR}^\dagger}) := f(\boldsymbol{X}_{\text{IR}^\dagger}) + b$. We aim to show that

$$\bar{g} = \underset{g}{\arg\min}\, \mathbb{E}\big[L_{\text{IR}^\dagger}(Y_{\text{IR}^\dagger}, Z)\big(1 - Y_{\text{IR}^\dagger} g(\boldsymbol{X}_{\text{IR}^\dagger})\big)_+\big] \tag{A.3}$$

is the same as the Bayes rule (3). By the law of iterated expectation, we

have

$$\mathbb{E}[L_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}, Z)\big(1-Y_{\mathrm{IR}^\dagger}g(\boldsymbol{X}_{\mathrm{IR}^\dagger})\big)_+] = \mathbb{E}\big[\mathbb{E}[L_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}, Z)(1-Y_{\mathrm{IR}^\dagger}g(\boldsymbol{X}_{\mathrm{IR}^\dagger}))_+|\boldsymbol{X}_{\mathrm{IR}^\dagger}]\big].$$

Therefore, if $\bar{g}(\boldsymbol{x})$ minimizes

$$\mathbb{E}\big[\mathbb{E}[L_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}, Z)(1 - Y_{\mathrm{IR}^\dagger}g(\boldsymbol{X}_{\mathrm{IR}^\dagger}))_+|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}]\big] \tag{A.4}$$

for all given $\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}$, then $\bar{g}$ also solves (A.3). This approach reduces the optimization problem on RKHS into the one on $\mathbb{R}$. We show that in the reduced problem, the solution lies in $[-1, 1]$ and the objective function becomes a linear function on $[-1, 1]$. Then, the solution is derived naturally.

Fix $\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}$. As in (7), denote the posterior probability of $\mathbb{P}_{\mathrm{IR}^\dagger}$ as

$$p_s(\boldsymbol{x}) = \mathbb{P}_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger} = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) = \frac{\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})g^+(\boldsymbol{x})}{\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})g^+(\boldsymbol{x}) + \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})g^-(\boldsymbol{x})}.$$

For simplicity, we define the following related quantities:

$$\begin{aligned}
p_s^{syn}(\boldsymbol{x}) &:= \mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1, Z = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&= \mathbb{P}(Z = 1|Y_{\mathrm{IR}^\dagger} = 1, \boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x})\mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&\overset{(i)}{=} \mathbb{P}(Z = 1|Y_{\mathrm{IR}^\dagger} = 1)\mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&= \mathbb{P}(Z = 1|Y_{\mathrm{IR}^\dagger} = 1)p_s(\boldsymbol{x}) \\
&= \frac{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})}p_s(\boldsymbol{x}), \quad \text{and} \\
p_s^{og}(\boldsymbol{x}) &:= \mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1, Z = 0|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&= \mathbb{P}(Z = 0|Y_{\mathrm{IR}^\dagger} = 1, \boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x})\mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&\overset{(ii)}{=} \mathbb{P}(Z = 0|Y_{\mathrm{IR}^\dagger} = 1)\mathbb{P}(Y_{\mathrm{IR}^\dagger} = 1|\boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}) \\
&= \mathbb{P}(Z = 0|Y_{\mathrm{IR}^\dagger} = 1)p_s(\boldsymbol{x}) \\
&= \frac{1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})}p_s(\boldsymbol{x}),
\end{aligned} \tag{A.5}$$

where (i) and (ii) use conditional independence. For simplicity, let $w := g(\boldsymbol{x}) \in \mathbb{R}$. Since $(Y_{\mathrm{IR}^\dagger}, Z)$ is a discrete random pair, the expectation (A.4) is

expressed as a sum as follows:

$$\begin{aligned} A(w) &:= \mathbb{E}[L_{\mathrm{IR}^\dagger}(Y_{\mathrm{IR}^\dagger}, Z)(1 - Y_{\mathrm{IR}^\dagger} w)_+ \mid \boldsymbol{X}_{\mathrm{IR}^\dagger} = \boldsymbol{x}] \\ &= L_{\mathrm{IR}^\dagger}(1,0)(1-w)_+ p_s^{og}(\boldsymbol{x}) + L_{\mathrm{IR}^\dagger}(1,1)(1-w)_+ p_s^{syn}(\boldsymbol{x}) \\ &\quad + L_{\mathrm{IR}^\dagger}(-1,0)(1+w)_+ (1 - p_s(\boldsymbol{x})). \end{aligned} \tag{A.6}$$

Because we fix $\boldsymbol{x}$, the quantities $p_s^{og}(\boldsymbol{x}), p_s^{syn}(\boldsymbol{x})$ and $p_s(\boldsymbol{x})$ are also fixed real numbers. The decision variable here is w, which is determined by the function g. This is why the expectation (A.6) is indexed solely by w.

First notice that the minimizer of $A(w)$ must be in $[-1, 1]$; For any w outside $[-1, 1]$, letting $w' = \mathrm{sign}(w)$, w' is in $[-1, 1]$ and it is easy to check that $A(w') < A(w)$. Therefore it suffices to search for the minimizer in $[-1, 1]$. Note that for this range of values of w, $(1 + w)_+ = 1 + w$ and $(1 - w)_+ = 1 - w$. For notational simplicity, let $\zeta := p_s^{og}(\boldsymbol{x}) L_{\mathrm{IR}^\dagger}(1,0) + p_s^{syn}(\boldsymbol{x}) L_{\mathrm{IR}^\dagger}(1,1)$ and $\varphi := \big(1 - p_s(\boldsymbol{x})\big) L_{\mathrm{IR}^\dagger}(-1,0)$. Then the expectation (A.4) is expressed as a linear function of w as follows:

$$A(w) = \zeta(1-w) + \varphi(1+w) = (\varphi - \zeta)w + (\varphi + \zeta).$$

Therefore, $A(w)$ is minimized at $w = 1$ when the slope is negative i.e. $\varphi - \zeta < 0$, and at $w = -1$ otherwise. The condition $\varphi - \zeta < 0$ is equivalent to

$$p_s^{og}(\boldsymbol{x}) L_{\mathrm{IR}^\dagger}(1,0) + p_s^{syn}(\boldsymbol{x}) L_{\mathrm{IR}^\dagger}(1,1) > (1 - p_s(\boldsymbol{x})) L_{\mathrm{IR}^\dagger}(-1,0).$$

Keeping in mind the definitions of $p_s^{og}(\boldsymbol{x})$ and $p_s^{syn}(\boldsymbol{x})$ in (A.5), this condition is equivalent to

$$p_s(\boldsymbol{x}) \big[L_{\mathrm{IR}^\dagger}(-1,0) + \frac{1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})} L_{\mathrm{IR}^\dagger}(1,0) + \frac{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})} L_{\mathrm{IR}^\dagger}(1,1) \big] > L_{\mathrm{IR}^\dagger}(-1,0).$$

Using $L_{\mathrm{IR}^\dagger}(-1,0) = L(-1)$ and the following equality:

$$\frac{1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})} L_{\mathrm{IR}^\dagger}(1,0) + \frac{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger}) - 1}{\mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})} L_{\mathrm{IR}^\dagger}(1,1) = L(1),$$

which is obvious from the definition of $L_{\mathrm{IR}^\dagger}$ function in (9), the condition above is equivalent to

$$p_s(\boldsymbol{x}) > L(-1)/(L(-1) + L(1)).$$

Thus the minimizer of $A(w)$, denoted as $\bar{w}$, is

$$\bar{w} = \begin{cases} 1 & \text{if} \quad p_s(\boldsymbol{x}) > \dfrac{L(-1)}{L(-1)+L(1)} \\ -1 & \text{otherwise,} \end{cases}$$

which means that

$$\bar{w} = \operatorname{sign}\left(p_s(\boldsymbol{x}) - \frac{L(-1)}{L(-1)+L(1)} \right).$$

Now by the equality (8), the lemma is proved.

Appendix B. Proof of Theorem 3.2

We prove the six convergence statements in the same order as presented in the theorem.

1. Proof of $n^+/n \xrightarrow{P} \pi^+(\mathbb{P})$. Since $\mathbb{1}(Y_i = 1) \overset{i.i.d.}{\sim} \mathrm{Ber}\big(\mathbb{P}(Y_1 = 1)\big)$, by the weak law of large numbers, we have:

$$\frac{n^+}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(Y_i = 1) \xrightarrow{P} \mathbb{E}\big[\mathbb{1}(Y_1 = 1)\big] = \mathbb{P}(Y_1 = 1) = \pi^+(\mathbb{P}). \quad \text{(B.1)}$$

2. Proof of $(n - n^+)/n \xrightarrow{P} \pi^-(\mathbb{P})$. By (B.1) and the continuous mapping theorem, we have:

$$(n - n^+)/n = 1 - n^+/n \xrightarrow{P} 1 - \pi^+(\mathbb{P}) = \pi^-(\mathbb{P}). \quad \text{(B.2)}$$

3. Proof of $\widehat{\mathrm{IR}}(\mathbb{P}) \xrightarrow{P} \mathrm{IR}(\mathbb{P})$. By (B.1), (B.2) and the continuous mapping theorem, we have:

$$\widehat{\mathrm{IR}}(\mathbb{P}) = \frac{n - n^+}{n^+} = \frac{(n - n^+)/n}{n^+/n} \xrightarrow{P} \frac{\pi^-(\mathbb{P})}{\pi^+(\mathbb{P})} = \mathrm{IR}(\mathbb{P}). \quad \text{(B.3)}$$

4. Proof of $(n - n^+)/(n + n_s^+) \xrightarrow{P} \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})$. Recall from (13) that $n_s^+ = \big\lfloor (n - n^+)/\mathrm{IR}^\dagger - n^+ \big\rfloor$. Define a quantity $\tilde{n}_s^+ := (n - n^+)/\mathrm{IR}^\dagger - n^+$. We

first show that $(n-n^+)/(n+\tilde{n}_s^+) = \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})$. Note that

$$n+\tilde{n}_s^+ = n+\left(\frac{n-n^+}{\mathrm{IR}^\dagger}-n^+\right) = (n-n^+)\frac{\mathrm{IR}^\dagger+1}{\mathrm{IR}^\dagger}.$$

Since $\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})+\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})=1$, the equality above implies

$$\frac{n-n^+}{n+\tilde{n}_s^+} = \frac{\mathrm{IR}^\dagger}{\mathrm{IR}^\dagger+1} = \frac{\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})/\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})}{\big(\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})+\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})\big)/\pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})} = \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger}).$$

Now to show that $(n+\tilde{n}_s^+)/(n+n_s^+) \xrightarrow{P} 1$, first note that

$$\tilde{n}_s^+ - 1 < n_s^+ = \lfloor \tilde{n}_s^+ \rfloor \le \tilde{n}_s^+,$$

which implies:

$$1 = \frac{n+\tilde{n}_s^+}{n+\tilde{n}_s^+} \le \frac{n+\tilde{n}_s^+}{n+n_s^+} = \frac{(n-n^+)/(n+n_s^+)}{\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})} < \frac{n+\tilde{n}_s^+}{n+\tilde{n}_s^+-1}.$$

Now it suffices to show that $(n+\tilde{n}_s^+)/(n+\tilde{n}_s^+-1) \xrightarrow{P} 1$. To this end, by continuous mapping theorem, it suffices to show the convergence of the reciprocal: $(n+\tilde{n}_s^+-1)/(n+\tilde{n}_s^+) \xrightarrow{P} 1$. Since $(n+\tilde{n}_s^+-1)/(n+\tilde{n}_s^+) = 1-1/(n+\tilde{n}_s^+)$, it suffices to show that $1/(n+\tilde{n}_s^+) \xrightarrow{P} 0$, which is trivially true. Therefore, we have $(n-n^+)/(n+n_s^+) \xrightarrow{P} \pi^-(\mathbb{P}_{\mathrm{IR}^\dagger})$.

5. Proof of $(n^+ + n_s^+)/(n+n_s^+) \xrightarrow{P} \pi^+(\mathbb{P}_{\mathrm{IR}^\dagger})$. By the previous convergence result and the continuous mapping theorem, we have:

$$\frac{n^+ + n_s^+}{n+n_s^+} = 1 - \frac{n-n^+}{n+n_s^+} \xrightarrow{P} 1-\pi^-(\mathbb{P}_{\mathrm{IR}^\dagger}) = \pi^+(\mathbb{P}_{\mathrm{IR}^\dagger}).$$

6. Proof of $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) \xrightarrow{P} \mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})$. Recall from (12) that $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) = \widehat{\mathrm{IR}}(\mathbb{P})/\mathrm{IR}^\dagger$. Since $\widehat{\mathrm{IR}}(\mathbb{P}) \xrightarrow{P} \mathrm{IR}(\mathbb{P})$ (B.3), by the continuous mapping theorem, we have $\widehat{\mathrm{OR}}(\mathbb{P}_{\mathrm{IR}^\dagger}) \xrightarrow{P} \mathrm{IR}(\mathbb{P})/\mathrm{IR}^\dagger = \mathrm{OR}(\mathbb{P}_{\mathrm{IR}^\dagger})$, where the last equality comes from (11). This concludes the proof of Theorem 3.2.

Appendix C. Baseline methods

This sections gives details for the baseline methods used in Section 4.

1. SVM (Cortes and Vapnik, 1995) is the original Gaussian kernel SVM without any oversampling or cost-sensitive learning.

2. SMOTE-SVM (Chawla et al., 2002) generates synthetic samples using the SMOTE algorithm, then fits Gaussian kernel SVM on the expanded dataset. SMOTE linearly interpolates between a positive class sample $\boldsymbol{x}_p$ and one of its five nearest neighbor of positive class, denoted as $\boldsymbol{x}_q$, to create a synthetic positive sample $\boldsymbol{x}_{new}$. In a more formal way:

$$\boldsymbol{x}_{new} = \boldsymbol{x}_p + (\boldsymbol{x}_q - \boldsymbol{x}_p) \times \delta,$$

   where δ is drawn from uniform distribution $\mathcal{U}[0, 1]$.

3. BLSMOTE-SVM replaces SMOTE algorithm in SMOTE-SVM with Boarderline SMOTE (BLSMOTE; Han et al. (2005)) algorithm. BLSMOTE generates synthetic samples through the same procedure as SMOTE, but the way of choosing neighbors is different; For a given positive class sample, select its m nearest neighbors regardless of their class. If more than half of them belong to the negative class, we call the given positive class sample as DANGER, which means that it is on the borderline of classes and is difficult to classify. Only the DANGER samples are considered as neighbors. For the value of m, we empirically set it as $|n_{min}|/4$.

4. DBSMOTE-SVM replaces SMOTE algorithm in SMOTE-SVM with Density-Based SMOTE (DBSMOTE; Bunkhumpornpat et al. (2012)) algorithm. DBSMOTE generates synthetic samples through the following procedure. We first identify arbitrarily shaped clusters using DBSCAN, a very popular density-based nonparametric clustering algorithm (Ester et al. (1996)). The values of all hyperparamters are automatically chosen by the algorithm in `smotefamily` package. Then for a given positive class sample, we generate synthetic samples along a shortest path from the centroid of the cluster it belongs to.

5. ClusterSVM (Gu and Han, 2013) divides the data into clusters and trains a linear SVM within each cluster to handle local separations.

Additionally, ClusterSVM incorporates a global regularization term, which aligns the weight vectors of each local linear SVM with a global weight vector. This global regularization helps leverage information across clusters and prevents overfitting within individual clusters. The ClusterSVM approach provides a balance between the computational efficiency of linear SVMs and the flexibility of kernel SVMs.

Appendix D. Details for cluster number estimation methods

This section gives detailed description on cluster number estimation methods used in Section 4.2 to validate the subgroup assumption of the positive class.

Gap statistic (Tibshirani et al., 2001). Let $W(K)$ be the within-cluster sum of squares for the original dataset, where K indicates the potential number of clusters. The Gap statistic compares the curve $\log\big(W(K)\big)$ to the curve obtained from data uniformly distributed over a rectangle containing the data. In a more formal way, B different uniform datasets, each with the same range as the original data, are produced. For bth uniform datasets, the within-cluster sum of squares $W_b^*(K)$ is calculated for different numbers of clusters K. Then the Gap statistic is defined as

$$\mathrm{Gap}(K) = \frac{1}{B}\sum_b \log\big(W_b^*(K)\big) - \log\big(W(K)\big).$$

One approach would be to maximize $\mathrm{Gap}(K)$. However, to avoid adding unnecessary clusters, an estimate of the standard deviation of $\log\big(W_b^*(K)\big)$, s_K, is produced, and the smallest value of K such that

$$\mathrm{Gap}(K) \geq \mathrm{Gap}(K+1) - s_{K+1}$$

is chosen as the number of clusters.

Jump statistic (Sugar and James, 2003). Let $\boldsymbol{X}$ be a p-dimensional random vector with a mixture distribution of K^* components, each with covariance Γ. For $K \geq 1$, let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_K$ be a set of candidate cluster centers, and let $\boldsymbol{c}_{\boldsymbol{X}}$ be the one closest to $\boldsymbol{X}$. Define

$$d_K = \frac{1}{p}\min_{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_K} \mathbb{E}[(\boldsymbol{X}-\boldsymbol{c}_{\boldsymbol{X}})^\top \Gamma^{-1}(\boldsymbol{X}-\boldsymbol{c}_{\boldsymbol{X}})],$$

which is simply the average Mahalanobis distance, per dimension, between $\boldsymbol{X}$ and $\boldsymbol{c_X}$. Let $\hat{d}_K$ be an estimate of d_K where the cluster centers are obtained by applying the k-means clustering algorithm to the observed data. Then we can obtain a curb by plotting $\hat{d}_K$ versus K. Sugar and James (2003) shows that for a large class of distributions, this curve, when transformed to an appropriate negative power, will exhibit a sharp jump at the true number of clusters. In a more formal way, the procedure for estimating K is as follows:

1. Run the k-means algorithm for different numbers of clusters, K, and calculate the corresponding $\hat{d}_K$.

2. Select a transformation power, $R > 0$ (A typical value is $R = p/2$).

3. Calculate the "jumps": $J(K) = \hat{d}_K^{-R} - \hat{d}_{K-1}^{-R}$.

4. Estimate the number of clusters in the dataset by

$$\hat{K}^* = \arg\max_K J(K),$$

the value of K associated with the largest jump. Note that we define $\hat{d}_0^{-R} \equiv 0$, so that the method can select $\hat{K}^* = 1$ if there is no clustering in the data. For approximately normal data, one may use a large value of R, but for very non-Gaussian data the transformation power needs to be considerably lower.

References

Azizyan, M., Singh, A., Wasserman, L., 2013. Minimax theory for high-dimensional gaussian mixtures with sparse mean separation, in: Advances in Neural Information Processing Systems.

Balakrishnan, S., Wainwright, M.J., Yu, B., 2017. Statistical guarantees for the EM algorithm: From population to sample-based analysis. The Annals of Statistics 45, 77–120.

Bang, S., Jhun, M., 2014. Weighted Support Vector Machine Using k-Means Clustering. Communications in Statistics - Simulation and Computation 43, 2307–2324.

Bang, S., Kim, J., 2020. Sampling Method Using Gaussian Mixture Clustering for Classification Analysis of Imbalanced Data. Journal of The Korean Data Analysis Society 22, 565–574.

Bao, F., Xu, K., Li, C., Hong, L., Zhu, J., Zhang, B., 2021. Variational (gradient) estimate of the score function in energy-based latent variable models, in: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning, PMLR. pp. 651–661.

Bartlett, P.L., Jordan, M.I., McAuliffe, J.D., 2006. Convexity, Classification, and Risk Bounds. Journal of the American Statistical Association 101, 138–156.

Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.B. (Eds.), Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 475–482.

Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2012. DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. Applied Intelligence 36, 664–684.

Chaure, F.J., Rey, H.G., Quian Quiroga, R., 2018. A novel and fully automatic spike-sorting implementation with variable number of features. Journal of Neurophysiology 120, 1859–1871.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.

Chi, J., Tian, Y., Gordon, G.J., Zhao, H., 2021. Understanding and Mitigating Accuracy Disparity in Regression, in: Proceedings of the 38th International Conference on Machine Learning, PMLR. pp. 1866–1876.

Chung, J.E., Magland, J.F., Barnett, A.H., Tolosa, V.M., Tooker, A.C., Lee, K.Y., Shah, K.G., Felix, S.H., Frank, L.M., Greengard, L.F., 2017. A Fully Automated Approach to Spike Sorting. Neuron 95, 1381–1394.e6.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20, 273–297.

Delon, J., Desolneux, A., 2020. A wasserstein-type distance in the space of gaussian mixture models. SIAM Journal on Imaging Sciences 13, 936–970.

Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39, 1–38.

Doss, N., Wu, Y., Yang, P., Zhou, H.H., 2023. Optimal estimation of high-dimensional Gaussian location mixtures. The Annals of Statistics 51, 62–95.

Duchi, J.C., Namkoong, H., 2021. Learning models with uniform performance via distributionally robust optimization. The Annals of Statistics 49, 1378–1406.

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, Portland, Oregon. pp. 226–231.

Fatumo, S., Chikowore, T., Choudhury, A., Ayub, M., Martin, A.R., Kuchenbaecker, K., 2022. A roadmap to increase diversity in genomic studies. Nature Medicine 28, 243–250.

Fonseca, J.R., 2013. Clustering in the field of social sciences: that is your choice. International Journal of Social Research Methodology 16, 403–428.

Garg, S., Wu, Y., Balakrishnan, S., Lipton, Z., 2020. A Unified View of Label Shift Estimation, in: Advances in Neural Information Processing Systems, pp. 3290–3300.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

Gu, Q., Han, J., 2013. Clustered Support Vector Machines, in: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, PMLR. pp. 307–315.

Han, H., Wang, W.Y., Mao, B.H., 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning, in: Huang, D.S., Zhang, X.P., Huang, G.B. (Eds.), Advances in Intelligent Computing, Springer, Berlin, Heidelberg. pp. 878–887.

Hastie, T., Tibshirani, R., 1996. Discriminant Analysis by Gaussian Mixtures. Journal of the Royal Statistical Society: Series B (Methodological) 58, 155–176.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Springer Series in Statistics, Springer, New York, NY.

He, H., Bai, Y., Garcia, E.A., Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328.

He, T., 2016. SwarmSVM: Ensemble Learning Algorithms Based on Support Vector Machines. R package version 0.1-7.

Heriche, J.K., 2021. LOMAR: Localization Microscopy Data Analysis. R package version 0.5.0.

Kim, A.K.H., Guntuboyina, A., 2022. Minimax bounds for estimating multivariate Gaussian location mixtures. Electronic Journal of Statistics 16, 1461–1484.

Krawczyk, B., 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5, 221–232.

Kuhn, M., 2008. Building predictive models in R using the caret package. Journal of Statistical Software 28, 1–26.

Kwon, J., Caramanis, C., 2020. The EM Algorithm gives Sample-Optimality for Learning Mixtures of Well-Separated Gaussians, in: Proceedings of Thirty Third Conference on Learning Theory, PMLR. pp. 2425–2487.

Lin, Y., 2002. Support Vector Machines and the Bayes Rule in Classification. Data Mining and Knowledge Discovery 6, 259–275.

Lin, Y., 2004. A note on margin-based loss functions in classification. Statistics & Probability Letters 68, 73–82.

Lin, Y., Lee, Y., Wahba, G., 2002. Support Vector Machines for Classification in Nonstandard Situations. Machine Learning 46, 191–202.

Maaten, L.v.d., Hinton, G., 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research 9, 2579–2605.

Mangasarian, O.L., Musicant, D.R., 2001. Lagrangian support vector machines. Journal of Machine Learning Research 1, 161–177.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., 2024. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-16.

Nam, J.H., Mun, J., Jo, S., Kim, J., 2024. Prediction of Forest Fire Risk for Artillery Military Training using Weighted Support Vector Machine for Imbalanced Data. Journal of Classification 41, 170–189.

Park, Y.G., Kwon, Y.W., Koh, C.S., Kim, E., Lee, D.H., Kim, S., Mun, J., Hong, Y.M., Lee, S., Kim, J.Y., Lee, J.H., Jung, H.H., Cheon, J., Chang, J.W., Park, J.U., 2024. In-vivo integration of soft neural probes through high-resolution printing of liquid electronics on the cranium. Nature Communications 15, 1772.

Schwarz, G., 1978. Estimating the dimension of a model. The Annals of Statistics 6, 461–464.

Scrucca, L., Fraley, C., Murphy, T., Raftery, A., 2023. Model-Based Clustering, Classification, and Density Estimation Using mclust in R. Chapman and Hall/CRC.

Segol, N., Nadler, B., 2021. Improved convergence guarantees for learning Gaussian mixture models by EM and gradient EM. Electronic Journal of Statistics 15, 4510–4544.

Simpson, E.H., 1951. The Interpretation of Interaction in Contingency Tables. Journal of the Royal Statistical Society. Series B (Methodological) 13, 238–241.

Siriseriwan, W., 2024. smotefamily: A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE. R package version 1.4.0.

Sugar, C.A., James, G.M., 2003. Finding the Number of Clusters in a Dataset: An Information-Theoretic Approach. Journal of the American Statistical Association 98, 750–763.

Suresh, A.T., Orlitsky, A., Acharya, J., Jafarpour, A., 2014. Near-Optimal-Sample Estimators for Spherical Gaussian Mixtures, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.

Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the Number of Clusters in a Data Set Via the Gap Statistic. Journal of the Royal Statistical Society Series B: Statistical Methodology 63, 411–423.

Veropoulos, K., Campbell, C., Cristianini, N., 1999. Controlling the sensitivity of support vector machines, in: Proceedings of the international joint conference on AI, Stockholm. p. 60.

Wang, X., Liu, X., Matwin, S., Japkowicz, N., 2014. Applying instance-weighted support vector machines to class imbalanced datasets, in: 2014 IEEE International Conference on Big Data (Big Data), pp. 112–118.

Wu, G., Chang, E.Y., 2003. Class-boundary alignment for imbalanced dataset learning, in: ICML 2003 workshop on learning from imbalanced data sets II, Washington, DC. pp. 49–56.

Xu, T., Chang, C.C., Lin, C.C., Chang, M.W., Lin, H.T., Tsai, M.H., Ho, C.H., Yu, H.F., Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., 2011. WeightSVM: Subject Weighted Support Vector Machines. R package version 1.7-16.

Yang, X., Song, Q., Cao, A., 2005. Weighted support vector machine for data classification, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., pp. 859–864 vol. 2.