Statistica Sinica

# LEVERAGE CLASSIFIER: ANOTHER LOOK AT SUPPORT VECTOR MACHINE

Yixin Han[1], Jun Yu[2], Nan Zhang[3], Cheng Meng[4], Ping Ma[5], Wenxuan Zhong[5], and Changliang Zou[1]

[1] *School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin, P.R. China*

[2] *School of Mathematics and Statistics, Beijing Institute of Technology, Beijing, P.R.China*

[3] *School of Data Science, Fudan University, Shanghai, P.R.China*

[4] *Institute of Statistics and Big Data, Renmin University, Beijing, P.R.China*

[5] *Department of Statistics, University of Georgia, Athens, GA, USA*

*Abstract:* Support vector machine (SVM) is a popular classifier known for accuracy, flexibility, and robustness. However, its intensive computation has hindered its application to large-scale datasets. In this paper, we propose a new optimal leverage classifier based on linear SVM under a nonseparable setting. Our classifier aims to select an informative subset of the training sample to reduce data size, enabling efficient computation while maintaining high accuracy. We take a novel view of SVM under the general subsampling framework and rigorously investigate the statistical properties. We propose a two-step subsampling procedure consisting of a pilot estimation of the optimal subsampling probabilities and a subsampling step to construct the classifier. We develop a new Bahadur representation of the SVM coefficients and derive unconditional asymptotic distribution and optimal subsampling probabilities without giving the full sample. Numerical results demonstrate that our classifiers outperform the existing methods in terms of estimation, computation, and prediction.

*Keywords and phrases:* Classification; Large-scale dataset; Martingale; Optimal subsampling; Support vector machine.

## 1. Introduction

Consider the binary classification problem for a training sample of size $N$, $\mathcal{D}_N = \{(\boldsymbol{X}_j, Y_j)\}_{j=1}^N$, where $\boldsymbol{X}_j \in \mathbb{R}^p$ denotes covariates (a.k.a.features), $Y_j = \{1, -1\}$ represents class labels. The central task is to build a classifier that predicts the label based on the observed covariates. Numerous literature is available on binary classification procedures, including nearest neighbor classifiers, discriminant analysis, logistic regression, tree-based methods, support vector machine, and ensemble learning. See, for example, Hastie et al. (2010); Fan et al. (2020) for a comprehensive review.

Support vector machine (SVM) is a theoretically motivated classifier and has gained significant popularity in various applications (Boser et al., 1992; Cortes and Vapnik, 1995; Vapnik, 2013). As a margin-based approach, SVM aims to find the maximum-margin hyperplane in either the original or extended kernel feature space. According to the elegant geometric interpretation, only a subset of the training dataset called the *support vectors*, needs to be considered for evaluating the separating hyperplane. This property is attractive compared to likelihood-based classifiers, such as logistic regression, which depend on all training data to determine the discriminative boundary. Moreover, logistic regression is typically fitted under the assumption that the response follows a binomial distribution, whereas SVM does not require any distributional assumption and thus leads to more robust performance (Steinwart and Christmann, 2008).

Despite the advantages mentioned above, constructing an SVM classifier is com-

Corresponding author: pingma@uga.edu (Ping Ma) putationally intensive as it typically involves solving quadratic programming optimization problems. In general, the computational cost of SVM is $O(N^2 N_s)$ (Kaufman, 1998), where $N_s$ represents the number of support vectors. In practice, $N_s$ usually increases linearly with the sample size $N$ of the training data. As a result, the number of support vectors significantly affects the training time and the evaluation of the decision boundary. Various methods have been proposed to mitigate the computational complexity of training SVM classifiers. For example, specialized algorithms for solving quadratic programming have been suggested, including the sequential minimal optimization (Platt, 1998) and various decomposition methods used in the LibLinear software library (Hsieh et al., 2008). Other fast computation methods based on low-rank approximation (Williams and Seeger, 2000), gradient descent (Bordes et al., 2005; Shalev-Shwartz et al., 2011; Wang et al., 2012), core set (Tsang et al., 2005), and nearest neighbor (Camelo et al., 2015) have also been developed. However, it is worth noting that most of these methods still incur a computational cost of at least $O(N^2)$ or lack optimal statistical guarantees. Therefore, when the sample size of the training data is huge, both time complexity and statistical guarantees become prohibitively demanding.

Observing that the discriminative boundary of the SVM depends on only a subset of the training data, we take another look at the SVM from the perspective of data reduction. A crucial insight from the SVM is that a relatively small subset of the training data is sufficient to build up an effective classifier. Inspired by leverage score sampling methods developed for least-squares regression (Drineas et al., 2011;

Ma et al., 2015b) and low-rank matrix approximation (Mahoney and Drineas, 2009), our strategy is to construct an importance sampling distribution for all the training data points, which effectively reduces the data size before constructing the classifier. The nonuniform subsampling strategy we employ is straightforward to design and implement. As long as the reduced dataset remains informative or representative, the corresponding estimator can provide a satisfactory approximation to the estimator based on the full sample. For example, the statistical leveraging framework (Drineas et al., 2012; Ma et al., 2015b, 2022; Li and Meng, 2020) has achieved great success in large-scale ordinary least squares regression. More recently, optimal subsampling procedures have been also established for various statistical models, including logistic regression (Wang et al., 2018), generalized linear models (Ai et al., 2018; Yu et al., 2022), quantile regression (Wang and Ma, 2021), nonparametric regression (Ma et al., 2015a; Meng et al., 2020, 2021), and designed for testing problems (Ren et al., 2022; Han et al., 2023). However, none of the existing can be directly applied to SVM due to its distinguishing geometric feature. Consequently, our goal is to develop a leverage classifier that is computationally efficient for large datasets and theoretically provable as the SVM.

In this paper, we introduce a novel binary classifier based on linear SVM in a non-separable setting. To construct the optimal classifier, we propose a two-step optimal subsampling algorithm that involves a pilot study to estimate the optimal subsampling probabilities and a subsampling step. Our subsampling procedure significantly reduces the computational costs without scarfing too much estimation efficiency. With a novel view of the SVM under the general subsampling framework, we rigorously investigate the statistical properties of the proposed classifier. Specifically, we derive the asymptotic distribution and the optimal subsampling probabilities. Our contributions can be summarized as follows:

(1) Double randomnesses are addressed: one arising from the training data and the other from the subsampling procedure. Our approach yields an unconditional asymptotic result regardless of the full sample and thus allows for random subsampling probabilities.

(2) We utilize the martingale technique as observations in the selected samples are no longer independent. Our theoretical framework builds upon the Bahadur representation of the linear SVM estimator, which is nonstandard in the context of the general subsampling strategy.

(3) The nonuniform subsampling probabilities are computed by minimizing specific criteria derived from the asymptotic variance, leading to optimality within the experimental design theory. Numerical results also demonstrate that our leverage classifier is computationally fast, and the identified separating hyperplane is close to that obtained using the full sample SVM.

The remainder of this paper is organized as follows. Section 2 reviews the linear SVM for nonseparable binary classification and motivates the leverage classifier framework. Section 3 investigates the theoretical properties of leverage classifiers and develops efficient algorithms for constructing optimal leverage classifiers. Simulation studies and a real-world example are presented in Sections 4–5. Section 6 concludes the paper with some potential improvements. All theoretical proofs and additional numerical results are provided in the Supplementary Material. The implementing codes for this work are available in https://github.com/yuxiaohaihyx0517/Leverage-Classifier.

## 2. Support vector machine and leverage classifier

### 2.1 Support vector machine

Binary linear classification problem aims to find the best separating hyperplane of the form $f(\boldsymbol{X}, \boldsymbol{\beta}) = \beta_0 + \boldsymbol{X}^\top \boldsymbol{\beta}_1$, with intercept $\beta_0$ and slope vector $\boldsymbol{\beta}_1$. Write $\boldsymbol{\beta} = \left(\beta_0, \boldsymbol{\beta}_1^\top\right)^\top \in \mathbb{R}^{p+1}$ and $\widetilde{\boldsymbol{X}} = \left(1, \boldsymbol{X}^\top\right)^\top \in \mathbb{R}^{p+1}$ as the augmented parameter and data vectors, and then $f(\boldsymbol{X}, \boldsymbol{\beta}) = \widetilde{\boldsymbol{X}}^\top \boldsymbol{\beta}$. When the training data are not linearly separable, the linear SVM solves the following optimization problem

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \left\{ \frac{1}{N} \sum_{j=1}^{N} \left[1 - Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta})\right]_+ + \frac{\lambda_{\mathrm{FULL}}}{2} \|\boldsymbol{\beta}_1\|^2 \right\}, \tag{2.1}$$

where $[u]_+ = \max(u, 0)$ is the hinge loss function, $\|\cdot\|$ denotes the Euclidean norm of a vector, and the tuning parameter $\lambda_{\mathrm{FULL}} > 0$ controls the amount of regularization on model complexity.

From the theoretical perspective, Koo et al. (2008) investigated the asymptotic behavior of the coefficient of the linear SVM. Denote the population version of the loss function in (2.1) without penalty by $L(\boldsymbol{\beta}) = \mathbb{E}\left[1 - Y f(\boldsymbol{X}, \boldsymbol{\beta})\right]_+$, and its minimizer $\boldsymbol{\beta}^\dagger = \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{\beta})$. Define

$$\boldsymbol{S}(\boldsymbol{\beta}) = -\mathbb{E}\left\{\mathbb{I}\left(Y f(\boldsymbol{X}, \boldsymbol{\beta}) \leq 1\right) Y \widetilde{\boldsymbol{X}}\right\}, \quad \mathbf{H}(\boldsymbol{\beta}) = \mathbb{E}\left\{\psi\left(1 - Y f(\boldsymbol{X}, \boldsymbol{\beta})\right) \widetilde{\boldsymbol{X}} \widetilde{\boldsymbol{X}}^\top\right\},$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\psi(\cdot)$ is the Dirac delta function. Provided that $\boldsymbol{S}(\boldsymbol{\beta})$ and $\mathbf{H}(\boldsymbol{\beta})$ are well-defined (Koo et al., 2008), they are interpreted as the gradient and Hessian matrix of $L(\boldsymbol{\beta})$. Subsequently, under regularity conditions, $\widehat{\boldsymbol{\beta}}$ satisfies

$$\sqrt{N}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{\dagger}) \to \mathcal{N}\left(\mathbf{0}, \mathbf{H}(\boldsymbol{\beta}^{\dagger})^{-1}\mathbb{E}\{\mathbb{I}(Yf(\boldsymbol{X}, \boldsymbol{\beta}^{\dagger}) \leq 1)\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^{\top}\}\mathbf{H}(\boldsymbol{\beta}^{\dagger})^{-1}\right). \tag{2.2}$$

From an optimization perspective, the representer theorem (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001) states that the solution to the quadratic programming in (2.1) admits a finite-dimensional expression of basis functions. In general, solving a quadratic programming optimization problem has a computational cost of $O(N^3)$ (Mehrotra, 1992; Chang, 2011), which becomes prohibitively expensive when the training data size $N$ is large. However, in the case of the linear SVM, a significant fraction of the basis coefficients can be zero. The training data associated with the nonzero basis coefficients are called support vectors, which play a crucial role in determining the discriminative boundary. As a result, the computational cost is significantly reduced as the number of support vectors is much smaller than the training sample size, making it more feasible for large-scale datasets.

### 2.2 Leverage classifier

Inspired by the appealing property of support vectors, we revisit the SVM and develop a new classifier called leverage classifier. Our strategy first selects an informative subset of the training data with some nonuniform subsampling probabilities and then constructs the linear SVM classifier based on the reduced dataset. The leverage classifier integrates leverage score sampling with the margin-based classifier, and its advantage is to approximate the discriminative boundary well with significantly reduced computational cost. In our subsampling framework, we employ the subsampling with replacement strategy to ensure theoretical convenience. The detailed procedure of the leverage classifier is described in Algorithm 1.

**Algorithm 1** Leverage classifier.

**Step 1** Assign subsampling probabilities $\boldsymbol{\pi} = \{\boldsymbol{\pi}_j\}_{j=1}^N$ to all training samples in $\mathcal{D}_N$;

**Step 2** Draw a subset of size $n \ll N$ from $\mathcal{D}_N$ according to $\boldsymbol{\pi}$ via subsampling with replacement. Denote the subsample by $\mathcal{S}_n = \{(\boldsymbol{X}_i^*, Y_i^*)\}_{i=1}^n$ and the corresponding subsampling probabilities by $\boldsymbol{\pi}^* = \{\boldsymbol{\pi}_i^*\}_{i=1}^n$;

**Step 3** Use $\mathcal{S}_n$ to train the linear SVM by minimizing the penalized weighted hinge loss with a properly tuned parameter $\lambda$

$$\widetilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min}\left\{\frac{1}{n}\sum_{i=1}^{n}\frac{[1 - Y_i^* f(\boldsymbol{X}_i^*, \boldsymbol{\beta})]_+}{N\boldsymbol{\pi}_i^*} + \frac{\lambda}{2}\|\boldsymbol{\beta}_1\|^2\right\}.$$

**Step 4** The separating hyperplane is $f(\boldsymbol{X}, \widetilde{\boldsymbol{\beta}}) = \widetilde{\boldsymbol{X}}^\top\widetilde{\boldsymbol{\beta}}$.

The performance of the leverage classifier relies on the subsampling probability $\boldsymbol{\pi}$, the subsample size $n$, and the tuning parameter $\lambda$. First, the reduced dataset $\mathcal{S}_n$ is obtained according to $\boldsymbol{\pi}$. A simple choice, $\pi_j = N^{-1}$, leads to uniform subsampling. Although this strategy is useful for exploratory data analysis, it often fails to extract important information by ignoring the distinctive characteristics of statistical models. Recent studies on logistic regression (Wang et al., 2018) and quantile regression (Wang and Ma, 2021) have highlighted the importance of designing nonuniform subsampling strategies. Our subsequent analysis reveals that the leverage classifier, with carefully designed $\boldsymbol{\pi}$, can attain a certain level of optimality in terms of experimental design. Second, Kaufman (1998) pointed out that the number of support vectors typically increases linearly with the training sample size. As a result, the leverage classifier with $\mathcal{S}_n$ of size $n$ offers a more efficient computational approach compared to the SVM utilizing the full sample size $N$. Lastly, training the leverage classifier involves tuning parameter selection, which differs from the aforementioned literature. We employ the Generalized Approximate Cross-Validation method (GACV). Specifically, minimize objective function $N^{-1}\sum_{k=1}^{N}[1-Y_k f_\lambda^{[-k]}(\boldsymbol{X}_k,\boldsymbol{\beta})]_+$, where $f_\lambda^{[-k]}(\boldsymbol{X}_k,\boldsymbol{\beta})$ is the SVM solution with $k$-th data point removed. This objective function stems from the penalized likelihood estimates in SVM and serves as a generalization of the generalized cross-validation. GACV does not need to train and test every possible hyperparameter combination and thus is a computationally efficient method. See Wahba et al. (2003) for its optimal properties and implementation details.

Before proceeding with theoretical analysis, we provide a toy example to illustrate the intuition of the leverage classifier. Please refer to Section 4 for the implementation details. In Figure 1, the right panel showcases the best separating hyperplane determined solely by the support vectors associated with the full sample SVM. The left panel displays the leverage classifier with A-optimality (explained in Section 3), which tends to select data points close to the full sample support vectors, resulting in a reduced dataset that is informative in identifying the discriminative boundary. In contrast, the middle panel demonstrates the uniform subsampling strategy, which overlooks the characteristics of the full sample support vectors. As a result, the selected subsample is less informative. Unless the subsample size $n$ is relatively large, the uniform subsampling strategy will be inferior to a carefully designed nonuniform subsampling strategy used by the leverage classifier.

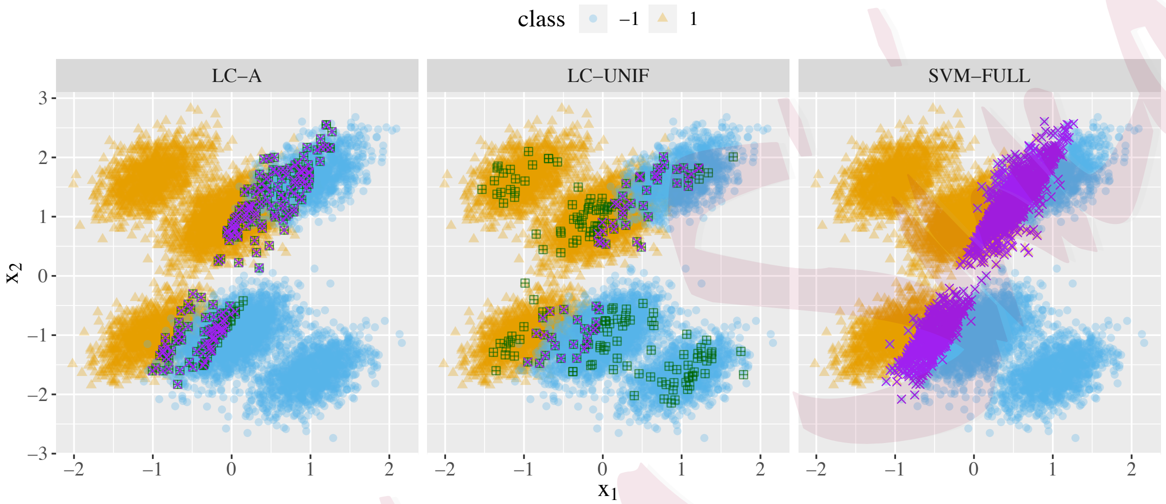

Figure 1: Toy example for linear classification. Classifiers are the proposed optimal leverage classifier with A-optimality (LC-A), the leverage classifier with uniform subsampling (LC-UNIF), and the full sample linear SVM (SVM-FULL). The green ⊞'s denote the selected subsamples, and the purple ×'s denote the support vectors.

## 3. Theoretical properties and optimal leverage classifier

In this section, we establish theoretical properties and provide an efficient algorithm for the proposed leverage classifiers under the subsampling framework.

### 3.1 Asymptotic normality

**Assumption 1.** *The conditional densities of $\boldsymbol{X}$ given class $Y = 1$ and $Y = -1$ with respect to the Lebesgue measure are continuous and have finite fourth moments.*

**Assumption 2.** *The covariates for the two classes have different mean values in at least one dimension.*

**Assumption 3.** *The nonzero minimizer $\boldsymbol{\beta}^\dagger$ of $L(\boldsymbol{\beta})$ is unique and satisfies that $S(\boldsymbol{\beta}^\dagger) = 0$. $\mathbf{H}(\boldsymbol{\beta})$ is positive-defined around $\boldsymbol{\beta}^\dagger$ in a compact set $\mathcal{B}$ with a nonzero radius.*

**Assumption 4.** *The subsampling probabilities satisfy that*

$$\frac{1}{N^3}\sum_{j=1}^{N}\mathbb{E}\left(\frac{1}{\pi_j^2}\right) = O(1).$$

Assumptions 1–3 are commonly imposed to establish the asymptotic normality of the linear SVM, and they typically hold under the regularity conditions outlined in Koo et al. (2008). Assumption 4 allows for random subsampling probabilities since the full dataset is not fixed. Furthermore, Assumption 4 restricts $\boldsymbol{\pi}$ from being extremely small, preventing any training sample from dominating the weighted penalized hinge loss function in Step 3 of Algorithm 1. When we condition on the full dataset, Assumption 4 is in the similar spirit of the commonly used subsampling schemes, for example, Ai et al. (2018); Wang et al. (2018).

**Theorem 1** (The Bahadur representation)**.** *Suppose Assumptions 1–4 hold. For $\lambda = o(n^{-1/2})$, we have*

$$\sqrt{n}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger) = -\frac{1}{\sqrt{n}}\mathbf{H}(\boldsymbol{\beta}^\dagger)^{-1}\sum_{i=1}^{n}\frac{1}{N\pi_i^*}\xi_i^* Y_i^* \widetilde{\boldsymbol{X}}_i^* + o_P(1), \tag{3.1}$$

*where $\xi_i^* = \mathbb{I}\left(Y_i^* f(\boldsymbol{X}_i^*, \boldsymbol{\beta}^\dagger) \le 1\right)$ and $\widetilde{\boldsymbol{X}}_i^* = \left(1, \boldsymbol{X}_i^{*\top}\right)^\top$, $i = 1, \ldots, n$.*

Theorem 1 presents a Bahadur representation of $\widetilde{\boldsymbol{\beta}}$ for the leverage classifier under the subsampling framework, which is the building block for establishing the asymptotic normality. As discussed in (Koo et al., 2008), the condition $\lambda = o(n^{-1/2})$ is an appropriate order for nonseparable SVM, and additional simulation results confirm the rationality of this condition. The use of subsampling with replacement and the integration of the subsampling probability makes Theorem 1 a nontrivial extension of Koo et al. (2008), which only considered SVMs learned from independent and identically distributed data. The Bahadur representation reveals how the subsampling strategy and margins of the optimal separating hyperplane determine the statistical behavior of the estimator.

Next, we establish the unconditional asymptotic normality of $\widetilde{\boldsymbol{\beta}}$ based on the Bahadur representation. To this end, we define $\boldsymbol{T} = n^{-1}\sum_{i=1}^{n}(N\pi_i^*)^{-1}\xi_i^* Y_i^* \widetilde{\boldsymbol{X}}_i^*$ as a term on the right hand side of (3.1). As Algorithm 1 conducts subsampling with replacement, the data in the reduced dataset $\mathcal{S}_n$ are no longer independent unless conditioned on the full training sample. Hence, we treat the subsampling procedure as a stochastic process and employ the martingale technique to study the asymptotic property of $\boldsymbol{T}$. Let $\boldsymbol{X}_1^N = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N)$ and $Y_1^N = (Y_1, \ldots, Y_N)$. Step 2 in Algorithm 1 can be viewed as a $n$-step sequential sampling procedure: in the $i$-th step, we select one data point with replacement from the full training sample and denote it by $(\boldsymbol{X}_i^*, Y_i^*)$. Let $\sigma(*_i)$ be the $\sigma$-algebra (Durrett, 2019) generated by the $i$-th sampling step, which is closed under complement, countable unions, and countable intersections. Accordingly, we thus define a filtration as $\mathcal{F}_{N,0} = \sigma\left(\boldsymbol{X}_1^N, Y_1^N\right)$ and $\mathcal{F}_{N,i} = \sigma\left(\boldsymbol{X}_1^N, Y_1^N\right) \vee \sigma\left(*_1\right) \vee \cdots \vee \sigma\left(*_i\right)$

for $i = 1, \ldots, n$. This filtration $\mathcal{F}_{N,i}$ be explained as the smallest $\sigma$-algebra containing all the information after the $i$-th sampling step. Based on this filtration, we define $\boldsymbol{M} = \sum_{i=1}^{n} \boldsymbol{M}_i$, where

$$\boldsymbol{M}_i = \frac{1}{nN\pi_i^*}\xi_i^* Y_i^* \widetilde{\boldsymbol{X}}_i^* - \frac{1}{nN}\sum_{j=1}^{N} \xi_j Y_j \widetilde{\boldsymbol{X}}_j.$$

We can express $\boldsymbol{T} = \boldsymbol{M} + \boldsymbol{Q}$ with $\boldsymbol{Q} = N^{-1}\sum_{j=1}^{N} \xi_j Y_j \widetilde{\boldsymbol{X}}_j$, where above decomposition allows for decoupling the variabilities from the sampling procedure and the full dataset, which are measured by $\boldsymbol{M}$ and $\boldsymbol{Q}$, respectively. In the Supplementary Material, we demonstrate that $\{\boldsymbol{M}_i, i = 1, \ldots, n\}$ forms a martingale difference sequence adapted to filtration $\{\mathcal{F}_{n,i}, i = 1, \ldots, n\}$. Using the martingale central limit theorem (Ohlsson, 1989), we establish the unconditional asymptotic normality of $\widetilde{\boldsymbol{\beta}}$.

**Theorem 2** (Asymptotic normality)**.** *Suppose Assumptions 1–4 hold. Then the variance of* $\boldsymbol{T}$*, denoted by* $\mathbf{V}_T$*, can be written as*

$$\mathbf{V}_T = \frac{1}{nN^2}\sum_{j=1}^{N} \mathbb{E}_{Y|\boldsymbol{X}}\left(\frac{1}{\pi_j}\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right) \widetilde{\boldsymbol{X}}_j \widetilde{\boldsymbol{X}}_j^\top\right) + \mathbf{C},$$

*where* $\mathbf{C}$ *is a constant matrix that does not depend on* $\boldsymbol{\pi}$*. As* $N \to \infty$*,* $n \to \infty$*, we have*

$$\mathbf{V}^{-1/2}(\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^\dagger) \to \mathcal{N}(\mathbf{0}, \mathbf{I}_{p+1}),$$

*in distribution, where* $\mathbf{V} = \mathbf{H}(\boldsymbol{\beta}^\dagger)^{-1}\mathbf{V}_T\mathbf{H}(\boldsymbol{\beta}^\dagger)^{-1}$ *and* $\mathbf{I}_{p+1}$ *is the identity matrix of dimension* $p + 1$*.*

Theorem 2 typically allows for random $\boldsymbol{\pi}$ since the subsampling probabilities may depend on the response. When $\boldsymbol{\pi}$ is prespecified or does not depend on $Y$, the variance can be further simplified to $\mathbf{V}_T = (nN^2)^{-1}\sum_{j=1}^{N} \pi_j{}^{-1}\mathrm{P}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right) \widetilde{\boldsymbol{X}}_j \widetilde{\boldsymbol{X}}_j^\top + \mathbf{C}$. In this case, the subsampling procedure affects all the data points, making it impossible to identify the support vectors without any information about $Y$. Assumptions 4 and the moment condition in Assumption 1 are utilized to verify the martingale version of the Lindeberg-Feller conditions. In the proof of Theorem 2, we observe that the first term in $\mathbf{V}_T$ is derived from the variance of $\boldsymbol{M}$, while the second term $\mathbf{C}$ comes from $\boldsymbol{Q}$ and some terms in the variance of $\boldsymbol{M}$ that are independent of $\boldsymbol{\pi}$. In particular, when $n/N \to 0$, the variability from the full dataset is insignificant. This evokes us to determine optimal subsampling probabilities by minimizing certain criteria based on the first term of $\mathbf{V}_T$.

### 3.2 Optimal leverage classifier

The leverage classifier enables fast computation by using a reduced dataset $\mathcal{S}_n$. Take the uniform subsampling strategy with $\pi_j^{\mathrm{UNIF}} = N^{-1}$, $j = 1, \ldots, N$ as an example. Assumption 4 is satisfied, and thus the corresponding leverage classifier admits the asymptotic properties described in Theorems 1 and 2. However, the uniform subsampling procedure does not account for any statistical model assumption and may fail to capture the most informative sample points leading to unsatisfactory estimates; see Figure 1 for illustration.

We next explore how to determine the subsampling probabilities $\boldsymbol{\pi} = \{\pi_j\}_{j=1}^{N}$, by which the leverage classifier attains certain statistical optimality based on the asymptotic properties. A key observation is that in Theorem 2 the asymptotic variance matrix $\mathbf{V}$ is a function of the subsampling probabilities. It motivates us to derive nonuniform subsampling probabilities by minimizing some criterion associated with $\mathbf{V}$. To this end, we borrow the concepts from the design of experiments and consider A- and L-optimality criteria (Atkinson et al., 2007). Note that we expect the subsampling probabilities to satisfy Assumption 4 although it is not required in the following theorem. We will provide a fix for this issue shortly afterward.

**Theorem 3.** *When minimizing the traces of $\mathbf{V}$ and $\mathbf{V}_T$, two sets of optimal subsampling probabilities based on A- and L-optimality are*

$$\begin{aligned} \pi_j^{\mathrm{A}} &= \frac{\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right) \|\mathbf{H}(\boldsymbol{\beta}^\dagger)^{-1}\widetilde{\boldsymbol{X}}_j\|}{\sum_{k=1}^{N} \mathbb{I}\left(Y_k f(\boldsymbol{X}_k, \boldsymbol{\beta}^\dagger) \le 1\right) \|\mathbf{H}(\boldsymbol{\beta}^\dagger)^{-1}\widetilde{\boldsymbol{X}}_k\|}, \\ \pi_j^{\mathrm{L}} &= \frac{\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right) \|\widetilde{\boldsymbol{X}}_j\|}{\sum_{k=1}^{N} \mathbb{I}\left(Y_k f(\boldsymbol{X}_k, \boldsymbol{\beta}^\dagger) \le 1\right) \|\widetilde{\boldsymbol{X}}_k\|}, \end{aligned} \tag{3.2}$$

*where $j = 1, \ldots, N$. Correspondingly, the traces of $\mathbf{V}$ and $\mathbf{V}_T$ attain their minima.*

Theorem 3 takes an optimization approach to deriving the subsampling probabilities by minimizing the traces of $\mathbf{V}$ and $\mathbf{V}_T$ in Theorem 2, respectively. The indicator functions $\mathbb{I}(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1)$ in (3.2) are related to the definition of support vectors, implying that the leverage classifier inherits the virtue of SVM. Moreover, this result differs substantially from the literature, e.g., Wang et al. (2018), which focuses on fixed subsampling probabilities by conditioning on the full dataset. The random response variable $Y_j$ enters into the expressions (3.2) via $\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right)$. Given the full dataset, our result will degenerate to fix subsampling probabilities.

Two issues arise when applying the subsampling probabilities (3.2) in practice. First, several population quantities, including the true parameter $\boldsymbol{\beta}^\dagger$, the Hessian matrix $\mathbf{H}(\boldsymbol{\beta}^\dagger)$, and the indicator function $\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1\right)$, need to be estimated. Second, the appearance of indicator functions in (3.2) may lead to a breakdown of Assumption 4. To address them, we propose to conduct *a pilot* study and substitute the unknown population quantities with their corresponding pilot estimates; and apply *an additional thresholding* to the indicator functions.

Specifically, for the pilot study, we select a pilot sample $\mathcal{S}_0 = \{(\boldsymbol{X}_{i0}^*, Y_{i0}^*)\}_{i=1}^{n_0}$ with some proper probabilities $\boldsymbol{\pi}_0^* = \{\pi_{i0}^*\}_{i=1}^{n_0}$ from $\mathcal{D}_N$, for instance, using a simple uniform subsampling procedure. We can then replace the true value of $\boldsymbol{\beta}^\dagger$ with the pilot estimator $\widetilde{\boldsymbol{\beta}}^0$. Moreover, the Hessian matrix can be estimated using a nonparametric method, as suggested by Koo et al. (2008),

$$\widetilde{\mathbf{H}}(\widetilde{\boldsymbol{\beta}}^0) = \frac{1}{n_0} \sum_{i=1}^{n_0} \frac{1}{N\pi_{i0}^*} K_h\left(1 - Y_{i0}^* f(\boldsymbol{X}_{i0}^*, \widetilde{\boldsymbol{\beta}}^0)\right) \widetilde{\boldsymbol{X}}_{i0} \widetilde{\boldsymbol{X}}_{i0}^\top, \tag{3.3}$$

where $K_h(t) = K(t/h)/h$ with bandwidth $h \to 0$ and the kernel function $K(\cdot)$ satisfying $K(t) \ge 0$ and $\int_{-\infty}^{\infty} K(t)\,\mathrm{d}t = 1$. The indicator $\mathbb{I}(Y_j f(\boldsymbol{X}_j, \boldsymbol{\beta}^\dagger) \le 1)$ can be replaced by $\mathbb{I}(Y_j f(\boldsymbol{X}_j, \widetilde{\boldsymbol{\beta}}^0) \le 1)$. For the additional thresholding for the indicator functions, we work under the level $\delta_N > 0$ such that

$$\begin{aligned}
\widehat{\pi}_j^{\mathrm{A}} &= \frac{\max\left\{\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \widetilde{\boldsymbol{\beta}}^0) \le 1\right) \|\widetilde{\mathbf{H}}(\widetilde{\boldsymbol{\beta}}^0)^{-1} \widetilde{\boldsymbol{X}}_j\|, \delta_N\right\}}{\sum_{k=1}^{N} \max\left\{\mathbb{I}\left(Y_k f(\boldsymbol{X}_k, \widetilde{\boldsymbol{\beta}}^0) \le 1\right) \|\widetilde{\mathbf{H}}(\widetilde{\boldsymbol{\beta}}^0)^{-1} \widetilde{\boldsymbol{X}}_k\|, \delta_N\right\}}, \\
\widehat{\pi}_j^{\mathrm{L}} &= \frac{\max\left\{\mathbb{I}\left(Y_j f(\boldsymbol{X}_j, \widetilde{\boldsymbol{\beta}}^0) \le 1\right) \|\widetilde{\boldsymbol{X}}_j\|, \delta_N\right\}}{\sum_{k=1}^{N} \max\left\{\mathbb{I}\left(Y_k f(\boldsymbol{X}_k, \widetilde{\boldsymbol{\beta}}^0) \le 1\right) \|\widetilde{\boldsymbol{X}}_k\|, \delta_N\right\}},
\end{aligned} \tag{3.4}$$

where $\widetilde{\boldsymbol{\beta}}^0$ is the pilot estimate of $\boldsymbol{\beta}^\dagger$, and $\delta_N$ is a user-specified constant. If we choose $\delta_N \propto N^{-1}$, the estimated subsampling probabilities (3.4) strike a balance between (3.2) and uniform subsampling probabilities. A simple calculation can verify that the estimated subsampling probabilities (3.4) meet Assumption 4, and the asymptotic results follow with $\boldsymbol{\pi}^*$ replaced by $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$. The two-step optimal leverage classifier is summarized in Algorithm 2.

**Algorithm 2** Optimal leverage classifier.

```
Step 1 Select $n_0$ pilot training samples $\mathcal{S}_0 = \{(\boldsymbol{X}_{i0}^*, Y_{i0}^*)\}_{i=1}^{n_0}$ with subsampling probabilities $\boldsymbol{\pi}_0^*$ from $\mathcal{D}_N$. Obtain the pilot estimates $\widetilde{\boldsymbol{\beta}}^0$ and $\widetilde{\mathbf{H}}(\widetilde{\boldsymbol{\beta}}^0)$;
Step 2 Calculate the optimal subsampling probabilities $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$ as in (3.4);
Step 3 Sample $n$ training samples as $\mathcal{S}_n = \{(\boldsymbol{X}_i^*, Y_i^*)\}_{i=1}^{n}$ with $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$ from $\mathcal{D}_N$;
Step 4 Implement Algorithm 1 with $\mathcal{S}_0 \cup \mathcal{S}_n$ and a proper tuning parameter $\lambda$ to obtain $\widetilde{\boldsymbol{\beta}}$ and the separating hyperplane $f(\boldsymbol{X}, \widetilde{\boldsymbol{\beta}}) = \widetilde{\boldsymbol{X}}^\top \widetilde{\boldsymbol{\beta}}$.
```

The choice of the pilot sample size $n_0$ involves a trade-off between estimation efficiency and computational complexity. A larger $n_0$ makes a more precise pilot estimate of $\boldsymbol{\beta}^\dagger$ and the Hessian matrix estimation which is estimated by the nonparametric method. However, the computational complexity of the pilot study should be negligible compared to those in Steps 3 and 4. Hence, we prefer a relatively small $n_0$; Please refer to the Supplementary Martial for a practical recommendation for $n_0$ with empirical evidence. Moreover, it is worth noting that the combination of $\mathcal{S}_0$ and $\mathcal{S}_n$ in Step 4 maximizes the utilization of selected samples for hyperplane estimation. To obtain the final subsampling estimate in Step 4, we tune $\lambda$ using the weighted version of GACV, which minimizes $n^{-1}\sum_{k=1}^{n} (N\boldsymbol{\pi}_k^*)^{-1}\left[1 - Y_k^* f_\lambda^{[-k]}(\boldsymbol{X}_k^*, \boldsymbol{\beta})\right]_+$.

The overall computational complexity of the optimal leverage classifier comprises three components. First, the cost of the pilot estimates is $O(n_0^3)$. Second, calculating the subsampling probabilities $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$ requires $O(N(p+1)^2)$ and $O(N(p+1))$, respectively. Third, constructing the the separating hyperplane $\widetilde{\boldsymbol{\beta}}$ with $\mathcal{S}_0 \cup \mathcal{S}_n$ takes $O\left((n+n_0)^3\right)$. In sum, the computational complexities of optimal leverage classifiers with $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$ are $O\left(n_0^3 + N(p+1)^2 + (n+n_0)^3\right)$ and $O\left(n_0^3 + N(p+1) + (n+n_0)^3\right)$, respectively. For extremely large $N$, the computational complexity is reduced to $O\left(N(p+1)^2\right)$ and $O\left(N(p+1)\right)$, which is linear in $N$. Compared with $O(N^3)$ for the full sample SVM, the optimal leverage classifier achieves fast computation with provable optimality.

We conclude with a discussion on the Fisher consistency of the leverage classifier. Fisher consistency is a desirable property of the loss function used by classifiers, that is, the population minimizer of the loss function leads to the Bayes optimal rule of classification (Lin, 2004). Lin et al. (2002) has shown that the hinge loss function used by the SVM satisfies Fisher consistency for classification. Under the framework of the leverage classifier as in Algorithms 1, it is clear that $\mathbb{E}\left([1-Y^* f(\boldsymbol{X}^*, \boldsymbol{\beta})]_+\right) = \mathbb{E}\left\{\mathbb{E}\left([1-Y_i^* f(\boldsymbol{X}_i^*, \boldsymbol{\beta})]_+ | \mathcal{D}_N\right)\right\} = \mathbb{E}\left([1-Y f(\boldsymbol{X}, \boldsymbol{\beta})]_+\right)$, which implies that the leverage classifier inherits the Fisher consistency from SVM.

## 4. Simulation Studies

In this section, we conduct extensive simulated experiments to demonstrate the numerical performance of our optimal leverage classifiers from the perspectives of estimation, prediction, and computation.

### 4.1 Settings

We generate a set of data points with covariate dimension $p = 8$ and randomly split them into two halves as training and testing datasets. The training dataset $\mathcal{D}_N$ is of size $N = 10^5$. The testing dataset is used to evaluate the prediction accuracy. We uniformly sample $n_0 = 500$ pilot samples for the pilot study. All simulation results are based on 500 replications. Table S1 in our Supplementary Material discusses the selection of bandwidth for Hessian matrix estimation and shows that the effect of different bandwidths can be ignorable. Therefore, we employ Silverman's rule of thumb (Silverman, 1986) to determine the appropriate bandwidth. We set the thresholding constant in (3.4) as $\delta_N = 0.01N^{-1}$. For a scalar $c$, write $\boldsymbol{c}_p = (c, \ldots, c)$ be the $p$-dimensional row vector of $c$'s. Four scenarios are considered:

(I) im-Uniform. The covariate $\boldsymbol{X}$ is independent and identically distributed from the uniform distribution. The $l$-coordinate of $\boldsymbol{X}$ is $U[0, 1]$ given $Y = 1$ and is $U[0.3, 1.3]$ given $Y = -1$, $l = 1, \ldots, p$. The proportions of data points for two classes are 80% and 20%. This is an imbalanced case.

(II) normMIX. The covariate $\boldsymbol{X}$ follows a mixture of three multivariate normal distributions with the same covariance matrix but different means. Let $\boldsymbol{X} \sim 0.5\mathcal{N}(\boldsymbol{\mu}_{11}, \boldsymbol{\Sigma}) + 0.25\mathcal{N}(\boldsymbol{\mu}_{12}, \boldsymbol{\Sigma}) + 0.25\mathcal{N}(\boldsymbol{\mu}_{13}, \boldsymbol{\Sigma})$ given $Y = 1$, $\boldsymbol{X} \sim 0.5\mathcal{N}(\boldsymbol{\mu}_{-11}, \boldsymbol{\Sigma}) + 0.25\mathcal{N}(\boldsymbol{\mu}_{-12}, \boldsymbol{\Sigma}) + 0.25\mathcal{N}(\boldsymbol{\mu}_{-13}, \boldsymbol{\Sigma})$ given $Y = -1$, where $\boldsymbol{\mu}_{11} = (\mathbf{0}_{p/2}, \mathbf{3}_{p/2})^\top$, $\boldsymbol{\mu}_{12} = (-\mathbf{3}_{p/2}, \mathbf{5}_{p/2})^\top$, $\boldsymbol{\mu}_{13} = -\mathbf{3}_p^\top$, $\boldsymbol{\mu}_{-11} = (\mathbf{0}_{p/2}, -\mathbf{3}_{p/2})^\top$, $\boldsymbol{\mu}_{-12} = (\mathbf{3}_{p/2}, -\mathbf{5}_{p/2})^\top$, and $\boldsymbol{\mu}_{-13} = (\mathbf{3}_{p/2}, \mathbf{5}_{p/2})^\top$. The proportions of two classes are equal to 50%.

(III) T3. The covariate $\boldsymbol{X}$ follows a multivariate $t(3)$ distribution with different means. Let $\boldsymbol{X} \sim t_3\left(\boldsymbol{\mu}_1, \mathbf{I}_p\right)/10$ given $Y = 1$ and $\boldsymbol{X} \sim t_3\left(\boldsymbol{\mu}_{-1}, \mathbf{I}_p\right)/10$ given $Y = -1$ , where $\boldsymbol{\mu}_1 = \mathbf{0.75}_p$, $\boldsymbol{\mu}_{-1} = -\mathbf{0.75}_p$. The proportions of two classes are equal to 50%.

(IV) T3MIX. The covariate $\boldsymbol{X}$ follows a mixture of two multivariate $t(3)$ distributions with different means. Let $\boldsymbol{X} \sim 0.3t_3\left(\boldsymbol{\mu}_{11}, \mathbf{I}_p\right) + 0.7t_3\left(\boldsymbol{\mu}_{12}, \mathbf{I}_p\right)$ given $Y = 1$ and $\boldsymbol{X} \sim 0.4t_3\left(\boldsymbol{\mu}_{-11}, \mathbf{I}_p\right)+0.6t_3\left(\boldsymbol{\mu}_{-12}, \mathbf{I}_p\right)$ given $Y = -1$, where $\boldsymbol{\mu}_{11} = \mathbf{2}_p^\top$, $\boldsymbol{\mu}_{12} = -\mathbf{3}_p^\top$, $\boldsymbol{\mu}_{-11} = -\mathbf{1}_p^\top$, $\boldsymbol{\mu}_{-12} = \mathbf{8}_p^\top$. The proportions of two classes are equal to 50%.

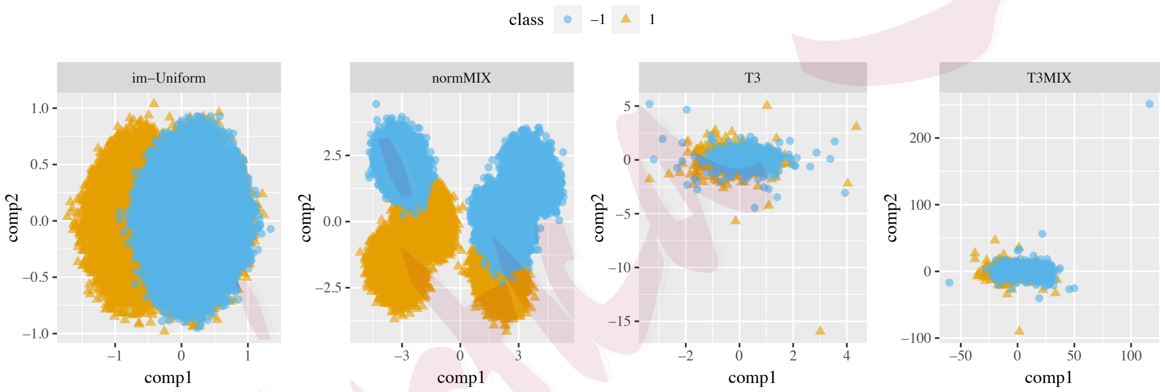

Figure 2: Full dataset visualization with principal component analysis under Scenarios I–IV.

We first project the full datasets of Scenarios I–IV into their first two principal components in Figure 2 to make an intuitive visualization. Besides the optimal leverage classifiers, we also consider Algorithm 1 with $n + n_0$ subsamples uniformly sampled from the training set and the full sample SVM, termed as LC-UNIF and SVM-FULL, respectively.

### 4.2 Results

To assess the estimation performance in approximating the full sample SVM, we calculate the mean squared error of $\widetilde{\boldsymbol{\beta}}$ on training set from $B = 500$ replications as $\text{MSE}(\widetilde{\boldsymbol{\beta}}) = B^{-1} \sum_{b=1}^{B} \|\widetilde{\boldsymbol{\beta}}^{(b)} - \widehat{\boldsymbol{\beta}}\|^2$, where $\widetilde{\boldsymbol{\beta}}^{(b)}$ is the estimator obtained from the $b$-th replication, and $\widehat{\boldsymbol{\beta}}$ is the estimator of the full sample SVM.

Figure 3 investigates the effect of subsample size on the estimation performance. Across all simulation scenarios, the optimal leverage classifiers outperform those with uniform subsampling, which aligns with our theoretical analysis in Theorem 3. The leverage classifier with A-optimal subsampling probabilities performs slightly better than that with L-optimality since A-optimality captures more sample information via the Hessian matrix. Moreover, the proposed methods outperform the leverage classifier with uniform subsampling under Scenario III (T3) and Scenario IV (T3MIX), where the heavy-tail distribution violates the moment assumption in Theorem 2. As our method is designed to identify points close to the classification hyperplane, it is expected to be robust to outliers. Under the imbalanced case in Scenario I, the optimal leverage classifiers also perform well. Additional simulations in Supplementary Material demonstrate that our method is not sensitive to the pilot sample size $n_0$. Then, we practically recommend the ratio $n_0/(n + n_0)$ to be around $(0.2, 0.4)$.

Figure 4 indicates that all methods approach the performance of the full sample SVM as $n$ increases. Remarkably, our optimal leverage classifier sometimes outperforms the full sample SVM in terms of prediction accuracy, as observed in Scenario IV. When $n$ is relatively small, our optimal leverage classifiers exhibit higher prediction accuracy

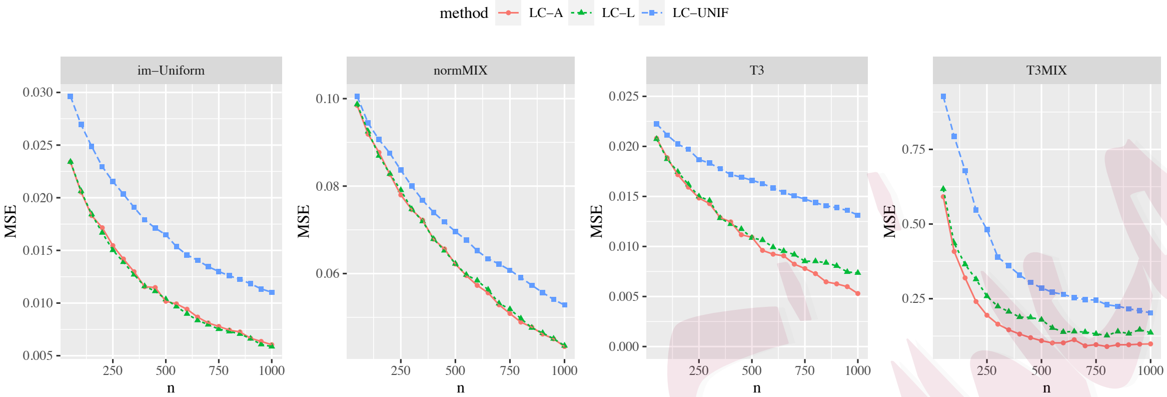

Figure 3: Comparison of MSE for approximating the full sample SVM estimator $\widehat{\boldsymbol{\beta}}$ against different subsample sizes under Scenarios I–IV.

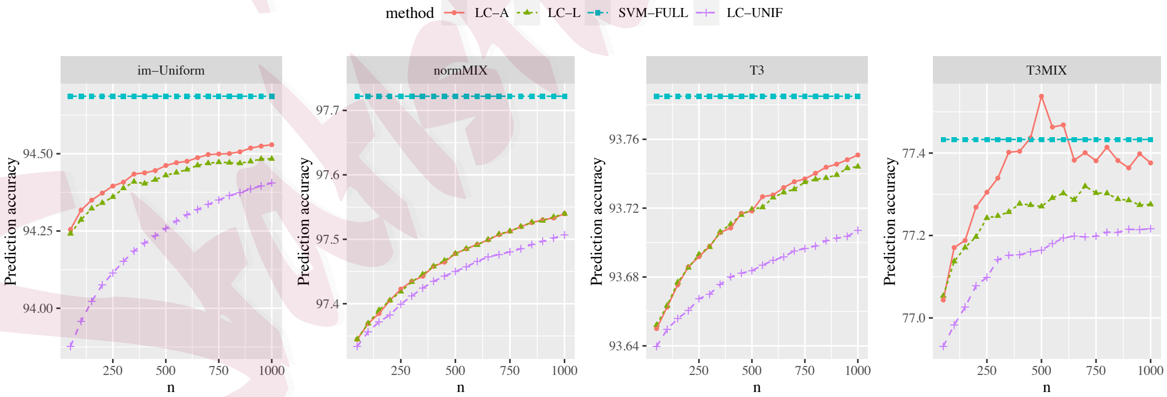

Figure 4: Comparison of prediction accuracy (%) against different subsample sizes under Scenarios I–IV.

than uniform subsampling, even in scenarios with heavy-tail covariate distribution and imbalanced classes. In addition, as pointed out by a reviewer, constructing classifiers using the support vectors from the pilot sample degenerates to the special case with $n = 0$ of LC-UNIF, which is typically challenging to outperform our optimal classifiers due to the larger subsample size $n$ and optimal subsampling probability $\boldsymbol{\pi}$ utilized in our approach.

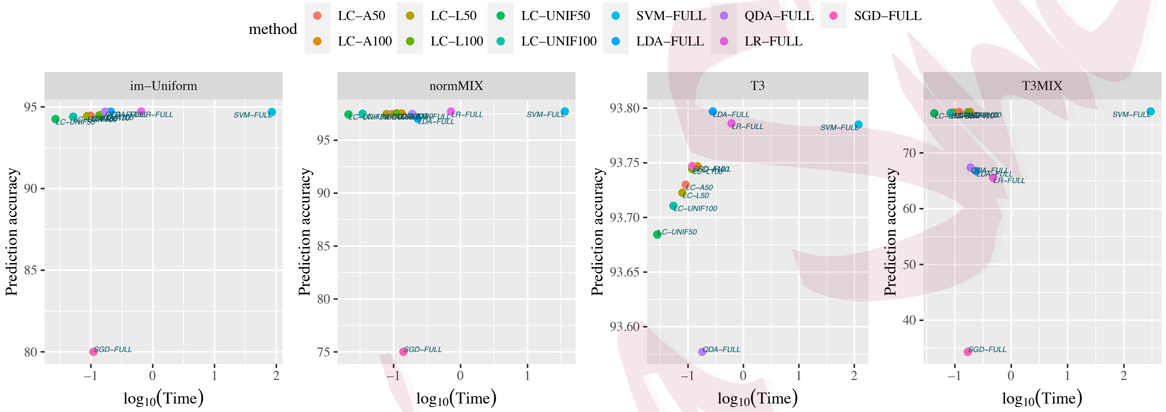

Figure 5: Comparison of prediction accuracy (%) and training time for several classifiers against different subsample sizes under Scenarios I–IV. The logarithm is taken on time for a better presentation of the figures.

Next, we compare the leverage classifiers with several benchmark classifiers, including logistic regression (LR), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and fast stochastic gradient descent (SGD), in terms of training time and prediction accuracy. All four competitors are trained based on the full dataset. Figure 5 elaborates that the optimal leverage classifiers achieve higher prediction accuracy with similar computing time under most scenarios. Compared to the full data approach, the proposed method yields significant computational time savings without sacrificing much accuracy. This aligns with our theoretical results that the convergence rate is only $O(\sqrt{N})$ while the computational cost is $O(N^3)$. In particular, leverage classifiers are more robust than logistic regression since the SVM only depends on the support vectors, while logistic regression is related to the likelihood of the full dataset. Linear discriminant analysis and quadratic discriminant analysis may work well because they are model-based classifiers requiring Gaussian distribution assumption. Stochastic gradient descent algorithm can significantly reduce computational resources for large-scale datasets or online datastreams, but each iteration is updated by random sampling, which may lead to the loss of informative data points, and affect accuracy, particularly in imbalanced and mixed settings. In Scenario IV, the prediction accuracy of our classifiers is about 10% higher than others. Overall, it is promising that the leverage classifiers using a reduced dataset can outperform some classifiers using the full sample.

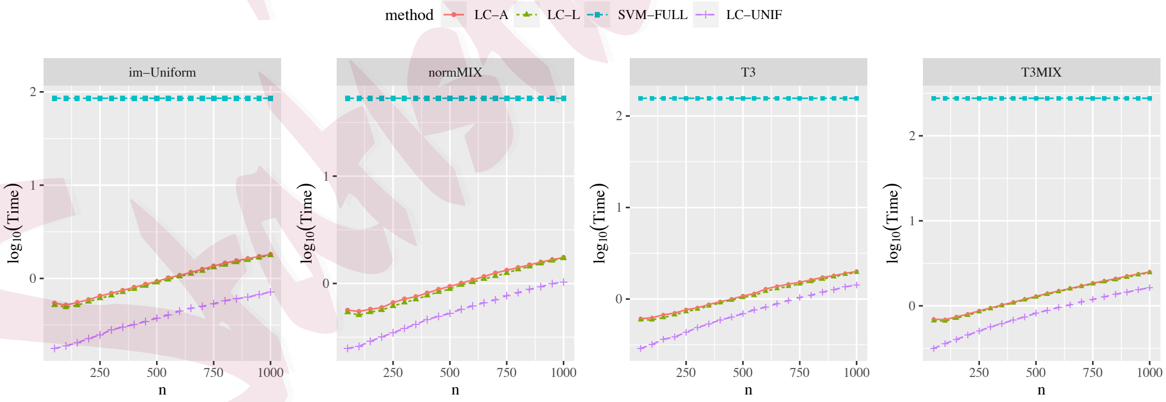

Figure 6: Comparison of CPU time (in seconds) against different subsample sizes under Scenarios I–IV. The logarithm is taken on time for a better presentation of the figures.

To validate the computational benefit of the leverage classifiers for large datasets, we further record the average computing time for each method during 500 replications. We use the fast R package `LiblineaR` to fit the full sample SVM. Figure 6 illustrates that the computing time of the full sample SVM is significantly larger than all leverage classifiers, as expected. our optimal leverage classifiers require slightly more time than uniform subsampling, this is due to the additional pilot study required to determine subsampling probabilities. Moreover, due to additional calculations with the Hessian matrix in A-optimality, the L-optimal subsampling probabilities take less computing time than A-optimality, which is consistent with our computational complexity analysis in Section 3.2. Figure 3 and Figure 6 both show that increasing $n$ leads to smaller MSE but also requires more computing time. The trade-off between estimation efficiency and computational efficiency actually affect by the practitioners' resource constraints and efficiency requirements, such as measurement cost, processing time, memory capacity, and prediction accuracy. We also report the computing time via one replication for different full sample sizes under Scenario I in Table 1. The computational advantage of leverage classifiers becomes significant as $N$ increases.

## 5. Real Data Analysis

Protein structure prediction is a critical challenge in computational biology (Lesk, 2019), and SVM has been a popular method for this task. However, the high computational cost associated with SVM has limited its widespread applications in this field. To this end, we examine the performance of our leverage classifier in protein structure prediction using the "Physicochemical Properties of Protein Tertiary Struc-

Table 1: Comparison of CPU time (in seconds) under Scenario I when $n = 1000$.

| Method | $N$ | | | | |
|---|---|---|---|---|---|
| | $10^3$ | $10^4$ | $10^5$ | $10^6$ | $10^7$ |
| **LC-A** | 1.32 | 1.33 | 1.75 | 1.85 | 3.82 |
| **LC-L** | 1.32 | 1.29 | 1.48 | 1.56 | 2.62 |
| **LC-UNIF** | 0.29 | 0.50 | 0.53 | 0.64 | 0.69 |
| **SVM-FULL** | 0.08 | 0.65 | 9.43 | 240.48 | 2526.90 |

ture Dataset”. This dataset is taken from the critical assessment of protein structure prediction (CASP) experiments and includes 45,730 decoys with nine covariates. More details are available at the UCI machine learning repository (Dua and Graff, 2017).

Root mean squared deviation (RMSD) is widely used as a metric for measuring the deviation of protein structures from their native protein structures (Iraji and Ameri, 2016). In this analysis, our goal is to construct a classifier and predict whether the root mean squared deviation is greater than ten or not. This setting leads to the proportions of two classes about 40% and 60%. Before applying our methods, we standardize each input variable with mean zero and standard deviation one, and then visualize it shown in the left panel of Figure 7. We randomly select half of the dataset as the training set and leave the rest as the testing set for prediction. Uniformly choose $n_0 = 500$ pilot subsamples from the training set to obtain the subsampling probabilities $\widehat{\boldsymbol{\pi}}^{\mathrm{A}}$ and $\widehat{\boldsymbol{\pi}}^{\mathrm{L}}$.

In Table 2, our optimal leverage classifiers are significantly faster than the full sample SVM which is implemented by the fast R package `LiblineaR`. This phenomenon

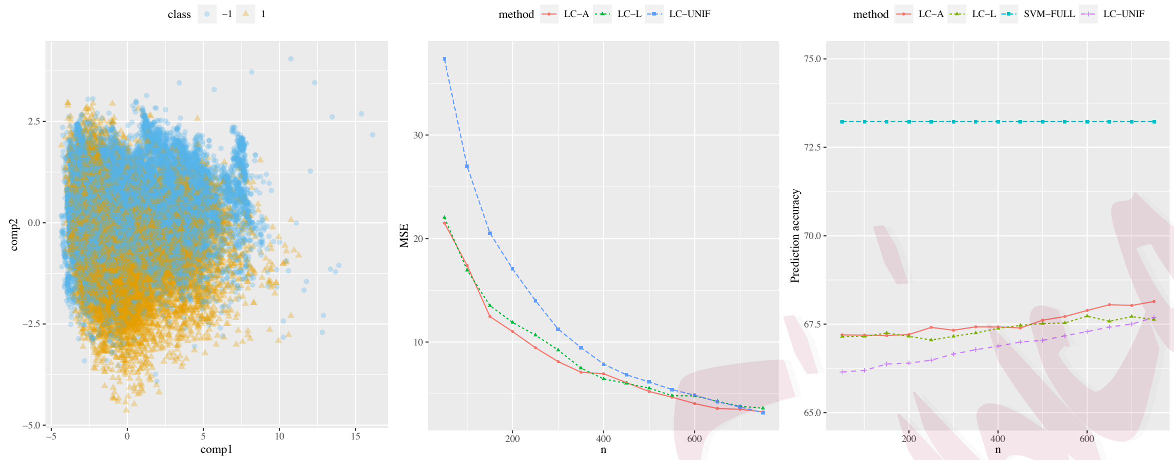

Figure 7: Analysis results for CASP dataset. Left panel: visualization with principal component analysis. Middle panel: MSE in approximating the full sample SVM estimator $\widehat{\boldsymbol{\beta}}$. Right panel: prediction accuracy (%).

Table 2: Comparison of CPU time (in seconds) for CASP dataset.

| Method | $n$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 50 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 |
| **LC-A** | 0.70 | 0.76 | 0.87 | 1.01 | 1.18 | 1.33 | 1.52 | 1.75 | 2.03 |
| **LC-L** | 0.69 | 0.75 | 0.85 | 1.00 | 1.16 | 1.33 | 1.51 | 1.76 | 2.03 |
| **LC-UNIF** | 0.39 | 0.42 | 0.53 | 0.66 | 0.81 | 0.96 | 1.11 | 1.20 | 1.52 |
| **SVM-FULL** | | | | 11.93 | | | | | |

agrees with numerical studies, and it is a great improvement of the method to approximate the full sample SVM. The middle and right panels of Figure 7 present the mean squared errors of approximating the full sample SVM estimator and the prediction performances. The good performance of the optimal leverage classifiers is consistent with our theory and the numerical studies.

## 6. Conclusion

Constructing accurate classifiers with informative subsamples from large-scale datasets is a crucial task in statistical analysis and machine learning. In this paper, we propose a novel leverage classifier for SVM under the subsampling framework to address the computational challenge. We construct optimal leverage classifiers by minimizing the unconditional asymptotic variance with double randomnesses. Our extensive numerical investigations demonstrate that the proposed methods provide satisfactory performances in estimation, computation, and prediction.

Subsampling is a fast and effective strategy for processing large-scale datasets and further research is needed for more delicate statistical models. We conclude this paper with several future topics. First, our binary subsampling leverage classifier may be extended to multi-classification problems by one-versus-one or one-versus-rest SVM in a linear nonseparable setting. Second, one limitation in our work is that we only focus on the linear SVM for nonseparable cases to shed light on the leverage classifiers. Extensions of the leverage classifiers to more general settings, such as kernel SVM in reproducing kernel Hilbert spaces, remain challenging because it is unclear how to integrate existing asymptotic results (Hable, 2012) with our subsampling framework. Third, it is worth further exploring the trade-off between estimation efficiency and computation complexity under measurement constraints. Finally, investigating other optimal criteria, such as minimizing the classification error or maximizing the prediction accuracy, also merits further research.

## Acknowledgement

The authors thank the Editor, Associate Editor, and anonymous referees for their helpful comments that have resulted in significant improvements in this article. The first three authors contributed equally to this work. Correspondence should be addressed to Ping Ma (pingma@uga.edu). Han and Zou's research were supported by the China National Key R&D Program grants No. 2019YFC1908502, 2022YFA1003703, 2022YFA1003802, 2022YFA1003803 and NNSF of China grants No. 11925106, 12231011, 11931001 and 11971247. Zhang's work was partially sponsored by the National Natural Science Foundation of China No.72101058, 11690014, and the Science and Technology Commission of Shanghai Municipality No.17JC1420200. Yu's work was supported by NSFC grant 12001042, Beijing Municipal Natural Science Foundation No. 1232019, and the Beijing Institute of Technology research fund program for young scholars. Meng's work was supported by NSFC grant 12101606. The research was partially supported by the University of Georgia Provost's Research grant, the U.S. National Science Foundation under grants DMS-1903226, DMS-1925066, DMS-2124493, DMS-2311297, DMS-2319279, U.S. National Institute of Health under grant R01GM152814.

## Supplementary Material

A supplementary PDF file contains the proof of theoretical results and additional simulation results in our paper.

## References

Ai, M., Yu, J., Zhang, H., and Wang, H. (2018). Optimal subsampling algorithms for big data regressions. *Statistica Sinica*, **6**(4):363–392.

Atkinson, A., Donev, A., and Tobias, R. (2007). *Optimum experimental designs, with SAS*. Oxford University Press.

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, **6**(9):1579–1619.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152.

Camelo, S. A., González-Lima, M. D., and Quiroz, A. (2015). Nearest neighbors methods for support vector machines. *Annals of Operations Research*, **235**(1):85–101.

Chang, E. Y. (2011). PSVM: Parallelizing support vector machines on distributed computers. In *Foundations of Large-Scale Multimedia Information Management and Retrieval*, pages 213–230. Springer.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**(3):273–297.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P. (2012). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, **13**(1):3475–3506.

Drineas, P., Mahoney, M. W., Muthukrishnan, S., and Sarlós, T. (2011). Faster least squares approximation. *Numerische Mathematik*, **117**(2):219–249.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Durrett, R. (2019). *Probability: theory and examples*, volume 49. Cambridge university press.

Fan, J., Li, R., Zhang, C.-H., and Zou, H. (2020). *Statistical foundations of data science*. Chapman and Hall/CRC.

Hable, R. (2012). Asymptotic normality of support vector machine variants and other regularized kernel methods. *Journal of Multivariate Analysis*, **106**:92–117.

Han, Y., Ma, P., Ren, H., and Wang, Z. (2023). Model checking in large-scale dataset via structure-adaptive-sampling. *Statistica Sinica*, **33**:303–329.

Hastie, T., Tibshirani, R., and Friedman, J. (2010). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S. (2008).

A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415.

Iraji, M. S. and Ameri, H. (2016). RMSD protein tertiary structure prediction with soft computing. *IJ Mathematical Sciences and Computing*, **2**:24–33.

Kaufman, L. (1998). Solving the quadratic programming problem arising in support vector classification. *Advances in Kernel Methods-Support Vector Learning*, pages 147–167.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, **33**(1):82–95.

Koo, J.-Y., Lee, Y., Kim, Y., and Park, C. (2008). A Bahadur representation of the linear support vector machine. *Journal of Machine Learning Research*, **9**(7):1343–1368.

Lesk, A. (2019). *Introduction to Bioinformatics*. Oxford University Press.

Li, T. and Meng, C. (2020). Modern subsampling methods for large-scale least squares regression. *International Journal of Cyber-Physical Systems (IJCPS)*, 2(2):1–28.

Lin, Y. (2004). A note on margin-based loss functions in classification. *Statistics & Probability letters*, **68**(1):73–82.

Lin, Y., Wahba, G., Zhang, H., and Lee, Y. (2002). Statistical properties and adaptive tuning of support vector machines. *Machine Learning*, **48**(1):115–136.

Ma, P., Chen, Y., Zhang, X., Xing, X., Ma, J., and Mahoney, M. W. (2022). Asymptotic analysis of sampling estimators for randomized numerical linear algebra algorithms. *Journal of Machine Learning Research*, **23**(1):7970–8014.

Ma, P., Huang, J. Z., and Zhang, N. (2015a). Efficient computation of smoothing splines via adaptive basis sampling. *Biometrika*, **102**(3):631–645.

Ma, P., Mahoney, M. W., and Yu, B. (2015b). A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, **16**(1):861–911.

Mahoney, M. W. and Drineas, P. (2009). CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, **106**(3):697–702.

Mehrotra, S. (1992). On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, **2**(4):575–601.

Meng, C., Xie, R., Mandal, A., Zhang, X., Zhong, W., and Ma, P. (2021). Lowcon: A design-based subsampling approach in a misspecified linear model. *Journal of Computational and Graphical Statistics*, **30**(3):694–708.

Meng, C., Zhang, X., Zhang, J., Zhong, W., and Ma, P. (2020). More efficient approximation of smoothing splines via space-filling basis selection. *Biometrika*, **107**(3):723–735.

Ohlsson, E. (1989). Asymptotic normality for two-stage sampling from a finite population. *Probability Theory and Related Fields*, **81**(3):341–352.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Ren, H., Zou, C., Chen, N., and Li, R. (2022). Large-scale datastreams surveillance via pattern-oriented-sampling. *Journal of the American Statistical Association*, **117**(538):794–808.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer.

Shalev-Shwartz, S., Singer, Y., Srebro, N., and Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for SVM. *Mathematical Programming*, **127**(1):3–30.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Champman & Hall.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Tsang, I. W., Kwok, J. T., and Cheung, P.-M. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, **6**(4):363–392.

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

Wahba, G., Lin, Y., Lee, Y., and Zhang, H. (2003). Optimal properties and adaptive tuning of standard and nonstandard support vector machines. In *Nonlinear Estimation and Classification*, pages 129–147. Springer.

Wang, H. and Ma, Y. (2021). Optimal subsampling for quantile regression in big data. *Biometrika*, **108**(1):99–112.

Wang, H., Zhu, R., and Ma, P. (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, **113**(522):829–844.

Wang, Z., Crammer, K., and Vucetic, S. (2012). Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training. *Journal of Machine Learning Research*, **13**(1):3103–3131.

Williams, C. and Seeger, M. (2000). Using the nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*, **13**.

Yu, J., Wang, H., Ai, M., and Zhang, H. (2022). Optimal distributed subsampling for maximum quasi-likelihood estimators with massive data. *Journal of the American Statistical Association*, **117**(537):265–276.