High-Dimensional Analysis of Bootstrap Ensemble Classifiers

Malik Tiomoko
Huawei Noah's Ark Lab,
France

Hamza Cherkaoui
SAMOVAR, Télécom SudParis
Institut Polytechnique de Paris
France

Mohamed E. A. Seddik
Technology Innovation Institute,
United Arab Emirates

Cosme Louart
Chinese University of Hong Kong,
China

Ekkehard Schnoor
Fraunhofer HHI, Germany,
Aalto University
Finland

Balázs Kégl
Huawei Noah's Ark Lab,
France

Abstract

Bootstrap methods have long been the cornerstone of ensemble learning in machine learning. This paper presents a theoretical analysis of bootstrap techniques applied to the Least Square Support Vector Machine (LSSVM) ensemble in the context of large and growing sample sizes and feature dimensionalities. Using tools from Random Matrix Theory, we investigate the performance of this classifier that aggregates decision functions from multiple weak classifiers, each trained on different subsets of the data. We provide insights into the use of bootstrap methods in high-dimensional settings, enhancing our understanding of their impact. Based on these findings, we propose strategies to select the number of subsets and the regularization parameter that maximize the performance of the LSSVM. Empirical experiments on synthetic and real-world datasets validate our theoretical results.

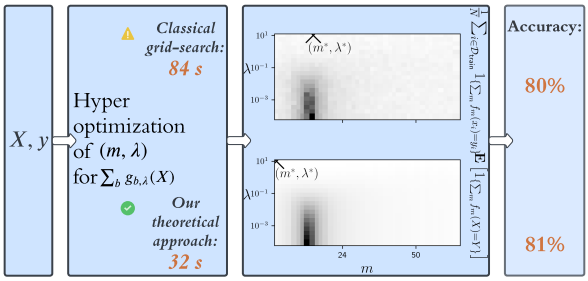

Figure 1: From costly grid search to theory-guided tuning. Heatmaps illustrate how hyperparameter choices (regularization and ensemble size) affect the LSSVM decision boundary. Our RMT-based analysis provides closed-form approximations that guide near-optimal settings without exhaustive search.

1 INTRODUCTION

Bootstrap methods (Efron, 1979) are a cornerstone of modern statistics and machine learning. By resampling data and aggregating models, ensemble techniques such as bagging (Breiman, 1996) reduce variance, improve robustness, and enhance generalization (Efron and Tibshirani, 1994). Despite their empirical success, however, their behavior in high-dimensional regimes remains only partially understood.

Classical bootstrap analyses assume that the sample size n grows while the dimension d is fixed (Freedman, 1981). These results break down when n and d grow jointly—a regime that is now ubiquitous in modern applications. Recent works have shown that bootstrap procedures can become inconsistent or severely biased in high dimensions (El Karoui and Purdom, 2018; Clarté et al., 2024; Bellec and Koriyama, 2024), raising fundamental questions: *When does bootstrapping remain reliable? How should one choose the resampling strategy and regularization in high dimensions?*

We address these questions in the context of *high-dimensional classification ensembles*. In contrast to prior works that primarily focus on regression or rely on Gaussian assumptions, we study bootstrap Least

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

Squares Support Vector Machines (LSSVMs) under a general class of *concentrated random vectors*, which captures structured data distributions commonly encountered in modern learning pipelines. Leveraging tools from *Random Matrix Theory (RMT)*, we derive an explicit and tractable characterization of the ensemble decision score. In particular, we show that it converges to a Gaussian distribution, for which we obtain closed-form expressions of the mean and variance, leading to an explicit approximation of the classification error. These results enable principled tuning of both the regularization parameter and the ensemble size (see Figure 1).

Contributions.

1. **Explicit asymptotic characterization.** We derive deterministic equivalents for the mean and variance of the bootstrap ensemble decision score in the proportional regime ($n, d \to \infty$), and show that it converges to a Gaussian distribution.
2. **Analysis of correlated bootstrap ensembles.** Our analysis accounts for dependencies induced by subsampling (without replacement), leading to new expressions that capture the effect of ensemble size and sampling strategy on performance.
3. **Closed-form classification error.** We obtain an explicit approximation of the misclassification rate as a function of data statistics, regularization, and ensemble size.
4. **Theory-driven model selection.** We propose a principled method to select hyperparameters without cross-validation, based on the derived asymptotic risk, leading to more stable and interpretable tuning procedures.

Relation to prior work. Most high-dimensional analyses of bootstrap methods focus on regression or parameter estimation (Du et al., 2023; Koriyama, 2024; Bellec and Koriyama, 2024), or rely on techniques from statistical mechanics and replica methods, which often yield implicit characterizations. In contrast, we consider classification ensembles under a broader class of data models and derive *explicit* and interpretable expressions for both the decision score and the classification error. This distinction is crucial for enabling practical model selection.

Takeaway. Overall, our results show that bootstrap ensembles in high dimensions can be understood as predictable systems with tractable behavior, rather than purely heuristic tools. This perspective provides both theoretical insight and practical benefits for designing and tuning ensemble classifiers.

The remainder of the paper is structured as follows: section 2 reviews related work, section 3 introduces the bootstrap LSSVM framework and statistical model, section 4 presents our theoretical results, section 5 discusses their implications, section 6 reports empirical validation, and section 7 concludes.

Notation. Matrices are denoted by capital letters, e.g., $\mathbf{X} \in \mathbb{R}^{d\times n}$ and $\mathbf{I}_d$. Vectors are in bold lowercase (e.g., $\mathbf{v} \in \mathbb{R}^d$), with $\mathbf{0}_d$ and $\mathbb{1}_d$ for all-zeros and all-ones. For $\mathbf{v} \in \mathbb{R}^d$, $\mathcal{D}_{\mathbf{v}}$ is the diagonal matrix with $\mathbf{v}$ on its diagonal, and $\mathrm{Tr}(\mathbf{A})$ denotes the trace of square matrix $\mathbf{A}$. Scalars appear in plain letters (a, λ). Indices i, j denote data samples; $\ell \in \{1, 2\}$ denotes class labels.

We use standard probability notation: $\mathbb{E}[X]$ for expectation, $\mathrm{Var}(X)$ for variance, and $\mathbf{\Sigma}_{\mathbf{v}} = \mathrm{Cov}(\mathbf{v})$ for covariance, where $\bar{\mathbf{v}}$ denotes the mean (or its deterministic equivalent). For $p \geq 1$, $\|\cdot\|_p$ denotes the ℓ_p-norm, and $\|\cdot\|_F$ the Frobenius norm. Concatenation is denoted by $[\mathbf{v}_1, \mathbf{v}_2]^\top \in \mathbb{R}^{2d}$ (horizontal) and $[\mathbf{v}_1|\mathbf{v}_2] \in \mathbb{R}^{d\times 2}$ (vertical).

2 RELATED WORK

Understanding bootstrap methods in high-dimensional regimes requires connecting several lines of work spanning statistics, machine learning, random matrix theory, and statistical mechanics. We organize the literature into four main directions: (i) classical bootstrap and ensemble methods, (ii) high-dimensional failures of resampling, (iii) constructive asymptotic analyses of bagging and subsampling, and (iv) model averaging and risk estimation.

Classical bootstrap and ensemble methods. The bootstrap, introduced by Efron (1979), was theoretically justified in low-dimensional settings (Freedman, 1981; Singh, 1981). Ensemble learning methods, including early work in neural networks (Hansen and Salamon, 1990; Perrone and Cooper, 1993; Krogh and Vedelsby, 1995), culminated in bagging (Breiman, 1996), which improves prediction by averaging models trained on resampled datasets. These results rely on fixed-dimensional asymptotics and do not extend to modern high-dimensional regimes.

Failures of bootstrap in high dimension. When both the sample size n and dimension d grow proportionally, classical bootstrap procedures can break down. El Karoui and Purdom (2018) show that residual and pairs bootstrap can be inconsistent for variance estimation in regression, while El Karoui and Purdom (2019) demonstrate failures in spectral statistics. These works highlight that naive resampling is not reliable in high-dimensional settings.

Constructive high-dimensional analyses. Recent work has focused on deriving precise asymptotic characterizations of resampling methods. Using tools from random matrix theory, Clarté et al. (2024) analyze bootstrap and subsampling for regularized regression and generalized linear models, establishing conditions under which they remain valid. Bellec and Koriyama (2024) derive fixed-point equations describing correlations in subsampled estimators, while Koriyama (2024) provide exact asymptotics for bagged M-estimators. Related works also study subsampled ridge ensembles and random-feature models (Du et al., 2023; Loureiro et al., 2022).

In parallel, a complementary line of work based on statistical mechanics and replica methods has provided sharp asymptotic predictions for ensemble methods. Early contributions (Sollich and Krogh, 1995; Krogh and Sollich, 1997) established fundamental connections between ensemble learning, overfitting, and regularization effects. More recent works extend these techniques to modern settings, analyzing subsampling, under-bagging, and ensemble methods for classification and variable selection (Takahashi, 2024, 2025). These approaches typically yield implicit characterizations of performance through fixed-point equations.

Model averaging and risk estimation. Ensembling can be interpreted as a form of implicit regularization. LeJeune et al. (2019) show that averaging least-squares estimators over subsets induces a ridge-like effect, while Ando and Komaki (2023) analyze optimal weighting in high-dimensional model averaging. For hyperparameter tuning, cross-validation is widely used, but its validity is not guaranteed: Bellec et al. (2023) show inconsistency of generalized cross-validation in some ensemble settings, whereas Du et al. (2023) establish consistency results under proportional asymptotics.

Contrast with our work. Most prior analyses focus on regression, spectral properties, or inference, and often rely on Gaussian assumptions or yield implicit characterizations through fixed-point equations. In contrast, we study *classification ensembles* based on LSSVMs under a general class of concentrated random vectors. Our approach provides *explicit and interpretable* expressions for the mean and variance of the decision score, leading to a closed-form approximation of the classification error. This explicit characterization enables practical and stable model selection, distinguishing our contribution from prior work.

A structured comparison with representative works is provided in Table 1 (Appendix 8).

3 PRELIMINARIES

3.1 Least Squares SVM (LSSVM)

We consider a binary classification problem with data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the feature vectors, and $y_i \in \{-1, +1\}$ are the corresponding class labels. The Least Squares Support Vector Machine (LSSVM) estimator $\boldsymbol{\omega}^\star$ solves the following regularized least-squares optimization problem:

$$\min_{\boldsymbol{\omega}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \left(y_i - \mathbf{x}_i^\top \boldsymbol{\omega}\right)^2 + \frac{\lambda}{2}\|\boldsymbol{\omega}\|_2^2,$$

where $\lambda > 0$ is the regularization parameter that controls the trade-off between fitting the data and avoiding overfitting. The closed-form solution to this optimization problem is given by:

$$\boldsymbol{\omega}^{\star\top} = \frac{1}{n}\mathbf{y}^\top \mathbf{X}^\top \left(\frac{1}{n}\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I}_d\right)^{-1}, \quad \boldsymbol{\omega}^\star \in \mathbb{R}^d, \quad (1)$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ is the data matrix, and $\mathbf{y} = (y_1, \ldots, y_n)^\top$ is the vector of class labels. The classification decision for a test point $\mathbf{x}$ is made based on the sign of the decision score $g(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\omega}^\star$, i.e., $\mathrm{sign}(g(\mathbf{x}))$.

Role of Regularization. The regularization parameter λ plays a crucial role in controlling the complexity of the model. A smaller value of λ leads to a model that is more flexible and can overfit the data, while a larger value increases the penalty for large values of $\boldsymbol{\omega}$, thus leading to a simpler model that may underfit. In high-dimensional settings where the number of features d is comparable to or larger than the number of samples n, proper tuning of λ is essential to avoid both overfitting and underfitting.

3.2 Bootstrap LSSVM Ensembles

To stabilize performance and reduce variance, we construct an ensemble of m bootstrapped classifiers. From the training set, m resampled subsets $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^m$ are drawn, each of size n_i. The i-th classifier is trained using the LSSVM formulation and yields a weight vector:

$$\boldsymbol{\omega}_i^\top = \frac{1}{n_i}\mathbf{y}_i^\top \mathbf{X}_i^\top \left(\frac{1}{n_i}\mathbf{X}_i\mathbf{X}_i^\top + \lambda_i \mathbf{I}_d\right)^{-1},$$

where λ_i is the regularization parameter for the i-th bootstrap subset. The corresponding decision score is $g_i(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\omega}_i$.

The ensemble average is computed as:

$$g(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m g_i(\mathbf{x}),$$

and the test point $\mathbf{x}$ is classified as $\mathcal{C}_2$ if $g(\mathbf{x}) > 0$ and as $\mathcal{C}_1$ otherwise. This averaging process helps reduce variance by mitigating the impact of fluctuations across bootstrap replicates. In our theoretical analysis, the decision score $g(\mathbf{x})$ plays a central role, as its mean and variance fully characterize the asymptotic classification error.

3.3 High-Dimensional Regime and Data Assumptions

We work in a proportional-growth setting, where both the sample size n and feature dimension d grow simultaneously, but their ratio remains constant. This allows the application of tools from Random Matrix Theory (RMT) to derive asymptotic results for the bootstrap ensemble classifiers. This setting is particularly relevant in modern machine learning applications where both n and d are large.

Assumption 3.1 (Large n, large d) As both n and d tend to infinity, their ratio $\frac{d}{n}$ converges to a constant $c_0 \in (0, \infty)$. Each subset size n_i scales as $n_i/n \to c_i > 0$, while the number of bootstrap replicates $m = \mathcal{O}(1)$. Furthermore, the class proportions converge to constants $\mathbf{c} = (c_1, c_2)$.

Assumption 3.2 (Concentrated random vectors) Each column of $\mathbf{X}$ is an independent random vector with mean $\boldsymbol{\mu}_\ell$ and covariance $\boldsymbol{\Sigma}_\ell$ conditional on its class $\mathcal{C}_\ell$. Furthermore, we assume that $\mathbf{x}$ satisfies an exponential concentration inequality. Specifically, for any 1-Lipschitz function $\varphi : \mathbb{R}^d \to \mathbb{R}$,

$$\mathbb{P}\left(|\varphi(\mathbf{x}) - \mathbb{E}[\varphi(\mathbf{x})]| \geq t\right) \leq Ce^{-(t/\sigma)^q}, \qquad \forall t > 0,$$

for some constants $C > 0$, $q > 0$, and dimension-independent σ. We denote this property as $\mathbf{x} \propto \mathcal{E}_q(\sigma \mid \|\cdot\|_2)$.

Interpretation. Exponential concentration implies that high-dimensional data behaves almost deterministically with respect to any Lipschitz-continuous function. In other words, fluctuations around the mean are exponentially unlikely. This assumption is satisfied by many natural distributions, such as (a) isotropic Gaussian vectors ($q = 2$), (b) uniformly distributed vectors on the sphere, and (c) any Lipschitz transformations of these (e.g., features produced by deep generative models (Seddik et al., 2020)).

This property allows for the concentration of sample averages and quadratic forms, which is key for deriving deterministic equivalents for the decision score $g(\mathbf{x})$.

For later use, we define the class means $\boldsymbol{\mu}_\ell = \mathbb{E}[\mathbf{x} \mid \mathbf{x} \in \mathcal{C}_\ell]$, covariances $\boldsymbol{\Sigma}_\ell = \mathrm{Cov}(\mathbf{x} \mid \mathbf{x} \in \mathcal{C}_\ell)$, the mean matrix $\mathbf{M} = [\boldsymbol{\mu}_1 | \boldsymbol{\mu}_2] \in \mathbb{R}^{d \times 2}$, and generalized covariances $\mathbf{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$.

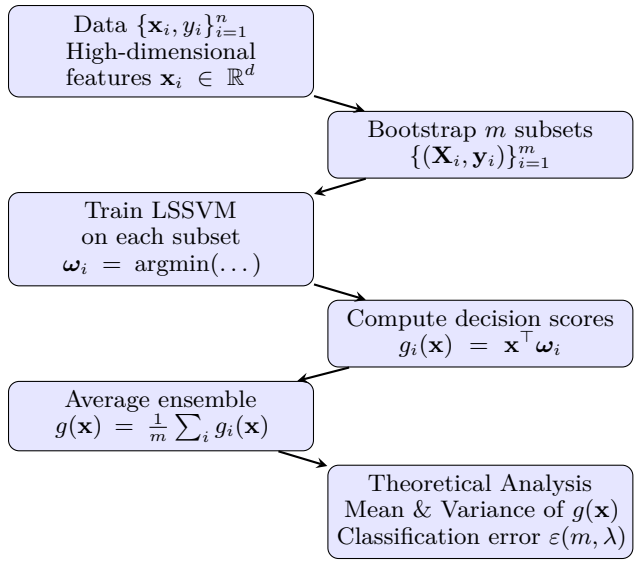

Figure 2: Flowchart of the bootstrap LSSVM ensemble framework. Data is bootstrapped, LSSVMs are trained on each subset, decision scores are computed and averaged, and theoretical analysis predicts classification error.

4 ASYMPTOTIC DISTRIBUTION OF THE DECISION SCORE

In this section, we characterize the behavior of the ensemble decision score $g(\mathbf{x})$ in the high-dimensional regime $n, d \to \infty$ with $d/n \to c_0 \in (0, \infty)$. We prove that $g(\mathbf{x})$ converges in distribution to a Gaussian random variable whose mean and variance can be expressed in closed form. This result provides the foundation for computing the asymptotic classification error in terms of the regularization parameter λ and ensemble size m.

4.1 Deterministic Equivalents: Key Quantities

The central tool is a deterministic equivalent of the resolvent matrix, which captures the effect of data covariance and class proportions. We introduce the following fixed-point system:

$$\bar{\mathbf{Q}} = \left(\sum_{\ell=1}^{2} \frac{n_\ell}{n} \frac{\mathbf{C}_\ell}{1 + \delta_\ell} + \lambda \mathbf{I}_d \right)^{-1}, \tag{2}$$

$$\delta_\ell = \frac{1}{n} \mathrm{Tr}(\mathbf{C}_\ell \bar{\mathbf{Q}}), \qquad \ell \in \{1, 2\}, \tag{3}$$

where $\mathbf{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$ are the generalized covariance matrices.

Intuitively, $\bar{\mathbf{Q}}$ plays the role of an "effective inverse covariance" adjusted by both regularization (λ) and bootstrap resampling. The correction terms δ_ℓ encode

the high-dimensional shrinkage that vanishes in the classical $d \ll n$ regime but remains essential when $d \sim n$.

To describe fluctuations of $g(\mathbf{x})$, we also define $\mathbf{K}_\ell$, a covariance proxy matrix:

$$\mathbf{K}_\ell = \bar{\mathbf{Q}}\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}} + \sum_{\ell'=1}^{2} \frac{n_{\ell'}}{n} \frac{d_{\ell'}^{(\ell)}}{(1+\delta_{\ell'})^2} \bar{\mathbf{Q}}\mathbf{C}_{\ell'}\bar{\mathbf{Q}},$$

where the coefficients $d_{\ell'}^{(\ell)}$ are obtained by solving

$$\mathbf{d}^{(\ell)} = \left(\mathbf{I}_2 - \tilde{\mathbf{V}}\tilde{\mathbf{A}}\right)^{-1}\bar{\mathbf{t}}^{(\ell)}.$$

Here,

$$\begin{aligned}
\bar{\mathbf{t}}^{(\ell)} &= \tfrac{1}{n}\left[\mathrm{Tr}(\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}),\ \mathrm{Tr}(\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}})\right]^\top, \\
\tilde{\mathbf{V}} &= \tfrac{1}{n}\begin{pmatrix} \mathrm{Tr}(\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}) & \mathrm{Tr}(\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}) \\ \mathrm{Tr}(\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}) & \mathrm{Tr}(\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}) \end{pmatrix}, \\
\tilde{\mathbf{A}} &= \tfrac{1}{n}\,\mathrm{diag}\left(\tfrac{n_1}{(1+\delta_1)^2},\ \tfrac{n_2}{(1+\delta_2)^2}\right).
\end{aligned}$$

Finally, we define the correction vector

$$\boldsymbol{\delta}' = \left(\delta_1', \delta_2'\right) = \left(\tfrac{1}{n}\,\mathrm{Tr}(\boldsymbol{\Sigma}_1\mathbf{K}_\ell),\ \tfrac{1}{n}\,\mathrm{Tr}(\boldsymbol{\Sigma}_2\mathbf{K}_\ell)\right),$$

which will appear in the variance expression.

4.2 Main Result: Gaussian Limit

Theorem 4.1 (Asymptotic Distribution of the Decision Score) Under Assumptions 3.1 – 3.2, the ensemble decision score of a test point $\mathbf{x} \in \mathcal{C}_\ell$ converges in distribution to a Gaussian:

$$g(\mathbf{x}) \xrightarrow[n,d\to\infty]{} \mathcal{N}(\mathfrak{m}_\ell, \sigma_\ell^2),$$

where the class-specific mean $\mathfrak{m}_\ell$ and variance σ_ℓ^2 are given by deterministic equivalents.

Mean.

$$\mathfrak{m}_\ell = \tilde{\mathbf{y}}^\top\mathbf{D}_\mathbf{c}\mathbf{D}_{\boldsymbol{\delta}}\,\mathbf{M}^\top\bar{\mathbf{Q}}\,\boldsymbol{\mu}_\ell,$$

where $\tilde{\mathbf{y}} = [-1, 1]^\top$, $\mathbf{M} = [\boldsymbol{\mu}_1 \,|\, \boldsymbol{\mu}_2]$, and $\mathbf{D}_\mathbf{c}, \mathbf{D}_{\boldsymbol{\delta}}$ are diagonal matrices of class proportions and fixed-point terms.

Variance.

$$\begin{aligned}
\sigma_\ell^2 = &\tfrac{1}{m}\tilde{\mathbf{y}}^\top\mathbf{V}_\ell\tilde{\mathbf{y}} + \tfrac{1}{m}\tilde{\mathbf{y}}^\top\mathbf{D}_\mathbf{c}\mathbf{D}_{\boldsymbol{\delta}}\mathbf{M}^\top\bar{\mathbf{V}}_\ell\mathbf{M}\mathbf{D}_\mathbf{c}\mathbf{D}_{\boldsymbol{\delta}}\tilde{\mathbf{y}} \\
&- \tfrac{2}{m}\tilde{\mathbf{y}}^\top\mathbf{D}_\mathbf{c}\mathbf{D}_{\boldsymbol{\delta}'}\mathbf{M}^\top\bar{\mathbf{Q}}\mathbf{M}\mathbf{D}_{\boldsymbol{\delta}}\mathbf{D}_\mathbf{c}\tilde{\mathbf{y}} \\
&+ \tfrac{m-1}{m}\tilde{\mathbf{y}}^\top\mathbf{D}_\mathbf{c}\mathbf{D}_{\boldsymbol{\delta}}\mathbf{M}^\top\bar{\mathbf{Q}}\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}\mathbf{M}\mathbf{D}_{\boldsymbol{\delta}}\mathbf{D}_\mathbf{c}\tilde{\mathbf{y}}.
\end{aligned}$$

The structure of σ_ℓ^2 highlights the trade-off between variance reduction due to averaging ($1/m$ scaling) and additional dependencies across bootstraps.

4.3 Theoretical Classification Error

Having obtained the asymptotic distribution of $g(\mathbf{x})$, the misclassification probability can be written in closed form.

Remark 4.1 (Classification Error) Suppose that for $\mathbf{x} \in \mathcal{C}_\ell$, the score follows $g(\mathbf{x}) \sim \mathcal{N}(\mathfrak{m}_\ell, \sigma_\ell^2)$. Then the overall error rate is

$$\varepsilon(\eta) = c_1\,\Phi\left(\frac{\eta - \mathfrak{m}_1}{\sigma_1}\right) + c_2\left[1 - \Phi\left(\frac{\eta - \mathfrak{m}_2}{\sigma_2}\right)\right],$$

where c_ℓ are class proportions, Φ is the standard Gaussian CDF, and η is the decision threshold. The optimal threshold $\eta^\star$ lies between $\mathfrak{m}_1$ and $\mathfrak{m}_2$ and minimizes $\varepsilon(\eta)$.

Interpretation. The above result formalizes how bootstrap ensembles behave in high dimensions:

1. The *means* $\mathfrak{m}_\ell$ quantify separation between classes.
2. The *variances* σ_ℓ^2 reflect both intrinsic randomness and ensemble averaging.
3. The classification error ε depends explicitly on the regularization λ and ensemble size m, allowing principled tuning without exhaustive grid search.

Generalization to Other Classifiers. While the results are derived for the LSSVM model, they can be generalized to other linear classifiers, such as logistic regression, under similar assumptions. This extension provides a more comprehensive understanding of ensemble methods in high-dimensional classification.

Exponential Concentration. The assumption of exponential concentration of the data (Assumption 3.2) ensures that high-dimensional data behaves deterministically with respect to any Lipschitz-continuous function, which is critical for deriving deterministic equivalents for the decision score. This assumption is satisfied by many common data distributions, including isotropic Gaussian vectors and uniform distributions on the sphere.

5 DISCUSSION

Our main result, Theorem 4.1, establishes that the ensemble decision score $g(\mathbf{x})$ is asymptotically Gaussian, with explicit formulas for its mean and variance. These two quantities determine the classification error and allow us to understand, in a principled way, how the parameters of the bootstrapped LSSVM interact.

5.1 Simplified Gaussian Example

To build intuition, let us specialize to the simplest setting: two Gaussian classes with opposite means $\pm\boldsymbol{\mu}$

and identity covariance. In this case, the mean of the decision score reduces to

$$\mathfrak{m}_\ell = (-1)^\ell \frac{\|\boldsymbol{\mu}\|^2}{1+\lambda+\lambda\delta+\|\boldsymbol{\mu}\|^2},$$
$$\delta = \frac{\sqrt{(1+\lambda-md/n)^2+4\lambda md/n}-(1+\lambda-md/n)}{2\lambda}.$$

This expression highlights the key forces at play:

1. The numerator $\|\boldsymbol{\mu}\|^2$ represents the intrinsic *class separation* (signal).
2. The denominator contains penalty terms due to *regularization* (λ), *bootstrap sampling* (m), and *dimensionality* (d).

Hence, stronger class separation improves performance, while higher regularization, more bootstraps, and higher dimensionality all reduce the effective mean, tightening the decision boundary. These penalties are not purely detrimental: they are the mechanism that prevents overfitting and ensures stability.

5.2 Variance Contributions

The variance of the decision score decomposes into four interpretable terms (details in Appendix 11.2). While the exact formulas are technical, the main insights can be summarized as follows:

- Regularization–dimension term (term_1): scales as $1/m$, decreases with more classifiers, and stabilizes fluctuations.
- Signal-driven terms (term_2): polynomial in $\|\boldsymbol{\mu}\|^2$, representing how stronger signal also introduces variability. These are suppressed by λ and d.
- Interaction term (term_3): mixes regularization and signal, but decays with larger m and higher δ.
- Cross-classifier interaction (term_4): accounts for dependencies across bootstrapped classifiers. For large m it tends to a constant, but vanishes with higher dimensionality.

The key qualitative message is:

1. Increasing the ensemble size m reduces variance through $1/m$ averaging, but at the cost of weaker per-classifier signal (since each sees fewer samples).
2. Regularization λ and dimensionality d both act to stabilize fluctuations, but they also shrink the effective separation of the classes.
3. Stronger signal $\|\boldsymbol{\mu}\|^2$ boosts both the mean (good) and the variance (bad).

This antagonistic effect of mean versus variance is nothing but the **bias–variance tradeoff** in a high-dimensional, bootstrapped LSSVM setting.

5.3 Implications for Classification Error and Model Selection

From the Gaussian approximation of Theorem 4.1, the misclassification error depends directly on the ratio

$$\frac{\mathfrak{m}_1-\mathfrak{m}_2}{\sigma_1+\sigma_2},$$

which measures signal separation relative to uncertainty. Increasing m initially helps by reducing variance, but after a point the per-classifier weakness dominates, and performance may deteriorate. This explains the non-monotonic dependence of error on m observed in practice.

Similarly, tuning λ trades variance reduction for bias: too small λ leads to unstable classifiers, too large λ shrinks the margin excessively. The dimensionality d has an unavoidable effect: as d/n grows, both mean separation shrinks and variance grows, underscoring the well-known curse of dimensionality.

Practical takeaway. Our analysis provides a principled way to tune (m, λ) without resorting to exhaustive cross-validation. By computing theoretical means and variances of the decision score, one can estimate classification error in closed form and select parameters that balance the bias–variance tradeoff optimally. For example, one can choose m to maximize variance reduction without excessively weakening the per-classifier signal, and λ to prevent overfitting while maintaining a large margin between classes.

5.4 Summary

1. The mean of the decision score grows with signal strength $\|\boldsymbol{\mu}\|^2$ but decreases with regularization λ, number of bootstraps m, and dimension d.
2. The variance shrinks with m and λ, but grows with $\|\boldsymbol{\mu}\|^2$ and unfavorable d/n ratios.
3. The classification error reflects this competition, leading to an optimal regime where moderate m and carefully tuned λ outperform extremes.

Overall, the theory unifies classical insights on bias–variance tradeoff with modern high-dimensional random matrix effects, giving a clear roadmap for robust model selection in bootstrapped LSSVMs.

6 EXPERIMENTS

We validate our theoretical results on both synthetic and real datasets. Our objectives are twofold: first, to show that the asymptotic error predictions closely match the observed performance; second, to demonstrate that these predictions can effectively guide hyperparameter selection in practice. All experiments

were implemented in Python and run on 30 *Intel Xeon @ 3.20,GHz* CPUs, with a total runtime of approximately one hour. All code is open source and publicly available at https://github.com/hcherkaoui/optimal_bootstrap

6.1 Synthetic Illustration

We begin with controlled synthetic experiments, where all parameters are explicitly defined. This allows us to isolate the effects of the number of classifiers m and the regularization parameter λ.

Setup. We consider a binary classification task with 500 samples per class and feature dimension $d = 100$. The two classes, encoded as $\{-1, +1\}$, are drawn from Gaussian distributions with covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$. The class means are symmetric and sparse:

$$\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = 0.9\boldsymbol{e}_1,$$

where $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^\top \in \mathbb{R}^d$ is the first standard basis vector.

We build an ensemble of m classifiers $\{g_b\}_{b=1}^m$ (see section 3) and measure performance via the empirical classification error:

$$\tilde{\varepsilon}(m, \lambda) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\left\{ \sum_{b=1}^{m} g_b(x_i) \neq y_i \right\},$$

with $m \in \{1, \ldots, 50\}$ and $\lambda \in [10^{-2}, 10^{-1}]$. We then compare $\tilde{\varepsilon}(m, \lambda)$ with the theoretical error $\varepsilon(m, \lambda)$ derived in section 4. Each configuration was repeated 10 times and the results were averaged; uncertainty is indicated by transparency.

6.1.1 Hyperparameter Map Illustration

First, we set $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I}_d$, *i.e.*, we consider an isotropic covariance.

As shown in Figure 3, the empirical and theoretical error maps align almost perfectly. Both capture the same low-error and high-error regions, including a clear error peak around $m = 10$ when $\lambda = 0.0002$. The one-dimensional slices (right panels) confirm this agreement: theoretical curves lie exactly on top of empirical averages, demonstrating the accuracy of our predictions.

6.1.2 Effect of Covariance Structure

We next test robustness under different covariance matrices. Alongside the identity covariance, we use a Toeplitz covariance with exponentially decaying entries in its first row.

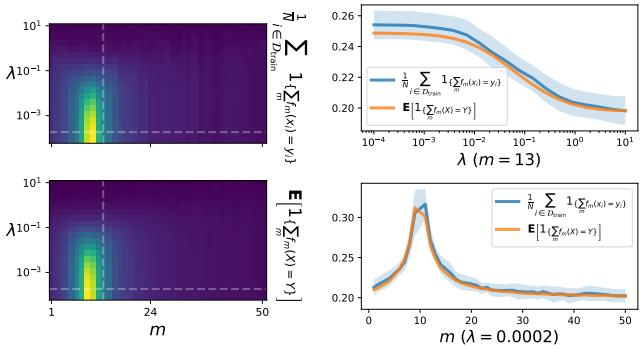

Figure 3: **(top-left)** Empirical classification error and **(bottom-left)** theoretical classification error as a function of m and λ. Darker blue indicates lower error. **(top-right)** Error as a function of λ (for fixed $m = 13$). **(bottom-right)** Error as a function of m (for fixed $\lambda = 0.0002$). Empirical errors are in blue, theory in orange. Shaded regions indicate standard deviation.

In Figure 4, the theory tracks correctly the empirical error across both covariance settings. Interestingly, qualitative behavior differs: under identity covariance, the error decreases steadily with m, whereas under Toeplitz covariance, the lowest error occurs already at $m = 1$, and adding classifiers worsens the performance. This highlights how the covariance structure directly influences the optimal ensemble size.

6.2 Real Data Benchmarks

We now turn to real-world datasets. Our approach selects hyperparameters by minimizing the theoretical error of Theorem 4.1:

$$(m^*, \lambda^*) = \arg\min_{m,\lambda} \varepsilon(m, \lambda).$$

We compare our theory-guided hyperoptimization ($m^\star$) against standard hyperparameter selection approaches: grid search (`Grid`), random search (`Rand`), a hyperoptimization method (`HPO` (Bergstra et al., 2011, 2022)) for different values of λ.

6.2.1 Practical Estimation of Moments

Our framework requires estimates of means and covariance matrices:

Means. We estimate class means via bootstrap averaging: $\hat{\boldsymbol{\mu}}_1 = \mathbb{E}^*[\mathbf{X}_{\mathcal{C}_1}]$, $\hat{\boldsymbol{\mu}}_2 = \mathbb{E}^*[\mathbf{X}_{\mathcal{C}_2}]$, which reduces bias and stabilizes estimates in finite samples.

Covariances. For the estimation of covariances, we use the Ledoit–Wolf shrinkage estimator (Ledoit and Wolf, 2004): $\hat{\boldsymbol{\Sigma}}_1 = \mathrm{LW}(\mathbf{X}_{\mathcal{C}_1})$, $\hat{\boldsymbol{\Sigma}}_2 = \mathrm{LW}(\mathbf{X}_{\mathcal{C}_2})$, which is particularly robust in high dimensions.

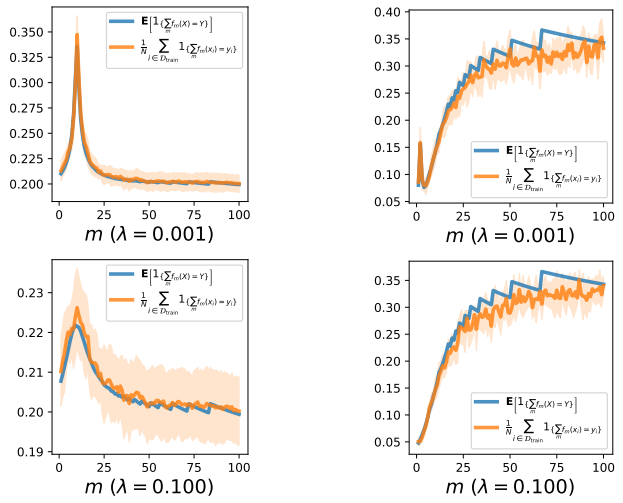

Figure 4: Classification error vs. m for $\lambda = 0.01$ (**top**) and $\lambda = 0.1$ (**bottom**), under identity (**left**) and Toeplitz (**right**) covariance. Empirical errors (orange) with standard deviation shading vs. theoretical errors (blue).

Remark on Computational Cost. An important advantage of our approach is its negligible additional cost. The theoretical classification error depends only on low-dimensional quantities. These quantities are already required for training the base classifiers and can be estimated efficiently (*e.g.,* via bootstrap means and Ledoit–Wolf covariance estimators). As a result, computing the theoretical error does not add asymptotic complexity beyond that of training the initial ensemble, making the method computationally efficient and scalable.

6.2.2 Model Selection on Real Datasets

We evaluate our method on three representative datasets, chosen to reflect distinct learning contexts where theory and practice may align differently:

1. *Pima Indians Diabetes* (Smith et al., 1988): $n = 768$, $d = 8$ clinical features, binary label. *Preprocessing:* shuffle and encode targets in $-1, 1$. This small-dimensional dataset serves as a stress test in a regime where concentration may be weak.
2. *Spambase* (Hopkins et al., 1999): $n = 4601$, $d = 57$ email features, binary spam label. *Preprocessing:* shuffle and encode targets in $\{-1, 1\}$. This dataset provides an intermediate-dimensional benchmark on real tabular data.
3. *Amazon Reviews (Books)* (Blitzer et al., 2007): $n = 2000$, $d = 100$ sentiment embeddings features obtained by applying PCA to the neural network weights, binary polarity task. *Preprocessing:* shuffle, select a balanced subset, PCA, and encode targets in $\{-1, 1\}$. This dataset is especially relevant since its deep embeddings align well with our theoretical assumptions.
4. *OpenL3 Audio (4Q)* (Cramer et al., 2019): $n = 3167$, $d = 100$ audio embeddings features obtained by applying PCA to the neural network weights, binary task obtained by collapsing the original 4 classes. *Preprocessing:* shuffle, PCA and encode targets in $\{-1, 1\}$. This dataset is also especially relevant since its deep embeddings align well with our theoretical assumptions.

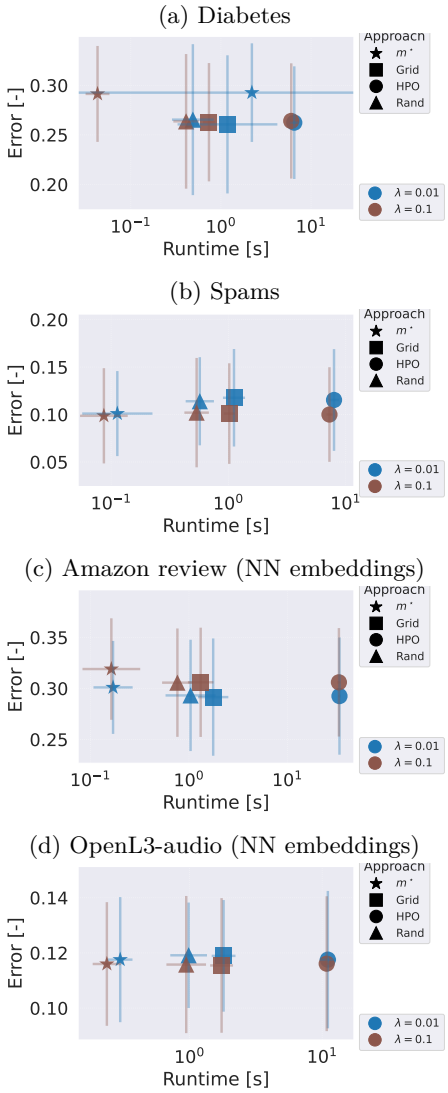

Figure 5: Performance comparison: test error versus wall-clock runtime across methods and regularization levels λ.

Discussion. Figure 5 shows that our method is consistently competitive in terms of both accuracy and

runtime:

1. **Diabetes:** Our method is slightly weaker in this case, which is expected since the dataset has only $d = 8$ features. In such a low-dimensional regime, concentration effects are limited, which reduces the validity of our theoretical assumptions. Including this example is nevertheless important, as it clearly illustrates the limitations of our approach.
2. **Spams:** This dataset shows that $d = 57$ is already sufficient to place the problem in a regime where our approach is effective: the approximation remains accurate and successfully identifies the hyperparameters yielding the lowest error.
3. **Amazon Reviews:** The performance is comparable to that of the best competing methods while requiring less computation time. Since these results are obtained on high-dimensional embeddings, they highlight the practical efficiency gains of our approach in realistic settings.
4. **OpenL3:** Our approach again performs similarly to the more costly alternative. It achieves strong predictive performance at a substantially lower computational cost, further confirming the practical relevance of the theory in this setting.

Summary. Overall, our approach is *fast* and *robust* compared to widely used alternatives. It performs well particularly in the neural embedding regime, which is precisely the class of problems motivating our theoretical framework. On purely low-dimensional tabular data (Diabetes), performance is slightly worse but well-understood, showing that our method is not a universal solution but is most relevant for modern machine learning.

6.2.3 Limitations: Mixed Feature Types

Our framework assumes concentrated data vectors, which holds for continuous embeddings but may fail with categorical or mixed features. To test this, we use the *Wisconsin Breast Cancer (Diagnostic)* dataset (Street et al., 1993) (569 samples, 30 features).

As shown in Figure 6, the empirical error remains essentially flat, whereas the theoretical prediction exhibits a marked increase around $m = 17$. This discrepancy is caused by the lack of concentration in the mixed features, which leads to an overestimation of the variance term.

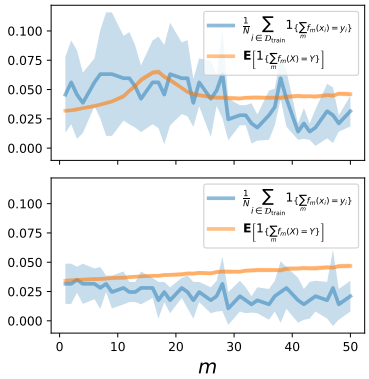

Figure 6: Breast Cancer dataset: empirical error (orange) vs. theory (blue).

7 CONCLUSION

We analyzed bootstrap-based ensemble methods for LSSVM using Random Matrix Theory, deriving the asymptotic distribution of the decision score and theoretical classification error. Our results highlight the trade-offs between classifier count, regularization, and data structure. Empirical results confirm the accuracy of our predictions and demonstrate the advantages of a closed-form approach. Future work includes (i) extending to non-linear models (*e.g.,* kernelized variants), (ii) multiclass extensions with calibrated outputs.

Acknowledgement

This work was supported by the German Research Foundation (DFG) as research unit DeSBi [KI-FOR 5363] (459422098), by the Research Council of Finland (Decision #363624) as *A Mathematical Theory of Trustworthy Federated Learning (MATHFUL)*, by the Jane and Aatos Erkko Foundation (Decision #A835) as *A Mathematical Theory of Federated Learning (TRUSTFELT)*, by the French National Research Agency (ANR) under grant ANR-24-CE40-3341 (project DECATTLON) and by Business Finland as *Forward-Looking AI Governance in Banking & Insurance (FLAIG)*. This work was granted access to the HPC resources of IDRIS under the allocation 2025-AD011016818, provided by GENCI.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with

specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

References

Ando, Y. and Komaki, F. (2023). On high dimensional asymptotics of model averaging. *Preprint arXiv:2308.09476*.

Bellec, P. C. et al. (2023). Corrected generalized cross-validation for finite ensembles of penalized estimators. *Preprint arXiv:2310.01374*.

Bellec, P. C. and Koriyama, T. (2024). Asymptotics of resampling without replacement in robust and logistic regression. *Preprint arXiv:2404.02070*. Submitted / preprint.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.

Bergstra, J., Yamins, D., and Cox, D. (2022). Hyperopt: Distributed asynchronous hyper-parameter optimization. *Astrophysics Source Code Library*.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, Prague, Czech Republic.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2).

Clarté, L., Vandenbroucque, A., Dalle, G., Loureiro, B., Krzakala, F., and Zdeborová, L. (2024). Analysis of bootstrap and subsampling in high-dimensional regularized regression. *arXiv preprint arXiv:2402.13622*.

Cramer, A., Wu, H., Salamon, J., and Bello, J. P. (2019). Look, listen and learn more: Design choices for deep audio embeddings. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK.

Dezeure, R., Bühlmann, P., and Zhang, C. (2016). High dimensional simultaneous inference with the bootstrap. *Preprint arXiv:1606.03940*.

Du, J., Patil, P., and Kuchibhotla, H. (2023). Subsample ridge ensembles: Equivalences and generalized cross-validation. *Preprint arXiv:2304.13016*.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1).

Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC press.

El Karoui, N. and Purdom, E. (2018). Can we trust the bootstrap in high-dimensions? the case of linear models. *Journal of Machine Learning Research*, 19. Preprint arXiv:1608.00696.

El Karoui, N. and Purdom, E. (2019). The non-parametric bootstrap and spectral analysis in moderate and high-dimension. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89. PMLR.

Fleury, B., Guédon, O., and Paouris, G. (2007). A stability result for mean width of lp-centroid bodies. *Advances in Mathematics*, 214(2).

Freedman, D. A. (1981). Bootstrapping regression models. *The Annals of Statistics*, 9(6).

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10).

Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (1999). Spambase. UCI Machine Learning Repository.

Klartag, B. (2007). A central limit theorem for convex sets. *The Annals of Probability*, 35(4).

Koriyama, T. (2024). Precise asymptotics of bagging regularized m-estimators. *Preprint arXiv:2409.15252*.

Krogh, A. and Sollich, P. (1997). Statistical mechanics of ensemble learning. *Physical Review E*, 55(1).

Krogh, A. and Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 7.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2).

LeJeune, M., Javadi, S., and Baraniuk, R. (2019). Implicit regularization of ordinary least squares ensembles. *Preprint arXiv:1910.04743*.

Louart, C. and Couillet, R. (2018). Concentration of measure and large random matrices with an application to sample covariance matrices. *arXiv preprint arXiv:1805.08295*.

Loureiro, B., Gerbelot, C., Refinetti, M., Sicuro, G., and Krzakala, F. (2022). Fluctuations, bias, variance & ensemble of learners: Exact asymptotics for convex losses in high-dimension. In *International Conference on Machine Learning*. PMLR.

Perrone, M. P. and Cooper, L. N. (1993). Improving regression estimates: Averaging methods for variance reduction with extensions to general convex measure optimization. In *Proceedings of the Workshop on Computational Learning Theory and Natural Learning Systems*.

Seddik, M. E. A., Louart, C., Tamaazousti, M., and Couillet, R. (2020). Random matrix theory proves that deep learning representations of gan-data behave as gaussian mixtures. In *International Conference on Machine Learning*. PMLR.

Singh, K. (1981). On the asymptotic accuracy of efron's bootstrap. *The Annals of Statistics*.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*.

Sollich, P. and Krogh, A. (1995). Learning with ensembles: How overfitting can be useful. In *Advances in Neural Information Processing Systems (NeurIPS 8)*.

Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. *Biomedical Image Processing and Biomedical Visualization*, 1905.

Takahashi, T. (2024). A replica analysis of underbagging. *Transactions on Machine Learning Research (TMLR)*.

Takahashi, T. (2025). Replica analysis for ensemble techniques in variable selection. *Journal of the Physical Society of Japan*, 94(3).

Tiomoko, M., Couillet, R., and Tiomoko, H. (2020). Large dimensional analysis and improvement of multi task learning. *arXiv preprint arXiv:2009.01591*.

Supplementary Materials for High-Dimensional Analysis of Bootstrap Ensemble Classifiers

Abstract

This supplementary material complements the main paper by providing (i) additional theoretical details on the bootstrap Least Squares Support Vector Machine (LSSVM) classifier, and (ii) an extensive review of related work.

On the theoretical side, we revisit in Section 9 the bootstrap formulation, where resampled subsets are used to train independent LSSVM classifiers. Section 10 develops the Random Matrix Theory (RMT) analysis in detail: we study the distribution of the decision score (10.2), compute its mean and variance (10.3, 10.4), and derive the asymptotic classification error (10.5). Additional interpretations and intuitive explanations are provided in Section 11.

On the literature side, Section 8 expands the related work discussion from the main paper. We provide a comparative table summarizing key contributions across high-dimensional bootstrap, bagging, and ensemble theory, and explicitly contrast them with our work on bootstrap classification ensembles.

8 Extensive Related Works

For completeness, we provide here a structured comparison of representative works on bootstrap and ensemble methods in high-dimensional regimes. This complements the discussion in the main paper by offering a compact reference table that situates our contribution relative to prior results in regression, spectral analysis, and inference.

9 Problem formulation

9.1 The LSSVM Classifier

We consider a binary classification problem where each observation $\mathbf{x} \in \mathbb{R}^d$ is described by d features, and the output space consists of two classes: $\mathcal{C}_1$ and $\mathcal{C}_2$. The training set consists of n labeled examples $(\mathbf{X}, \mathbf{y}) = (\mathbf{x}_i, y_i)_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the features of the i-th sample, and $y_i \in \{-1, +1\}$ denotes the corresponding class label.

The objective within the Least Squares Support Vector Machine (LSSVM) framework is to find the optimal separating hyperplane that best divides the two classes. This is achieved by solving the following optimization problem:

$$\min_{\boldsymbol{\omega} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \boldsymbol{\omega}^\top \mathbf{x}_i)^2 + \frac{\lambda}{2} \|\boldsymbol{\omega}\|_2^2, \tag{4}$$

where $\lambda > 0$ is the regularization parameter that controls the trade-off between the goodness of fit to the training data and the smoothness of the solution. The optimization problem in Equation (4) is convex and admits a unique closed-form solution for $\boldsymbol{\omega}^\star \in \mathbb{R}^d$, which is given by (for convenience we provide its transpose, as we will consider $\boldsymbol{\omega}^{\star\top} \mathbf{x}$)

$$\boldsymbol{\omega}^{\star\top} = \frac{1}{n} \mathbf{y}^\top \mathbf{X}^\top \left(\frac{\mathbf{X}\mathbf{X}^\top}{n} + \lambda \mathbf{I}_d \right)^{-1}, \tag{5}$$

where $\mathbf{X} \in \mathbb{R}^{d \times n}$ is the data matrix of training samples, and $\mathbf{y} \in \mathbb{R}^n$ is the vector of class labels for these samples. Once the optimal hyperplane is determined, for a new test sample $\mathbf{x} \in \mathbb{R}^d$, the classification decision is made

Work (ref)	Dim. regime	Target	Object / result	Contrast to our work
El Karoui and Purdom (2018, 2019)	Proportional	Regression / Spectral	Bootstrap and jackknife inconsistency; failures for spectral statistics	Diagnostic negatives; we provide constructive deterministic equivalents for classification scores.
Clarté et al. (2024)	Proportional	Reg./GLM	Validity conditions for bootstrap, subsampling, and jackknife	Regression focus; we derive classification error maps.
Bellec and Koriyama (2024)	Proportional	Robust/Logit	Fixed-point correlation equations for subsampling	Regression models; we analyze bootstrap LSSVM ensembles.
Koriyama (2024)	Proportional	M-estimation	Risk formulas for bagged M-estimators	Prediction-risk focus; we target classification error.
Bellec et al. (2023)	Proportional	Penalized LS ensembles	Corrected GCV restoring consistency	Risk-estimation focus; we construct analytic error maps.
Du et al. (2023)	Proportional	Ridge ensembles	Ridge–subsampling equivalences; uniform GCV consistency	Ridge ensembles; we treat LSSVMs with general covariance.
LeJeune et al. (2019)	High-dim	OLS ensembles	Implicit ridge via subsample averaging	Implicit reg. in OLS; we quantify ensemble effects on LSSVM scores.
Ando and Komaki (2023)	Proportional	Model averaging	Optimal weights; double-descent phenomena	Model averaging; we derive closed-form classification error maps.
Dezeure et al. (2016)	High-dim sparse	Sparse regression inference	Bootstrap-based simultaneous inference	Sparse inference; our focus is non-sparse classification ensembles.
This work	**Proportional, general covariance**	**Classification (LSSVM)**	**Deterministic equivalents for score mean/var; closed-form error map**	Constructive tuning of λ and m for classification ensembles.

Table 1: Representative related works organized by regime, target, and results, contrasted with our contribution. "Proportional" denotes $n, d \to \infty$ with $d/n \to \gamma \in (0, \infty)$.

based on the sign of the decision function $\mathbf{x}^\top \boldsymbol{\omega}^\star$, geometrically corresponding to which side of the hyperplane the test point falls into. Specifically,

$$\text{Class}(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if } \boldsymbol{\omega}^{\star\top}\mathbf{x} < 0, \\ \mathcal{C}_2 & \text{if } \boldsymbol{\omega}^{\star\top}\mathbf{x} > 0. \end{cases}$$

9.2 Bootstrapping in LSSVM

Having defined the general framework for Least Squares Support Vector Machines (LSSVM), we now consider the scenario where bootstrapping is applied to improve the robustness of the model. Bootstrapping in the context of LSSVM involves partitioning the entire dataset $\mathbf{X}$ into m subsets. Each subset is then used to train an independent LSSVM classifier. The final decision function is obtained by averaging the decision functions of all individual classifiers, thus combining their predictions for a more stable and accurate outcome.

Let $\{\mathbf{X}_i\}_{i=1}^m$ represent the subsets m, where each subset $\mathbf{X}_i \in \mathbb{R}^{d \times n_i}$ contains n_i samples. Each subset $\mathbf{X}_i$ is associated with a label vector $\mathbf{y}_i \in \{-1, +1\}^{n_i}$. For simplicity, we assume that all subsets contain the same number of elements, denoted n_m. Each bootstrapped classifier $\boldsymbol{\omega}_i$ is trained on a subset $\mathbf{X}_i$ using the LSSVM optimization problem. The decision function for each classifier is given by:

$$\boldsymbol{\omega}_i^\top = \frac{1}{n_i}\mathbf{y}_i^\top \mathbf{X}_i^\top \left(\frac{\mathbf{X}_i \mathbf{X}_i^\top}{n_i} + \lambda_i \mathbf{I}_d \right)^{-1},$$

where $\mathbf{X}_i \in \mathbb{R}^{d\times n_i}$ and $\mathbf{y}_i \in \mathbb{R}^{n_i}$ represent the data and label vectors for the i-th subset, respectively.

Once individual classifiers are trained, the final classifier is obtained by averaging the decision functions of all bootstrapped classifiers. This ensemble approach helps reduce variance and improve the generalization of the model. At the heart of this procedure lies the ability to evaluate the performance of the bootstrapped classifier. Specifically, we define the *classification accuracy* as the proportion of correctly classified test samples, and it is given by:

$$\text{Accuracy} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \mathbb{I}\left(\hat{\mathbf{y}}_i \neq \mathbf{y}_i\right),$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\mathbf{x}_i$ is the i-th test sample, y_i is the true label, and $\hat{\mathbf{y}}_i$ is the predicted label of the bootstrapped classifier.

Our aim in this section is to understand the theoretical behavior of the classification accuracy as a function of the number of bootstrapped classifiers m, the regularization parameter λ_i, and other parameters in the model. Specifically, we seek to derive insights into how the number of bootstrapped classifiers impacts the accuracy, and how the averaging of decision functions can help reduce overfitting and improve predictive performance.

10 Theoretical analysis of bootstrapping in LSSVM

In order to predict the theoretical classification error of the bootstrapped LSSVM model, we must first gain a deeper understanding of the statistical behavior of the decision score $g(\mathbf{x})$. The decision score is the key quantity that determines the classification decision for any given test sample. To analyze its effect on classification performance, it is crucial to examine its statistical distribution. This involves computing the first- and second-order moments of $g(\mathbf{x})$, as these will provide the foundation for deriving the theoretical classification error.

To do this, we will use the following assumptions on the data distribution

10.1 Data assumptions and growth rate assumption

To perform a theoretical analysis of bootstrapping in LSSVM, we make a few key assumptions about the data. These assumptions will help us derive statistical properties of the decision score and estimate the classification error.

First, we assume a double asymptotic regime in which both the number of samples n and the dimension d grow large. This is a common approach in high-dimensional statistics and random matrix theory.

Assumption 10.1 (Large n, large d) We assume that $n > d$, and as both n and d increase, the ratio d/n approaches a constant $c_0 \in (0, 1)$. Furthermore, the fraction of samples in each bootstrap subset, n_ℓ/n, tends to a constant $c_\ell > 0$ for each subset $\ell = 1, 2$. We further assume that the number of subsets does not grow with n, *i.e.*, we assume $m = \mathcal{O}(1)$.

Next, we assume that the data matrix $\mathbf{X}$ and its columns $\mathbf{x}$ (corresponding to individual samples) exhibit a strong concentration, a realistic and mathematically convenient assumption. Intuively, this means that any Lipschitz-continuous observations are tightly clustered around their means with high probability, which also helps to control the behavior of the decision scores based on inner products like $\mathbf{x}^\top \mathbf{w}$.

Assumption 10.2 (Data Concentration) We assume that the data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ and its columns $\mathbf{x} \in \mathbb{R}^d$ follow a q-exponential concentration property. Specifically, we assume:

$$\mathbf{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^d, \|\cdot\|_2), \qquad \mathbf{X} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^{d\times n}, \|\cdot\|_F),$$

where $\mathcal{E}_2$ denotes 2-exponential concentration with respect to the Euclidean and Frobenius norms; see Definition 10.1.

As a major example, consider $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for which $\mathbf{x} \propto \mathcal{E}_2(1 \,|\, \mathbb{R}^d, \|\cdot\|_2)$, *i.e.*, the observable diameter is independent of the dimension. Notably, let us also recall that concentration is preserved under Lipschitz-continuous transformations.

Definition 10.1 (q-exponential concentration; observable diameter) Let $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ be a normed vector space and $q > 0$. A random vector $\mathbf{x} \in \mathcal{X}$ is said to be q-exponentially concentrated, if for any 1-Lipschitz continuous (with respect to $\|\cdot\|_{\mathcal{X}}$) real-valued function $\varphi : \mathcal{X} \to \mathbb{R}$ there exists $C \geq 0$ independent of $\dim(\mathcal{X})$ and $\sigma > 0$ such that

$$\mathbb{P}\left(|\varphi(\mathbf{x}) - \mathbb{E}\varphi(\mathbf{x})| \geq t\right) \leq Ce^{-(t/\sigma)^q} \qquad \forall t > 0.$$

This is denoted as $\mathbf{x} \propto \mathcal{E}_q(\sigma \,|\, \mathcal{X}, \|\cdot\|_{\mathcal{X}})$, where σ is called the *observable diameter*. If σ does not depend on $\dim(\mathcal{X})$, we write $\mathbf{x} \propto \mathcal{E}_q(1 \,|\, \mathcal{X}, \|\cdot\|_{\mathcal{X}})$.

These assumptions on concentration and growth rates allow us to model the behavior of the decision score and ultimately estimate the classification error of the bootstrapped LSSVM. We also impose the condition that the columns of $\mathbf{X}$ are independent. These assumptions ensure the existence of both the mean and covariance for the columns of $\mathbf{X}$. For $\ell \in \{1, 2\}$, we define the following

$$\boldsymbol{\mu}_\ell = \mathbb{E}[\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{C}_\ell], \quad \boldsymbol{\Sigma}_\ell = \mathrm{Cov}(\mathbf{x}_i \mid \mathbf{x}_i \in \mathcal{C}_\ell)$$

Additionally, we collect the means of the class distributions into the matrix $\mathbf{M} = [\boldsymbol{\mu}_1 | \boldsymbol{\mu}_2] \in \mathbb{R}^{d \times 2}$ and we define the generalized covariance as $\mathbf{C}_\ell = \boldsymbol{\Sigma}_\ell + \boldsymbol{\mu}_\ell \boldsymbol{\mu}_\ell^\top$.

10.2 Distribution of the decision score

To demonstrate that the decision score $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ follows a Gaussian distribution, we rely on the **Central Limit Theorem (CLT)** for concentrated random vectors. This result, presented in (Klartag, 2007) and (Fleury et al., 2007), shows that the inner product $\mathbf{w}^\top \mathbf{x}$, where $\mathbf{x}$ is a random vector satisfying q-exponential concentration, approaches a Gaussian distribution.

The key conditions for the result are:

- $\mathbf{x}$ is a random vector with zero mean $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and identity covariance $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{I}_p$.
- $\mathbf{x}$ satisfies q-exponential concentration, meaning that the distribution of $\mathbf{x}$ is tightly concentrated around its mean, following the concentration property from Definition 10.1.
- The weight vector $\mathbf{w}$ is randomly chosen and independent of $\mathbf{x}$.

Central Limit Theorem: Under these conditions, for sufficiently large p, the distribution of the inner product $\mathbf{w}^\top \mathbf{x}$ (which is $g(\mathbf{x})$) is approximately Gaussian. Specifically, it converges to a standard normal distribution $\mathcal{N}(0, 1)$, with a small deviation bound for the cumulative distribution function $F(t)$, ensuring that the approximation is accurate up to an error of $p^{-1/4}$.

This result justifies the assumption that $g(\mathbf{x})$ behaves like a Gaussian random variable, which allows us to compute the theoretical classification error more precisely using numerical methods such as integration.

Furthermore, we define the notion of deterministic equivalent as

Definition 10.2 (Deterministic equivalents) A deterministic equivalent, say $\bar{\mathbf{F}} \in \mathbb{R}^{n \times d}$, of a given random matrix $\mathbf{F} \in \mathbb{R}^{n \times d}$, denoted $\mathbf{F} \leftrightarrow \bar{\mathbf{F}}$, is defined by the fact that, for any deterministic linear functional $f : \mathbb{R}^{n \times d} \to \mathbb{R}$, $f(\mathbf{F} - \bar{\mathbf{F}}) \to 0$ almost surely (for instance, for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ of unit norm, $\mathbf{u}^\top(\mathbf{F} - \bar{\mathbf{F}})\mathbf{v} \to 0$ and, for $\mathbf{A} \in \mathbb{R}^{d \times n}$ deterministic of bounded operator norm, $\frac{1}{n} \operatorname{tr} \mathbf{A}(\mathbf{F} - \bar{\mathbf{F}}) \to 0$).

Deterministic equivalents are thus particularly suitable to handle bilinear forms involving the random matrix $\mathbf{F}$.

10.3 Mean of the Decision Score

For a test data point $\mathbf{x} \in \mathbb{R}^d$ from class $\mathcal{C}_\ell$, where $\ell = 1, 2$, and assuming $\mathbf{x}$ is independent from the training data $\mathbf{X} \in \mathbb{R}^{d \times n}$, the mean classification score $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ can be computed as follows:

$$
\begin{aligned}
m_\ell =& \frac{1}{m}\sum_{t=1}^{m} \mathbb{E}_{\mathbf{X}_t}\left[\mathbb{E}_{\mathbf{x}}\left[g(\mathbf{x}) \mid \mathbf{x}\in\mathcal{C}_\ell\right]\right] && (6)\\
=& \frac{1}{m}\sum_{t=1}^{m}\frac{1}{n}\mathbb{E}_{\mathbf{X}_t}\left[\mathbb{E}_{\mathbf{x}}\left[\mathbf{y}_t^\top\mathbf{X}_t^\top\left(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^\top+\lambda\mathbf{I}_d\right)^{-1}\mathbf{x}\right]\right] && (7)\\
=& \frac{1}{m}\sum_{t=1}^{m}\frac{1}{n}\mathbb{E}_{\mathbf{X}_t}\left[\mathbf{y}_t^\top\mathbf{X}_t^\top\left(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^\top+\lambda\mathbf{I}_d\right)^{-1}\boldsymbol{\mu}_\ell\right] && (8)\\
=& \frac{1}{m}\sum_{t=1}^{m}\frac{1}{n}\mathbb{E}_{\mathbf{X}_t}\left[\mathbf{y}_t^\top\mathbf{X}_t^\top\mathbf{Q}_t\boldsymbol{\mu}_\ell\right] && (9)\\
=& \frac{1}{n}\mathbf{y}_t^\top\mathbf{J}\mathbf{M}_\delta^\top\bar{\mathbf{Q}}\boldsymbol{\mu}_\ell && (10)
\end{aligned}
$$

Where $\bar{\mathbf{Q}}=\bar{\mathbf{Q}}(\lambda)$ is the deterministic equivalent of $\mathbf{Q}=\left(\frac{1}{n}\mathbf{X}\mathbf{X}^\top+\lambda\mathbf{I}_d\right)^{-1}$, the resolvent of the random matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^\top$. The matrices $\mathbf{J}\in\mathbb{R}^{n\times 2}$ and $\mathbf{M}_\delta\in\mathbb{R}^{d\times 2}$ are defined as:

$$
\mathbf{J}=\begin{pmatrix}\mathbb{1}_{n_1} & \mathbf{0}_{n_1}\\ \mathbf{0}_{n_2} & \mathbb{1}_{n_2}\end{pmatrix}\in\mathbb{R}^{n\times 2},\quad \mathbf{M}_\delta=\left[\frac{1}{1+\delta_1}\boldsymbol{\mu}_1\middle|\frac{1}{1+\delta_2}\boldsymbol{\mu}_2\right]\in\mathbb{R}^{d\times 2}
$$

where $\boldsymbol{\delta}=(\delta_1,\delta_2)$. The matrix $\bar{\mathbf{Q}}$ is computed iteratively using the fixed-point equation:

$$
\bar{\mathbf{Q}}=\left(\sum_{\ell=1}^{2}\frac{n_\ell}{n}\frac{\mathbf{C}_\ell}{1+\delta_\ell}+\lambda\mathbf{I}_d\right)^{-1},\quad \delta_\ell=\frac{1}{n}\operatorname{Tr}\left(\mathbf{C}_\ell\bar{\mathbf{Q}}\right)
$$

The iterations begin with $\delta_1=\delta_2=0$ and continue until convergence.

Derivation Details - From (6) to (7): We apply the definition of the decision score $g(\mathbf{x})=\mathbf{y}^\top\mathbf{X}_t^\top\left(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^\top+\lambda\mathbf{I}_d\right)^{-1}\mathbf{x}$.

- From (7) to (8): We use the independence assumption between the training samples $\mathbf{X}_t$ and the test sample $\mathbf{x}$. Therefore, the expectation is taken only with respect to the test sample, yielding $\mathbb{E}[\mathbf{x}|\mathbf{x}\in\mathcal{C}_\ell]=\boldsymbol{\mu}_\ell$.

- From (8) to (9): This is a rewriting step where we define the resolvent matrix $\mathbf{Q}_t=\left(\frac{1}{n}\mathbf{X}_t\mathbf{X}_t^\top+\lambda\mathbf{I}_d\right)^{-1}$.

- From (9) to (10): This step involves some technical matrix algebra and tools from Random Matrix Theory, which is explained as follows.

Let $\mathbf{X}_{-i}$ represent the matrix $\mathbf{X}$ with the i-th column replaced by $\mathbf{0}$, such that $\mathbf{X}\mathbf{X}^\top=\mathbf{X}_{-i}\mathbf{X}_{-i}^\top+\mathbf{x}_i\mathbf{x}_i^\top$. Using the *Sherman-Morrison matrix inversion lemma*, we can express $\mathbf{Q}$ as:

$$
\mathbf{Q}=\left(\frac{\mathbf{X}\mathbf{X}^\top}{n}+\lambda\mathbf{I}_d\right)^{-1}=\mathbf{Q}_{-i}-\frac{1}{n}\frac{\mathbf{Q}_{-i}\mathbf{x}_i\mathbf{x}_i^\top\mathbf{Q}_{-i}}{1+\frac{1}{n}\mathbf{x}_i^\top\mathbf{Q}_{-i}\mathbf{x}_i}
$$

Next, we can calculate the interaction between $\mathbf{Q}$ and $\mathbf{x}_i$:

$$
\mathbf{Q}\mathbf{x}_i=\frac{\mathbf{Q}_{-i}\mathbf{x}_i}{1+\frac{1}{n}\mathbf{x}_i^\top\mathbf{Q}_{-i}\mathbf{x}_i}
$$

Now, we compute the expectation of the decision score for a given test sample $\mathbf{x}_i$:

$$
\mathbb{E}\left[\mathbf{y}_t\mathbf{x}_i^\top\mathbf{Q}\boldsymbol{\mu}_\ell\right]=\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\mathbf{y}_i\frac{\mathbf{x}_i^\top\mathbf{Q}_{-i}\boldsymbol{\mu}_\ell}{1+\frac{1}{n}\mathbf{x}_i^\top\mathbf{Q}_{-i}\mathbf{x}_i}\right]
$$

This simplifies to:

$$\frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{y}_i\frac{\mathbb{E}[\mathbf{x}_i]^\top\bar{\mathbf{Q}}\boldsymbol{\mu}_\ell}{1+\delta_{\mathcal{C}(i)}}\right]$$

where $\bar{\mathbf{Q}}$ is the *deterministic equivalent* of $\mathbf{Q}$ (recall Definition 10.2 above) and provided in (Louart and Couillet, 2018) as

$$\bar{\mathbf{Q}}=\left(\sum_{\ell=1}^{2}\frac{n_\ell}{n}\frac{\mathbf{C}_\ell}{1+\delta_\ell}+\lambda\mathbf{I}_d\right)^{-1},\qquad \delta_\ell=\frac{1}{n}\operatorname{Tr}\left(\mathbf{C}_\ell\bar{\mathbf{Q}}\right).$$

This completes the derivation for the mean of the decision score.

10.4 Variance of the decision score

Next, we compute the variance σ_ℓ^2 of the classification score $g(\mathbf{x})=\mathbf{x}^\top\mathbf{w}$ for a test dataset $\mathbf{x}\in\mathcal{C}_\ell$:

$$\begin{aligned}\sigma_\ell^2 =&\operatorname{Var}(g(\mathbf{x}))\\ =&\mathbb{E}\left[g(\mathbf{x})^2\right]-\left(\mathbb{E}[g(\mathbf{x})]\right)^2\\ =&\mathbb{E}\left[\mathbf{y}^\top\mathbf{X}^\top\mathbf{Q}\mathbf{x}\mathbf{x}^\top\mathbf{Q}\mathbf{X}\mathbf{y}\right]-\left(\mathbb{E}[g(\mathbf{x})]\right)^2\\ =&\mathbf{y}^\top\mathbf{J}\mathbf{M}_\delta^\top\mathbf{K}_\ell\mathbf{M}_\delta\mathbf{J}^\top\mathbf{y}+\mathbf{y}^\top\mathbf{V}_\ell\mathbf{y}-2\mathbf{y}^\top\mathbf{J}\mathbf{M}_{\boldsymbol{\delta}'}^\top\bar{\mathbf{Q}}\mathbf{M}_\delta\mathbf{J}^\top\mathbf{y}\end{aligned}\tag{11}$$

where $\mathbf{K}_\ell$, $\mathbf{V}_\ell$, and other necessary terms are given as follows:

$$\tilde{\mathbf{V}}=\frac{1}{n}\begin{pmatrix}\operatorname{Tr}(\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}) & \operatorname{Tr}(\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}})\\ \operatorname{Tr}(\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}) & \operatorname{Tr}(\boldsymbol{\Sigma}_2\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}})\end{pmatrix}\in\mathbb{R}^{2\times 2},\tag{12}$$

$$\tilde{\mathbf{A}}=\frac{1}{n}\begin{pmatrix}\frac{n_1}{(1+\delta_1)^2} & 0\\ 0 & \frac{n_2}{(1+\delta_2)^2}\end{pmatrix}\in\mathbb{R}^{2\times 2},\tag{13}$$

$$\bar{\mathbf{t}}^{(\ell)}=\frac{1}{n}\left[\operatorname{Tr}(\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}\boldsymbol{\Sigma}_1\bar{\mathbf{Q}}),\operatorname{Tr}(\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}\boldsymbol{\Sigma}_2\bar{\mathbf{Q}})\right]^\top\in\mathbb{R}^2,\tag{14}$$

$$\mathbf{d}^{(\ell)}=\left[d_1^{(\ell)},d_2^{(\ell)}\right]^\top=\left(\mathbf{I}_2-\tilde{\mathbf{V}}\tilde{\mathbf{A}}\right)^{-1}\bar{\mathbf{t}}^{(\ell)}\in\mathbb{R}^2,\tag{15}$$

$$\mathbf{K}_\ell=\bar{\mathbf{Q}}\boldsymbol{\Sigma}_\ell\bar{\mathbf{Q}}+\sum_{\ell'=1}^{2}\frac{n_{\ell'}}{n}\frac{d_{\ell'}^{(\ell)}}{(1+\delta_{\ell'})^2}\bar{\mathbf{Q}}\mathbf{C}_{\ell'}\bar{\mathbf{Q}},\tag{16}$$

$$\mathbf{V}_\ell=\mathcal{D}_{\mathbf{v}_\ell}\in\mathbb{R}^{n\times n},\tag{17}$$

$$\mathbf{v}_\ell=\left[\frac{\operatorname{Tr}\left(\boldsymbol{\Sigma}_1\mathbf{K}_\ell\right)}{(1+\delta_1)^2}\mathbb{1}_{n_1},\frac{\operatorname{Tr}\left(\boldsymbol{\Sigma}_2\mathbf{K}_\ell\right)}{(1+\delta_2)^2}\mathbb{1}_{n_2}\right]\in\mathbb{R}^n\tag{18}$$

The derivation of the variance follows the same structure as that of the mean, but now the goal is to find the deterministic equivalent of the matrices $\mathbf{QAQ}$, where $\mathbf{A}$ is an arbitrary matrix independent of $\mathbf{Q}$ and has bounded norms. The necessary calculations can be carried out in a similar manner to those in (Tiomoko et al., 2020) (Section A.2).

10.5 Derivation of the theoretical error

After determining the class-specific means and variances of the classification score $g(\mathbf{x})=\mathbf{w}^\top\mathbf{x}$ using Section 10.3 and 10.4, we can now compute the theoretical classification error, summarized in the following remark.

Remark 10.1 (Classification Accuracy) For a linear binary classifier $\mathbf{w} \in \mathbb{R}^d$, where the classification score $g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ follows normal distributions for each class, that is, $\mathcal{N}(m_1, \sigma_1^2)$ for $\mathbf{x} \in \mathcal{C}_1$ and $\mathcal{N}(m_2, \sigma_2^2)$ for $\mathbf{x} \in \mathcal{C}_2$, the classification error is given by

$$\varepsilon = c_1 \cdot \mathbb{P}\left(\mathbf{x} \to \mathcal{C}_2 \mid \mathbf{x} \in \mathcal{C}_1\right) + c_2 \cdot \mathbb{P}\left(\mathbf{x} \to \mathcal{C}_1 \mid \mathbf{x} \in \mathcal{C}_2\right),$$

where c_ℓ denotes the probabilities of the class and $c_1 = c_2 = \frac{1}{2}$ for balanced classes. The probabilities of misclassification for each class can be expressed as:

$$\varepsilon = c_1 \cdot \mathbb{P}\left(\mathbf{w}^\top \mathbf{x} > \eta \mid \mathbf{x} \in \mathcal{C}_1 \sim \mathcal{N}(m_1, \sigma_1^2)\right) + c_2 \cdot \mathbb{P}\left(\mathbf{w}^\top \mathbf{x} < \eta \mid \mathbf{x} \in \mathcal{C}_2 \sim \mathcal{N}(m_2, \sigma_2^2)\right),$$

where η is the decision threshold. These misclassification probabilities are determined by the cumulative distribution functions (CDF) of the normal distributions:

$$\varepsilon = c_1 \cdot \mathbb{P}\left(X > \eta \mid X \sim \mathcal{N}(m_1, \sigma_1^2)\right) + c_2 \cdot \mathbb{P}\left(Y < \eta \mid Y \sim \mathcal{N}(m_2, \sigma_2^2)\right).$$

For practical computation, these probabilities can be approximated using numerical integration of the normal distribution densities, given the class-specific means m_1, m_2 and variances σ_1^2, σ_2^2, and the threshold η.

The optimal threshold η minimizes the classification error and is typically found between the two means, that is, $m_1 < \eta < m_2$, where the densities of the two classes intersect.

11 Intuitions from the theoretical analysis

In this section, we aim to derive several intuitions into the bootstrap LSSVM approach. First we analyze the influence of the mean of the decision score

11.1 Analysis of the mean of the decision score

We recall that the mean of the decision score given that the test data is coming from the class $\mathcal{C}_1$ can be written as

$$\mathfrak{m}_1 = \frac{1}{n_m} \mathbf{y}^\top \mathbf{J}^\top \mathbf{M}_\delta^\top \bar{\mathbf{Q}} \boldsymbol{\mu}_1$$

Furthermore we will consider first the simple case of mixture of two Gaussian of opposite mean $\pm\boldsymbol{\mu}$ and identity covariance matrix $\boldsymbol{\Sigma} = \mathbf{I}_d$. By rewriting the labels $\mathbf{y} = \mathbf{J}\tilde{\mathbf{y}}$, the mean can be rewritten as

$$\mathfrak{m}_1 = c_1 \boldsymbol{\mu}^\top \bar{\mathbf{Q}} \boldsymbol{\mu}$$

Furthermore the deterministic equivalent $\bar{\mathbf{Q}}$ can be rewritten as

$$\bar{\mathbf{Q}} = \kappa \mathbf{I}_d - \kappa^2 \frac{\boldsymbol{\mu}\boldsymbol{\mu}^\top}{(1+\delta) + \kappa \boldsymbol{\mu}^\top \boldsymbol{\mu}}$$

where δ is given by the closed form solution

$$\delta = \frac{\sqrt{(1+\lambda-c_0)^2 + 4\lambda c_0} - (1+\lambda-c_0)}{2\lambda}$$

Finally the mean of the decision score can be rewritten as

$$\mathfrak{m}_1 = \frac{\boldsymbol{\mu}^\top \boldsymbol{\mu}}{(1+\lambda+\lambda\delta) + \boldsymbol{\mu}^\top \boldsymbol{\mu}}$$

As the number of classifiers increases, the ratio $c_0 = p/n$ increases and therefore δ increases, making the mean to decrease. The mean only depends on the increasing number of classifier through c_0.

The parameter m, given by $m = \frac{\boldsymbol{\mu}^\top \boldsymbol{\mu}}{(1+\lambda+\lambda\delta)+\boldsymbol{\mu}^\top \boldsymbol{\mu}}$, can be interpreted as a normalized measure of the signal-to-noise ratio. Here, $\boldsymbol{\mu}^\top \boldsymbol{\mu}$ represents the squared magnitude of the mean vector $\boldsymbol{\mu}$, which measures the separation between the classes. A larger $\boldsymbol{\mu}^\top \boldsymbol{\mu}$ implies stronger discrimination between the two classes, leading to a larger $\mathfrak{m}_1$, while $\boldsymbol{\mu}^\top \boldsymbol{\mu} \to 0$ results in $m \to 0$, indicating no class separation. The term $1 + \lambda$ serves as a baseline regularization factor that prevents $\mathfrak{m}_1$ from growing excessively large, even for high $\boldsymbol{\mu}^\top \boldsymbol{\mu}$. The regularization parameter λ controls the trade-off between fitting the data and penalizing model complexity: smaller λ emphasizes data fidelity, whereas larger λ prioritizes simplicity and robustness. The term $\lambda\delta$, where δ increases with d/n, penalizes $\mathfrak{m}_1$ to account for the challenges of high-dimensional noise. This highlights the curse of dimensionality, as a larger d/n reduces $\mathfrak{m}_1$, reflecting the difficulty of distinguishing classes in high dimensions. Overall, m can be expressed as $\frac{\text{signal}}{\text{signal+noise}}$, where the signal is $\boldsymbol{\mu}^\top \boldsymbol{\mu}$ (the class separation), and the noise is $1+\lambda+\lambda\delta$ (accounting for regularization and dimensionality effects). When the signal dominates the noise, $\mathfrak{m}_1$ increaseses, signifying strong class discrimination, while noise dominance results in $\mathfrak{m}_1 \to 0$, indicating poor discrimination. This framework captures the interplay between signal strength, regularization, and dimensionality effects.

11.2 Analysis of the variance of the decision score

As performed previously, we will experiment also the variation of the variance of the decision score. The variance term can be splitted into four terms.

The first term of the variance is rigorously defined as

$$\mathcal{V}_1 = \frac{1}{n^2 m} \tilde{\mathbf{y}}^\top \mathbf{V} \tilde{\mathbf{y}}$$

where $\mathbf{V}$ is defined as the diagonal matrix containing on its diagonal the vector $\mathbf{v} = [\frac{1}{n}\operatorname{Tr}(\mathbf{C}_1\mathbf{Q}\mathbf{C}_1\mathbf{Q})\mathbb{1}_{n_1}, \frac{1}{n}\operatorname{Tr}(\mathbf{C}_2\mathbf{Q}\mathbf{C}_2\mathbf{Q})\mathbb{1}_{n_2}]$. The term can be rewritten in the more convenient manner as

$$\mathcal{V}_1 = \tilde{\mathbf{y}}^\top \mathcal{D}_{\mathbf{c}} \tilde{\mathbf{v}} \mathcal{D}_{\mathbf{c}} \tilde{\mathbf{y}}$$

Using the covariance matrix identity assumption, one can further get

$$\mathcal{V}_1 = \frac{c_0}{(1+\lambda+\lambda\delta)^2 - c_0}$$

The variance of $g(x)$ is composed of four terms:

$$\operatorname{Var}(g(x)) = \text{term}_1 + \text{term}_2 + \text{term}_3 + \text{term}_4,$$

where:

$$\begin{aligned}
\text{term}_1 &= \frac{c_0}{((1+\lambda+\lambda\delta)^2 - c_0)\, m}, \\
\text{term}_2 &= \frac{1}{m}\left(A_1 \boldsymbol{\mu}^\top \boldsymbol{\mu} + A_2(\boldsymbol{\mu}^\top \boldsymbol{\mu})^2 + A_3(\boldsymbol{\mu}^\top \boldsymbol{\mu})^3 + A_4(\boldsymbol{\mu}^\top \boldsymbol{\mu})^4\right), \\
\text{term}_3 &= -\frac{2}{m}\frac{\Delta_0\left(\kappa \boldsymbol{\mu}^\top \boldsymbol{\mu} - \gamma(\boldsymbol{\mu}^\top \boldsymbol{\mu})^2\right)}{(1+\delta)^3}, \\
\text{term}_4 &= \frac{m(m-1)}{m^2}\frac{\kappa^2 \boldsymbol{\mu}^\top \boldsymbol{\mu} - 2\gamma\kappa(\boldsymbol{\mu}^\top \boldsymbol{\mu})^2 + \gamma^2(\boldsymbol{\mu}^\top \boldsymbol{\mu})^3}{(1+\delta)^2}.
\end{aligned}$$

We will simplify each term step by step, and then derive the intuition.

Step 1: Simplify Each Term

Term 1:

$$\text{term}_1 = \frac{c_0}{((1+\lambda+\lambda\delta)^2 - c_0)\, m}.$$

This term reflects the impact of regularization (λ) and the effective dimensionality (δ). For large m, this term diminishes due to the $1/m$ factor.

Term 2:

$$\text{term}_2 = \frac{1}{m}\left(A_1\boldsymbol{\mu}^\top\boldsymbol{\mu} + A_2(\boldsymbol{\mu}^\top\boldsymbol{\mu})^2 + A_3(\boldsymbol{\mu}^\top\boldsymbol{\mu})^3 + A_4(\boldsymbol{\mu}^\top\boldsymbol{\mu})^4\right),$$

where:

$$A_1 = \frac{\kappa^2 S}{(1+\delta)^2}, \quad A_2 = -\frac{2\gamma\kappa S}{(1+\delta)^2} + \frac{\kappa^2 d_{\text{test}}}{(1+\delta)^4},$$
$$A_3 = \frac{\gamma^2 S}{(1+\delta)^2} - \frac{2\kappa\gamma d_{\text{test}}}{(1+\delta)^4}, \quad A_4 = \frac{\gamma^2 d_{\text{test}}}{(1+\delta)^4}.$$

Here, $S = 1 + \frac{d_{\text{test}}}{(1+\delta)^2}$ and $d_{\text{test}} = \frac{c_0\kappa^2(1+\delta)^2}{(1+\delta)^2 - c_0\kappa^2}$ where $\kappa = \frac{1}{\frac{1}{1+\delta}+\lambda}$ and $\gamma = \frac{\kappa^2}{(1+\delta)+\kappa\boldsymbol{\mu}^\top\boldsymbol{\mu}}$. Each term in A_1, A_2, A_3, A_4 decays with increasing δ or λ, reducing the influence of high-order terms in $\boldsymbol{\mu}^\top\boldsymbol{\mu}$.

Term 3:

$$\text{term}_3 = -\frac{2}{m}\frac{\Delta_0\left(\kappa\boldsymbol{\mu}^\top\boldsymbol{\mu} - \gamma(\boldsymbol{\mu}^\top\boldsymbol{\mu})^2\right)}{(1+\delta)^3}.$$

This term is dominated by Δ_0 and decreases inversely with m. For large δ, this term becomes negligible due to the $(1+\delta)^{-3}$ scaling.

Term 4:

$$\text{term}_4 = \frac{m(m-1)}{m^2}\frac{\kappa^2\boldsymbol{\mu}^\top\boldsymbol{\mu} - 2\gamma\kappa(\boldsymbol{\mu}^\top\boldsymbol{\mu})^2 + \gamma^2(\boldsymbol{\mu}^\top\boldsymbol{\mu})^3}{(1+\delta)^2}.$$

As $m \to \infty$, the factor $\frac{m(m-1)}{m^2} \to 1$. This term captures the interaction between subsets, primarily governed by κ and γ. For large δ, it decays due to $(1+\delta)^{-2}$.

Step 2: Combine Terms and Simplify

Combining all terms yields:

$$\begin{aligned}\text{Var}(g(x)) =& \frac{1}{m}\left[c_0\frac{1}{(1+\lambda+\lambda\delta)^2 - c_0} + \sum_{k=1}^{4} A_k(\boldsymbol{\mu}^\top\boldsymbol{\mu})^k\right] \\ &- \frac{2}{m}\frac{\Delta_0\left(\kappa\boldsymbol{\mu}^\top\boldsymbol{\mu} - \gamma(\boldsymbol{\mu}^\top\boldsymbol{\mu})^2\right)}{(1+\delta)^3} + \frac{m-1}{m}\frac{\kappa^2\boldsymbol{\mu}^\top\boldsymbol{\mu} - 2\gamma\kappa(\boldsymbol{\mu}^\top\boldsymbol{\mu})^2 + \gamma^2(\boldsymbol{\mu}^\top\boldsymbol{\mu})^3}{(1+\delta)^2}.\end{aligned}$$

For large m, terms proportional to $1/m$ dominate initially, while for small m, the interaction term (term_4) becomes significant.

Step 3: Derive Intuition

- **Dependence on m:** The variance decreases with m due to the $1/m$ scaling of terms like $\text{term}_1, \text{term}_2, \text{term}_3$. However, the interaction term (term_4) introduces diminishing returns for large m.
- **Role of Regularization (λ):** Higher λ stabilizes the variance by increasing the denominator of term_1 and reducing the magnitudes of A_1, A_2, A_3, A_4. This effect is particularly strong for weak signals ($\boldsymbol{\mu}^\top\boldsymbol{\mu} \to 0$).
- **Dimensionality Effect (δ):** As $\delta \to \infty$, terms like $(1+\delta)^{-k}$ suppress the variance, reflecting the curse of dimensionality. However, this decay can also mask the signal for large $\boldsymbol{\mu}^\top\boldsymbol{\mu}$.
- **Signal Strength ($\boldsymbol{\mu}^\top\boldsymbol{\mu}$):** Strong signal ($\boldsymbol{\mu}^\top\boldsymbol{\mu} \to \infty$) amplifies higher-order terms in term_2 and term_4, emphasizing the model's ability to discriminate between classes. Weak signals are dominated by regularization and noise.

The variance $\mathrm{Var}(g(x))$ depends intricately on the interplay between m, λ, δ, and $\boldsymbol{\mu}^\top \boldsymbol{\mu}$. Increasing m reduces variance initially but exhibits diminishing returns due to the interaction term. Strong regularization (λ) and high dimensionality (δ) stabilize the variance, though they may suppress meaningful signal ($\boldsymbol{\mu}^\top \boldsymbol{\mu}$).