Boosting for high-dimensional linear models

Kristian Gjika, Sergio Leal, Pablo Pastor, Nicolò Santi

January 20, 2025

Abstract

This report explores the theoretical and practical aspects of L_2Boosting, a boosting method tailored for high-dimensional linear models. Building upon the foundational work by Bühlmann (2006), which established consistency under exponential growth of predictors, we analyze its algorithmic formulation and discuss strategies for controlling model complexity through stopping criteria based on information theory. The study also examines its applicability to classification tasks and considers the integration of dimensionality reduction techniques.

1 L_2Boosting with componentwise linear least squares

In the reference paper[3] the author introduces the L_2Boosting technique as a powerful algorithm for doing regression and classification on high dimensional datasets. Boosting methods were initially developed as techniques for combining multiple prediction models by averaging predictions obtained from reweighted datasets. Subsequently, Breiman [1], [2] observed that the AdaBoost algorithm could be interpreted as a gradient descent optimisation process and Friedman [5] extended boosting to regression problems by formulating it as an optimization procedure based on the squared error loss function, which is referred to as L_2Boosting. To explain boosting for linear models, we first start by considering the usual regression model

$$Y_i = \sum_{j=1}^{p} \beta_j X_{ji} + \epsilon_i \tag{1}$$

where p is the number the predictor variables and ϵ is a random, mean-zero error term. Given some input data $\{(X_i, U_i);\ i = 1, ..., n\}$, where $U_1, ..., U_n$ denote some response variables which are not necessarily the original $Y_1, ..., Y_n$, the procedure yields an estimated function. Here, we will exclusively consider the componentwise linear least squares base procedure

$$\hat{\beta}_j = \frac{\sum_{i=1}^{n} U_i X_{ij}}{\sum_{i=1}^{n} (X_{ij})^2} \qquad j = 1, \ldots, p \tag{2}$$

$$\hat{g}_{(\mathbf{X},\mathbf{U})}(x) = \hat{\beta}_{\hat{S}} x^{(\hat{S})}, \quad \text{where,} \quad \hat{S} = \arg\min_{1 \le j \le p} \sum_{i=1}^{n} (U_i - \hat{\beta}_j X_{ij})^2 \tag{3}$$

2 L_2Boosting Algorithm Overview

The L_2Boosting algorithm is an iterative implementation of the above linear regression with the objective of optimizing the estimator procedure based on the squared error loss function.

The algorithm is initialized given data $\{(X_i, Y_i); i = 1, \ldots, n\}$ and applying the base procedure it yields to the function estimate

$$\hat{F}^{(1)} = \hat{g}^{(1)}_{(\mathbf{X},\mathbf{U})} \tag{4}$$

where $\hat{g} = \hat{g}_{(\mathbf{X},\mathbf{U})}$ is estimated from the original data. We will refer to it as step $m = 1$. Subsequently, the real-valued base procedure is fitted to the current residuals of the m-th step $U_i = Y_i - \hat{F}^{(m)}(X_i)$. The fit is denoted by $\hat{g}^{(m+1)}$ which is an estimate based on the original predictor variables and the current residuals. At each step the predictor is then updated as follows

$$\hat{F}^{(m+1)} = \hat{F}^{(m)} + \hat{g}^{(m+1)}. \tag{5}$$

This process is repeated until the stopping iteration $\hat{M}$ is reached. $\hat{F}^{(M)}$ is an estimator of the regression function $\mathbb{E}[Y|X]$. L_2Boosting [3] is therefore a technique based on repeated least squares fitting of residuals. However, it is often better to use small step sizes on performing estimation corrections, as a consequence the previous steps can be redefined as:

$$\hat{F}^{(1)} = \nu\hat{g} \tag{6}$$

$$\hat{F}^{(m+1)} = \hat{F}^{(m)} + \nu\hat{g}^{(m+1)}, \qquad m \geq 1, \quad 0 \leq \nu \leq 1 \tag{7}$$

where ν is a small constant during boosting iterations. It can be seen as a shrinkage parameter, describing the step size when updating $\hat{F}^{(m+1)}$ along the function $\hat{g}^{(m+1)}$. Small step sizes make the boosting algorithm slower and require a larger number $\hat{M}$ of iterations. However, the computational slow-down often turns out to be advantageous for better out-of-sample empirical prediction performance.

2.1 Stopping the boosting iterations

We introduce a definition of degrees of freedom for L_2Boosting and propose its application in the corrected Akaike information criterion (AIC_c), which is an estimator of prediction error and quality of statistical models for a given set of data. Boosting algorithms need to be run an optimal number of iterations, to avoid overfitting, and our objective is therefore to do so. We begin assigning degrees of freedom for boosting algorithms. We denote

$$\mathcal{H}^{(j)} = \frac{X^{(j)}(X^{(j)})^T}{||X^{(j)}||^2} \tag{8}$$

as an $n \times n$ matrix for the linear least squares fitting operator using the j-th predictor vector $X^{(j)} = (X_1^{(j)}, \ldots, X_n^{(j)})^T$ only, and denoting $||x||^2 = x^T x$ as the Euclidean norm for a vector $x \in \mathbb{R}^n$. It can be shown that the L_2Boosting hat-matrix, using the step size $0 < \nu \leq 1$, equals to

$$\mathcal{B}_m = I - (I - \nu\mathcal{H}^{(\hat{S}_m)})(I - \nu\mathcal{H}^{(\hat{S}_{m-1})}) \ldots (I - \nu\mathcal{H}^{(\hat{S}_1)}), \tag{9}$$

where $\hat{S} \in \{1, \ldots, p\}$ denotes the component which is selected in the least squares base procedure for the i-th boosting iteration. Using the trace of $\mathcal{B}_m$ as degrees of freedom, we employ a corrected version of AIC to define a stopping rule for boosting [6]:

$$AIC_c(m) = \log(\hat{\sigma}^2) + \frac{1 + Tr(\mathcal{B}_m)/n}{1 - (Tr(\mathcal{B}_m) + 2)/n} \tag{10}$$

$$\text{Where} \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - (\mathcal{B}_m Y)_i)^2, \qquad Y = (Y_1, \ldots, Y_n)^T \tag{11}$$

A reasonable estimate for the number of boosting iterations is then

$$\hat{M} = \arg \min_{1<m\leq m_{upp}} \Big[AIC_c(m)\Big] \tag{12}$$

where m_{upp} is a large upper bound for the candidate number of boosting iterations. The computationally efficient AIC_c criterion can then be used in the context where the base procedure has linear components. An alternative method for stopping boosting iteration we will test our model on is the Bayesian information criterion (BIC) [8], formally defined as

$$BIC(m) = \log(\hat{\sigma}^2) + \frac{Tr(\mathcal{B}_m)}{n} \cdot \log(n). \tag{13}$$

Clearly, to select the optimal number of boosting iterations, $\hat{M}$, based on BIC, we minimise $BIC(m)$ over the range of iterations:

$$\hat{M} = \arg \min_{1<m\leq m_{upp}} \Big[BIC(m)\Big] \tag{14}$$

with m_{upp} defined as above. BIC is based on Bayesian principles, incorporating a prior on model parameters and assuming the true model is among the candidates, unlike AIC, which seeks the model that minimizes predictive error without assuming a true model exists. The BIC formula $BIC = k\ln(n) - 2\ln(\hat{L})$ arises from approximating the marginal likelihood under a uniform prior. In boosting, k is replaced by $Tr(\mathcal{B}_m)$, the effective degrees of freedom, leading to BIC being proportional to $\log(n)$, hence penalising complexity more heavily as the sample size grows.

2.2 Consistency of L_2Boosting in high dimensions

In this section we comment on the theoretical result provided by the paper to justify the performance of L_2Boosting in high-dimensional scenarios. By high-dimensional scenarios, we mean that the number of predictors is significantly higher than the number of samples. In particular, the paper provides a convergence result that allows the number of predictors to grow exponentially with the number of samples n. We consider the model

$$Y_i = f_n(X_i) + \varepsilon_i, \quad i = 1, ..., n, \qquad f_n(x) = \sum_{j=1}^{p_n} \beta_{j,n} x^{(j)}, \quad x \in \mathbb{R}^{p_n} \tag{15}$$

where $X_1, ..., X_n$ are i.i.d with $\mathbb{E}[|X^{(j)}|^2] = 1$ for all $j = 1, ..., p_n$ and $\varepsilon_1, ..., \varepsilon_n$ are i.i.d, independent from $\{X_s; 1 \leq s \leq n\}$, with $\mathbb{E}[\varepsilon] = 0$. The condition $\mathbb{E}[|X^{(j)}|^2] = 1$ is not necessary for L_2Boosting but it allows to better interpret the magnitude of the coefficients $\beta_{j,n}$, as all data is at the same scale. As we already mentioned, the number of predictors p_n is allowed to grow with the sample size n. This implies that the predictors $X_i = X_{i,n}$ and the response $Y_i = Y_{i,n}$ will also depend on n. More specifically, we allow an exponential dependence of the form

$$p_n = \mathcal{O}(\exp(Cn^{1-\xi})), \quad n \to \infty, \quad \text{for some } \; 0 < \xi < 1, \quad 0 < C < \infty \tag{16}$$

This assumption allows for a very large predictor dimension relative to the sample size n, which are the situations we are interested in. The next assumption accounts for sparseness of the data, in the sense that despite n (and consequently p_n) increasing arbitrarily, the number of relevant predictors is going to stay finite

$$\sup_{n\in\mathbb{N}} \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty \tag{17}$$

which holds in particular for regressions where the number of $\beta_{j,n} \neq 0$ is independent of n and finite. Finally, some other assumptions on the predictors and the errors include

$$\sup_{1 \leq j \leq p_n, n \in \mathbb{N}} ||X^{(j)}||_\infty < \infty, \quad \mathbb{E}[|\varepsilon|^s] < \infty \text{ for some } s > 4/\xi \tag{18}$$

which essentially ensure that the values of the predictors and the errors don't become too big as the sample size $n \to \infty$. These four assumptions together with the L_2Boosting algorithm described in previous sections guarantee the following result

Theorem 1. *Consider the L_2Boosting model satisfying all previous four assumptions. Then, the boosting estimate $\hat{F}^{(m)}() = \hat{F}_n^{(m)}()$ with the component-wise linear base procedure satisfies: for some sequence $(m_n)_{n \in \mathbb{N}}$ with $m_n \to \infty$ ($m_n \to \infty$) sufficiently slowly,*

$$\mathbb{E}[|\hat{F}_n^{(m_n)}(X) - f_n(X)|^2] = o_P(1), \quad n \to \infty \tag{19}$$

where X denotes a new predictor variable, independent of and with the same distribution as the X-component of the data $(X_i, Y_i), i = 1, ..., n$.

This implies L_2Boosting is a good estimator for a sparse regression function when the number of predictors p is allowed to increase exponentially with the sample size n. The paper also points out that no assumptions are made on the correlation structure of the predictors.
Theorem 1 holds also in the case of heteroscedastic errors ε_i, that is, when $\varepsilon_i = \varepsilon_i(X_i)$, by assuming $(X_1, Y_1), ..., (X_n, Y_n)$ i.i.d. and suitable conditions for Y_i.
We now focus on the case of binary classification with $Y_i \in \{0, 1\}$, which is the case of the dataset we have worked with for this project. Bühlmann [4] argues why L_2Boosting is a reasonable procedure for binary classification. We have a similar model

$$Y_i = f_n(X_i) + \varepsilon_i, \qquad \varepsilon_i = Y_i - f_n(X_i) \tag{20}$$

where the $\varepsilon_1, ...\varepsilon_n$ are independent but heteroscedastic (dependent on X) with $\mathbb{E}[\varepsilon_i] = 0$ and $Var(\varepsilon_i) = f_n(X_i)(1 - f_n(X_i))$. In the case of binary classification

$$\mathbb{E}[Y|X = x] = \mathbb{P}[Y = 1|X = x] \tag{21}$$

since Y can only take values 0 and 1. Therefore, L_2Boosting gives an estimate of the conditional probability $\mathbb{P}[Y = 1|X = x]$. By setting the threshold to 1/2, the L_2Boosting classifier is given by $\hat{C}_n^{(m)}(x) = \mathbb{I}_{[\hat{F}_n^{(m)}(x) > 1/2]}$

Corollary 2. *Consider a binary classification problem with $(X_1, Y_1), ..., (X_n, Y_n)$ independent and $Y_i \in \{0, 1\}$ for all $i = 1, ..., n$. Assume that $f_n(x) = \mathbb{P}_n[Y = 1|X = x] = \sum_{j=1}^{p_n} \beta_{j,n} x^{(j)}$ and the four assumptions hold. Then, for the L_2Boosting estimate: for some sequence $(m_n)_{n \in \mathbb{N}}$ with $m_n \to \infty$ ($n \to \infty$) sufficiently slowly,*

$$\mathbb{E}_X|\hat{F}_n^{(m_n)}(X) - f_n(X)|^2 = o_P(1), \quad n \to \infty \tag{22}$$

$$\mathbb{P}_{X,Y}[\hat{C}_n^{(m_n)}(X) \neq Y] - L_{n,Bayes} = o_P(1) \tag{23}$$

where $L_{n,Bayes}$ denotes the Bayes risk $E_X[min\{f_n(X), 1 - f_n(X)\}]$ and X, Y denote new response and predictor variables, independent of and with the same distribution as the data $(X_i, Y_i), i = 1, ..., n$.

3 Numerical Implementation and Dataset

We began by verifying the numerical simulation results presented in the reference paper [3], both for uncorrelated predictors and block-correlated predictors, which will be discussed in Section 4. Subsequently, we implemented the L_2Boosting algorithm for classification on a high-dimensional dataset. Such datasets are characterized by a large number of features and are typically uncommon to find in the financial industry while are more prevalent in the medical sector. We therefore opted to use a dataset that describes a cell line derived from a human breast cancer patient [7], where the column indices represent 22,350 gene expressions for each cell, and the rows correspond to 233 available cells. The dataset has a reference column for the condition of growth of each cell with respect to oxygen levels, hypoxic or normoxic. It is also sparse, with around 45% of its values being equal to zero. To this end, we implemented a classification algorithm based on the L_2Boosting techniques discussed in the previous chapters, with particular emphasis on regulating the number of iterations using the AIC and BIC criterion as stopping time conditions. Following the recommendations from the paper [3], we have used the following formula for AIC:

$$\text{AIC}(m) = -2 \cdot \log(\mathcal{L}) + 2 \cdot Tr(\mathcal{B}_m) \tag{24}$$

and the standard formula for BIC:

$$\text{BIC}(m) = -2 \cdot \log(\mathcal{L}) + \log(n) \cdot Tr(\mathcal{B}_m) \tag{25}$$

with the Bernoulli log-likelihood

$$\log(\mathcal{L}) = \sum_{i=1}^{n} \left[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \right]. \tag{26}$$

We will therefore compare the results stopping the problem under both criterions. In terms of model hyperparameters, several values of ν will be tested. We expect that higher shrinking factors produce more instability during training as well as overfitting, whereas lower convergence speed is produced by lower shrinking factors. We validate the model we developed by comparing the Mean Squared Errors (MSE) to the one presented in the reference paper [3] (Section 4).

Method	**p = 3**	**p = 10**	**p = 100**
uncorrelated predictors			
Developed model	2.181	3.188	9.285
Reference model	1.658	2.318	8.792
block-correlated predictors			
Developed Model	1.304	2.569	4.948
Reference Model	1.054	1.649	4.643

Table 1: Mean Squared Errors.

We observe that the results we obtained are in line with what the reference paper presented, hence validating our developed model.

4 Empirical Classification Performance

In this section we conduct a performance comparison of the L_2Boosting on the previously described high-dimensional dataset with other machine learning algorithms, namely the Logistic Regression, AdaBoost, and XGBoost. Results are illustrated in Figure 1.

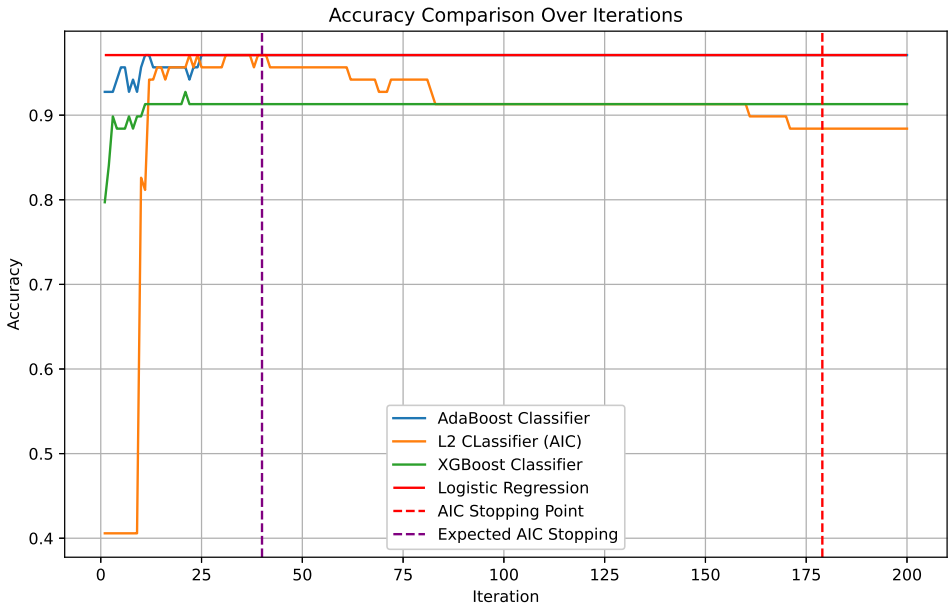


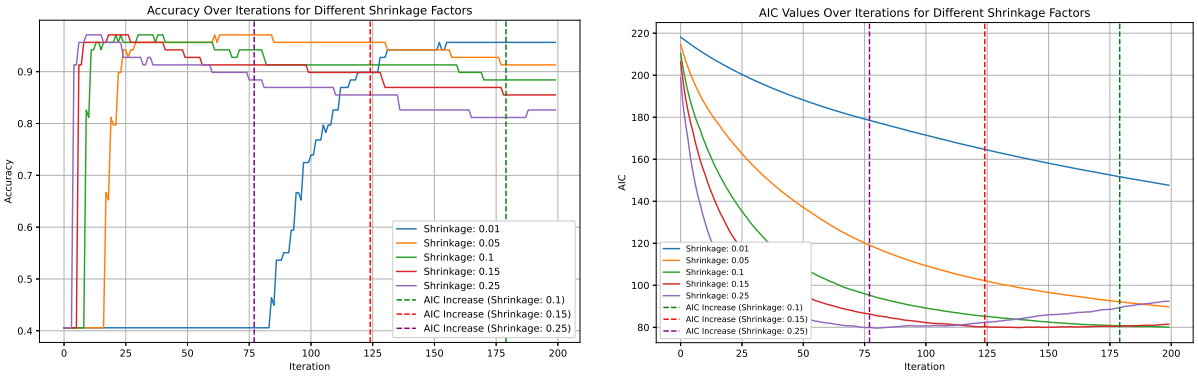
Figure 1: Level of accuracy achieved by different models.

As it is possible to evince from Figure 1, the L_2Boosting classifier performs competitively well compared to the other learning methods up to 160 iterations. The accuracy of the model then starts to decrease until it reaches a level of 88% at the AIC stopping time condition depicted in Figure 1. Although Logistic Regression is not a boosting algorithm, it serves as a meaningful benchmark for assessing classification performance. On the high-dimensional dataset used, it achieves an accuracy of approximately 97%. As it is possible to notice from the plot, the accuracy of the L_2Boosting model reaches its peak around the 40th iteration, after which the model's accuracy begins to decrease.

Analyzing the results, the AIC is theoretically expected to stop the learning process at the optimal training condition to prevent overfitting. However, we believe that this behavior is not reflected in practice. Specifically, as it is possible to notice from the plot, the model performance on the test set begins to decrease after the 40th iteration, while its complexity continues to increase as well, which are both symptoms of a possible overfitting condition. As a result, we can see that the predictive quality of the L_2 Boosting algorithm begins to decline on the high-dimensional dataset. Despite these observations and our diligent efforts to address the issue, we chose to retain the results and interpret them to the best of our ability. We therefore acknowledge that the theoretical stopping point should ideally align more closely with the purple dashed line shown in Figure 1.

Another note to make is that the BIC stopping criterion stops the model at the same point as the AIC, and therefore the results are just presented for AIC for simplicity. Focusing specifically on the performance of the L_2Boosting classifier, we conducted a parameter space analysis by evaluating the model across various shrinkage factors, $\nu \in \{0.01, 0.05, 0.1, 0.15, 0.25\}$. In Figure 2a, we observe that larger shrinkage values result in greater instability, as it was expected. For instance, $\nu = 0.25$, given a higher frequency of oscillations in the algorithm, the accuracy begins to decrease at an earlier stage with respect to the others. Conversely, smaller shrinkage values exhibit slower convergence, as is evident for $\nu = 0.01$, where the accuracy remains very low until approximately the 80th iteration. Based on the guidelines from the reference paper and our

comparative analysis, we selected $\nu = 0.1$ as the most suitable shrinking value for this dataset. In addition, it is also possible to see that in figure 2a the stopping times AIC are not drawn for $\nu = 0.05$ and $\nu = 0.01$ since they occur much later than the others, as expected. On the other hand, in Figure 2b are depicted the AIC curves for different shrinkage factors. As expected, we observe a shift towards a higher number of iterations for the minimum of the AIC curves.



(a) Accuracy (b) AIC

Figure 2: The two plots illustrate distinct aspects of the L_2Boosting algorithm's performance. The plot on the left depicts the change in accuracy for different shrinkage factors as the number of iterations increases. The plot on the right instead, shows the behavior of the AIC criterion for various shrinkage factors over the same range of iterations.

Regarding the AIC stopping condition, recalling equations 9 and 24, we can see that the stopping iteration based on the AIC is linearly dependent on the trace of $\mathcal{B}_m$. However, it itself depends on increasing powers of ν. For a reasonably small shrinking value, the trace of $\mathcal{B}_m$ will increase linearly with ν, as higher-order terms will have progressively less influence on the value of $Tr(\mathcal{B}_m)$. Examining the first iterations, we observe that

$$\begin{aligned}
\mathcal{B}_m^0 &= \mathcal{I} \\
\mathcal{B}_m^1 &= \mathcal{I} - (\mathcal{I} - \nu\mathcal{H}^1) = \nu\mathcal{H}^1 \\
\mathcal{B}_m^2 &= \mathcal{I} - (\mathcal{I} - \nu\mathcal{H}^1)(\mathcal{I} - \nu\mathcal{H}^2) = \nu(\mathcal{H}^1 + \mathcal{H}^2) - \nu^2\mathcal{H}^1\mathcal{H}^2 \\
&\vdots
\end{aligned}$$

The behavior of $Tr(\mathcal{B}_m)$ is approximately linear with respect to the number of iterations, with minor corrections on the order of $\mathcal{O}(\nu^2)$ notable on the purple line (due to the fact that we are choosing $\nu < 1$). Further details about this phenomenon can be found in Appendix A for the interested reader.
Going back to the L_2Boosting algorithm we already noticed that its performance applied to the dataset varies depending on the stopping time of the process, which is influenced by the choice of the shrinkage factor ν. As it is possible to see from Figure 3, for a reasonable shrinkage factor of $\nu = 0.1$, the model selects 33 relevant features out of 22,350.

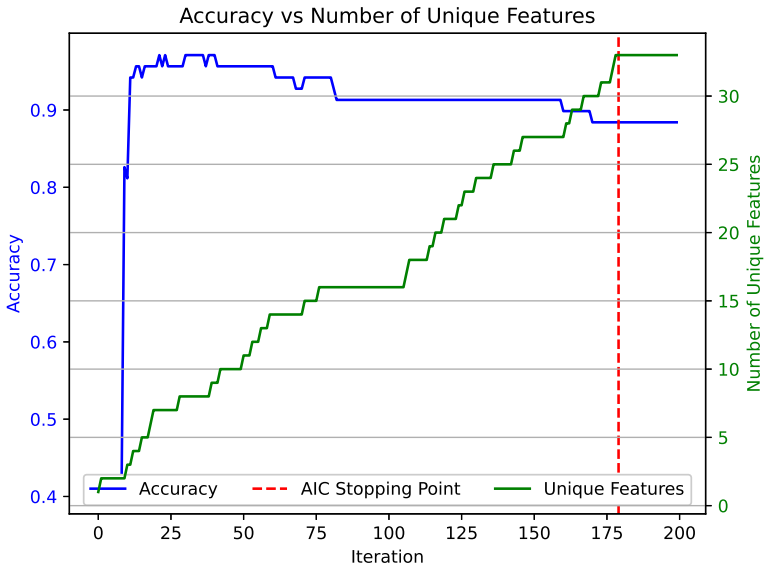


Figure 3: Accuracy vs Number of Unique Features

This phenomenom is observed because the model's accuracy becomes sufficiently high with a selection set of 33 features, corresponding to the genes most strongly associated with breast cancer across the samples. It is also possible to notice from the plot that the model experiences two "plateau" conditions in the new feature selection. The first occurs immediately after the AIC stopping criterion, which is reasonable, as the model reaches its maximum prediction accuracy by selecting the 33 most relevant features to effectively predict breast cancer. The second plateau instead, is observed between iterations 75 and 100, when the number of selected features is reduced to 16. Given these facts, we can infer that, out of the 33 features selected, 16 are likely more relevant than the others. This explains why the model temporarily stops selecting additional new features, as its accuracy is already high enough with the current set. This observation further reinforces the belief that the model should have probably been stopped earlier than iteration 175.

5 Effect of Dimensionality Reduction

Given that we are dealing with high-dimensional data, an approach that is worth exploring is that of dimensionality reduction. We first standardize the data and afterwards employ linear and nonlinear dimensionality reduction through Principal Component Analysis (PCA) and autoencoder. PCA aims to identify the most significant directions (principal components) that capture the maximum variance in the data. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n) \in \mathcal{M}_{p,n}$ represent our data matrix, where each $\mathbf{X}_k \in \mathbb{R}^p$ is a p-dimensional data point. The sample mean is denoted by

$$\boldsymbol{\mu} := \frac{1}{n} \sum_{k=1}^{n} \mathbf{X}_k,$$

and the variance-covariance matrix is Σ. PCA seeks to find an orthonormal basis $\mathbf{E} = (\mathbf{E}_1, \ldots, \mathbf{E}_d) \in \mathcal{M}_{p,d}$ that minimizes the reconstruction error defined by

$$\mathcal{D}(\boldsymbol{\mu}, \mathbf{B}, \mathbf{E}) := \sum_{k=1}^{n} \|\mathbf{X}_k - (\boldsymbol{\mu} + \mathbf{E}\mathbf{B}_k)\|_2^2,$$

where $\mathbf{B} \in \mathcal{M}_{n,d}$ contains the coefficients $\mathbf{B}_k \in \mathbb{R}^d$ for each data point. The optimization problem can be formulated as

$$\min_{\mathbf{E},\mathbf{B}:\|\mathbf{E}\|=1} \mathcal{D}(\boldsymbol{\mu}, \mathbf{B}, \mathbf{E}).$$

Solving this, involves projecting the centered data onto the principal components $\mathbf{E}$, which are the eigenvectors of the covariance matrix Σ corresponding to the largest eigenvalues. This projection captures the maximum possible variance with a reduced number of dimensions.
We empirically prove that only 125 components are needed to explain 95% of the variance. We then project our data into lower-dimensional space, fit an L_2 Boosting classifier, and observe that the accuracy is 0.7826. On the other hand, autoencoders also provide a nonlinear approach to dimensionality reduction by learning efficient data representations through neural networks. An autoencoder is a function $f : \mathbb{R}^p \to \mathbb{R}^p$ defined as:

$$f = g_{decode} \circ g_{encode} \tag{27}$$

$$g_{encode} : \mathbb{R}^p \to \mathbb{R}^d \qquad d \ll p \tag{28}$$

$$g_{decode} : \mathbb{R}^d \to \mathbb{R}^p \tag{29}$$

which is trained to minimise the loss function:

$$\mathcal{L}(f) = \sum_{i=1}^{n} ||\mathbf{X}_i - f(\mathbf{X}_i)||^2 \tag{30}$$

We choose the architecture to be of the form

$$f \in \mathcal{N}_4(I, 256, 128, 128, 256, I; ReLU, ReLU, ReLU, ReLU, ReLU, \mathbb{I}_I) \tag{31}$$

and we then use g_{encode} to project our dataset to a lower dimensional space. We then compare the performance of the L_2Boosting classifier on the reduced dataset to the full one. We obtain an accuracy of 0.6956. We observe that both methods, compared to the classifier trained on the whole dataset which obtained an accuracy of 0.9710, these methods obtain lower accuracy. There are multiple reasons why this may happen:

1. While PCA projects the data onto a subspace that maximises variance, these principal components may not capture the features most crucial for class discrimination.

2. Autoencoders aim to reconstruct the input data accurately, which means the identified subspace $\mathbb{R}^d$ might not preserve the discriminative features essential for binary classification. Additionally, with a limited sample size (227), the model likely overfitted, as indicated by the non decreasing validation loss in Figure 6, preventing proper convergence.

Figures detailing these results can be found in Appendix B for the curious reader.

6 Conclusion

This study investigated the implementation and performance of the L_2Boosting algorithm in high-dimensional regression and classification tasks, with a focus on sparse datasets. The results confirmed its theoretical strengths, particularly in feature selection and predictive accuracy, aligning well with the performance of more complex models such as AdaBoost and XGBoost. However, several challenges emerged, particularly regarding the practical application of stopping criteria and dimensionality reduction techniques.

The L_2Boosting algorithm demonstrated strong feature selection capabilities, efficiently identifying a minimal subset of predictors critical for classification. In the context of the breast cancer dataset, the algorithm selected around 33 features, reflecting its capacity to extract meaningful information even in high-dimensional and sparse data. Nonetheless, the analysis revealed inconsistencies between the theoretical stopping points determined by AIC and the observed behavior. In practice, AIC often stopped the algorithm earlier than necessary, potentially leading to overfitting. This underscores the need for more adaptive stopping criteria that better balance model complexity and predictive performance.
The role of the shrinkage factor (ν) was also crucial, influencing convergence speed, stability, and accuracy. Larger values of ν resulted in faster convergence but increased instability and overfitting, while smaller values provided smoother convergence at the cost of computational efficiency. This highlights the importance of careful calibration of ν to suit the dataset.
Exploration of dimensionality reduction techniques, including PCA and autoencoders, revealed additional insights. PCA effectively captured variance but often failed to retain discriminative features necessary for classification. Autoencoders, while capable of learning nonlinear representations, were hindered by the limited sample size, leading to overfitting and suboptimal performance. These findings suggest that dimensionality reduction techniques tailored to preserve class-specific information are essential for high-dimensional, sparse datasets.
Overall, L_2Boosting shows significant promise for high-dimensional classification, offering efficient feature selection and competitive accuracy. However, the study highlights areas for refinement, particularly in optimizing stopping criteria and integrating dimensionality reduction methods. Future work could focus on hybrid stopping strategies that combine information criteria with validation metrics or explore nonlinear extensions of L_2Boosting to enhance its applicability across diverse datasets and tasks.

References

[1] 1998. "Arcing classifiers (with discussion)". In: *The Annals of Statistics* 26.3 (2006). ISSN: 0090-5364. URL: https://www.jstor.org/stable/120055.

[2] L. Breiman. "Prediction Games and Arcing Algorithms". In: *Neural Computation* 11 (1999), pp. 1493–1517. URL: https://api.semanticscholar.org/CorpusID:14488820.

[3] Peter Bühlmann. "Boosting for high-dimensional linear models". In: *The Annals of Statistics* 34.2 (Apr. 2006). ISSN: 0090-5364. DOI: 10.1214/009053606000000092. URL: http://dx.doi.org/10.1214/009053606000000092.

[4] Peter Bühlmann and Bin Yu. "Boosting With the L2 Loss". In: *Journal of the American Statistical Association* 98.462 (2003), pp. 324–339. DOI: 10.1198/016214503000125. eprint: https://doi.org/10.1198/016214503000125. URL: https://doi.org/10.1198/016214503000125.

[5] Jerome Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29 (Nov. 2000). DOI: 10.1214/aos/1013203451.

[6] Clifford M. Hurvich, Jeffrey S. Simonoff, and Chih-Ling Tsai. "Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 60.2 (1998), pp. 271–293. ISSN: 1369-7412,1467-9868. DOI: 10.1111/1467-9868.00125. URL: https://doi.org/10.1111/1467-9868.00125.

[7] Andrea A. Perreault, Danielle M. Sprunger, and Bryan J. Venters. "Epigenetic and transcriptional profiling of triple negative breast cancer. (The dataset discussed in this article is not publicly available)". In: *Scientific Data* 6.1 (2019), p. 190033. DOI: 10.1038/sdata.2019.33.

[8] Gideon Schwarz. "Estimating the dimension of a model". In: *Ann. Statist.* 6.2 (1978), pp. 461–464. ISSN: 0090-5364,2168-8966. URL: http://links.jstor.org/sici?sici=0090-5364(197803)6:2%3C461:ETDOAM%3E2.0.CO;2-5&origin=MSN.

A L_2Boosting relevant plots

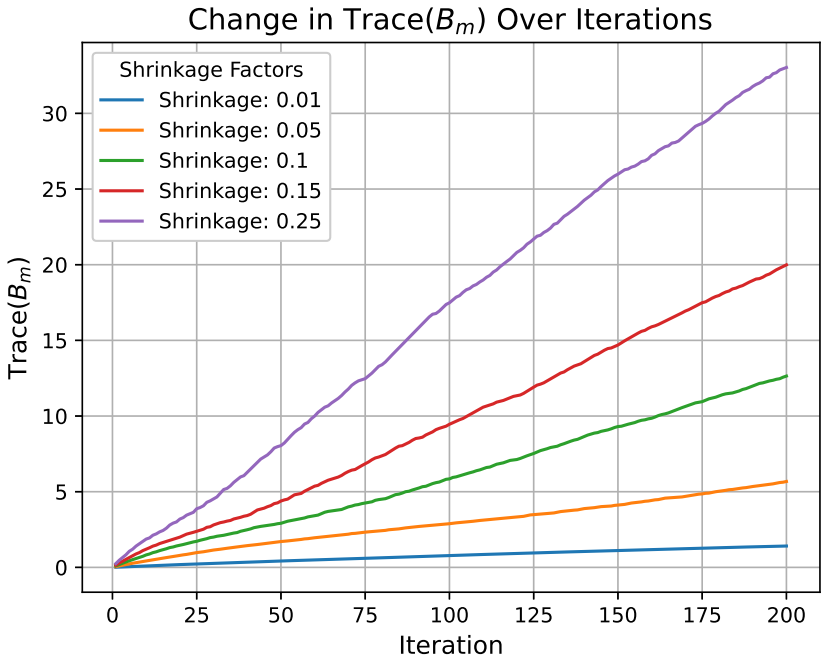


Figure 4: $Tr(\mathcal{B}_m)$ trends for different shrinkage factors in function of the number of iterations.

B Dimensionality reduction relevant plots

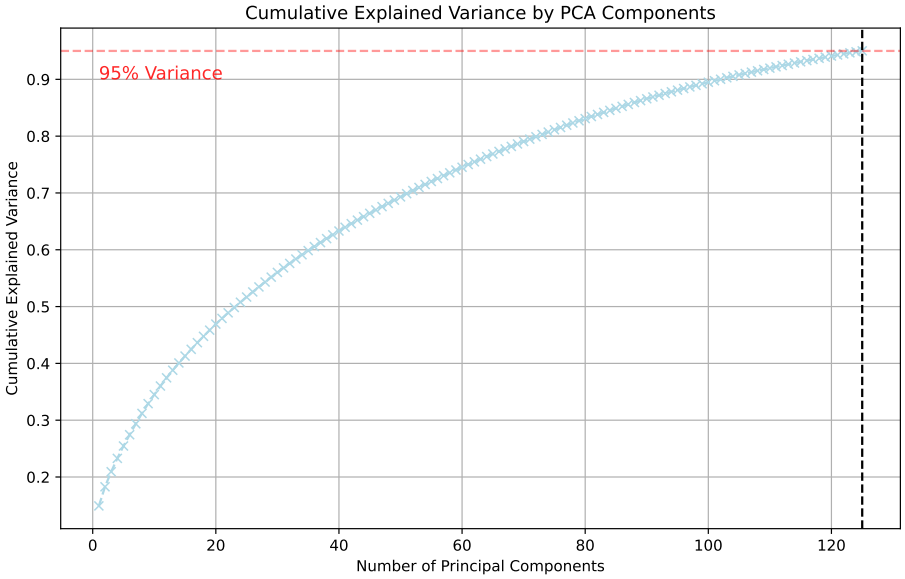


Figure 5: Explained variance versus Number of Principal components

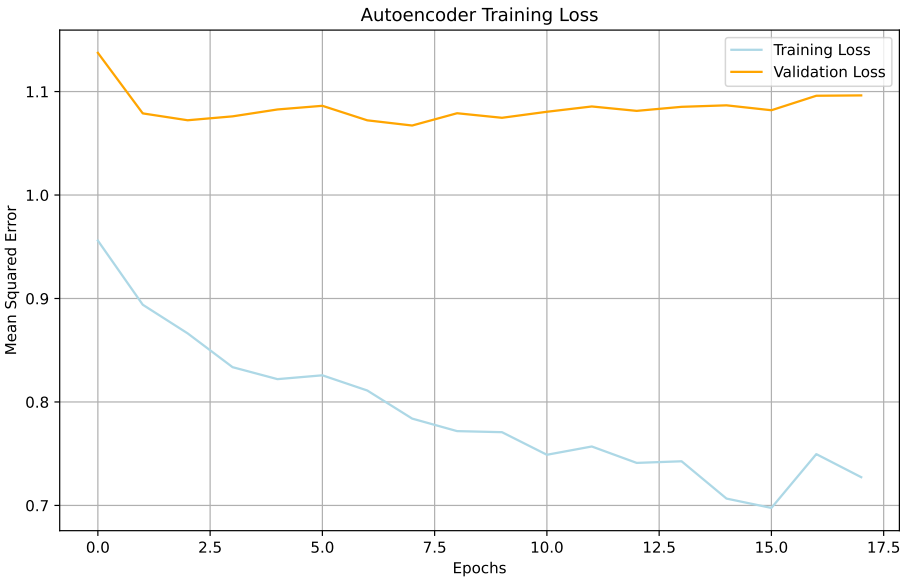


Figure 6: Training and validation loss