APL Machine Learning

RESEARCH ARTICLE | DECEMBER 11 2023

Quantifying the impact of uninformative features on the performance of supervised classification and dimensionality reduction algorithms

Weihua Lei ; Cleber Zanchettin ; Zoey E. Ho ; Luís A. Nunes Amaral

APL Mach. Learn. 1, 046118 (2023)
https://doi.org/10.1063/5.0170229

Quantifying the impact of uninformative features on the performance of supervised classification and dimensionality reduction algorithms

Cite as: APL Mach. Learn. 1, 046118 (2023); doi: 10.1063/5.0170229
Submitted: 1 August 2023 • Accepted: 12 November 2023 •
Published Online: 11 December 2023

Weihua Lei,[1] Cleber Zanchettin,[1,2] Zoey E. Ho,[3] and Luís A. Nunes Amaral[1,4,5,a)]

AFFILIATIONS

[1] Department of Physics and Astronomy, Northwestern University, Evanston, Illinois 60208, USA
[2] Centro de Informática, Universidade Federal de Pernambuco, Recife, Pernambuco 52061080, Brazil
[3] Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, Illinois 60208, USA
[4] Department of Chemical and Biological Engineering, Northwestern University, Evanston, Illinois 60208, USA
[5] Northwestern Institute on Complex Systems (NICO), Northwestern University, Evanston, Illinois 60208, USA

[a)] **Author to whom correspondence should be addressed:** amaral@northwestern.edu

ABSTRACT

Machine learning approaches have become critical tools in data mining and knowledge discovery, especially when attempting to uncover relationships in high-dimensional data. However, researchers have noticed that a large fraction of features in high-dimensional datasets are commonly uninformative (too noisy or irrelevant). Because optimal feature selection is an NP-hard task, it is essential to understand how uninformative features impact the performance of machine learning algorithms. Here, we conduct systematic experiments on algorithms from a wide range of taxonomy families using synthetic datasets with different numbers of uninformative features and different numbers of patterns to be learned. Upon visual inspection, we classify these algorithms into four groups with varying robustness against uninformative features. For the algorithms in three of the groups, we find that when the number of uninformative features exceeds the number of data instances per pattern to be learned, the algorithms fail to learn the patterns. Finally, we investigate whether increasing the distinguishability of patterns or adding training instances can mitigate the effect of uninformative features. Surprisingly, we find that uninformative features still cause algorithms to suffer big losses in performance, even when patterns should be easily distinguishable. Analyses of real-world data show that our conclusions hold beyond the synthetic datasets we study systematically.

INTRODUCTION

Learning algorithms have become increasingly popular in data mining and knowledge discovery in various domains, such as predicting COVID-19 sequelae,[1] discovering new drugs,[2] designing new materials,[3] understanding human mobility,[4,5] and predicting customer churn rate.[6] The performance of learning algorithms is determined by two main factors: the quality of the training data and the inductive bias of the algorithm.[7] The inductive bias (or learning bias) allows learning of general characteristics of the task based on specific training data instances. It refers to the generalizations of the algorithm for similar consistency with the observed data, compelling the model's quality to be related to the data quality. However, from a practical perspective, the performance of the algorithm depends on not only the quality but also the appropriateness of data.[8]

Real-world data are an admixture of informative (signal-rich) and uninformative (too noisy or irrelevant) features. Disappointingly, what contributes to the lack of information for a given feature in the dataset is nearly always unknown. Critically, the learning algorithms can become overly sensitive to noise in the input data, resulting in over-fitting.

In data mining applications, two types of noise are generally considered: class noise and feature noise.[7,9–11] Class noise is

primarily studied by mislabeling data to investigate the performance limits of the algorithm under weakly supervised or unsupervised settings. Feature noise is typically studied by infusing noise into existing features.

An aspect of noise that has not been studied systematically is the presence of uninformative features—those that are not relevant to the learning task—and their impacts on the algorithm's performance. The increasing availability of data means that for learning tasks, such as classification, clustering, or regression, keeping potentially uninformative features is less onerous than filtering them out.[12,13] For example, when attempting to predict disease progression[14] or to extract cell type from the expression levels of thousands of genes obtained through single-cell RNA sequencing,[15] it is far easier to include all features than to determine which features should be removed because they contain no information.[16]

Even though different techniques have been proposed to filter out less relevant and irrelevant features,[17,18] optimal selection of the subset of features is an NP-hard task.[19] As a result, large numbers of features, many of which may lack any information, are still widely used in tasks such as classification and clustering.[20]

Here, we test the hypothesis that for some learning algorithms, the training process can be significantly harmed by the presence of uninformative features. For example, learning algorithms that use nearest neighbors and rely on calculating distances between data instances in space could easily be affected by the "curse of dimensionality."[21] Uninformative features add extra dimensionality and noise while providing no additional separation for different clusters. In contrast, learning algorithms using decision trees have an embedded feature selection process that can potentially filter out irrelevant features and, therefore, be more robust to uninformative features. To our knowledge, these intuitions have never been systematically tested for supervised classification and dimensionality reduction algorithms.

Thus, we focus on addressing the gap in the understanding of the impact of uninformative features on knowledge mining through an experimental investigation using synthetic data *from the perspective of the user of machine learning algorithms*. Specifically, we analyze the interplay between the number of uninformative features, the number of patterns to be learned, the distinguishability of the patterns, and the sample size. We evaluate classification tasks with 2 to 9 Gaussian clusters varying from including no uninformative features to up to 8000 uninformative features and explore the limits of learning algorithms to maintain their proper function in the presence of uninformative features. We find that the performance of most algorithms is susceptible to uninformative features, which could usually influence model suitability and robustness. Finally, we test these insights using real-world data for single-cell phenotyping from cellular indexing of transcriptomes and epitopes by using the sequencing (CITE-seq) method.

RESULTS

We test the robustness of a suite of 15 popular classification algorithms—and an additional 11 algorithms in the supplementary material—in the scikit-learn (version "0.21.3") Python package.[22] These algorithms cover a broad range of machine learning (ML) approaches and are the most accessible to users of machine learning approaches (Table I). See Table S1 for hyper-parameter values.

TABLE I. Taxonomy of benchmarked algorithms.

Type	Algorithm
Discriminant analysis	Linear discriminant analysis
Ensemble methods	Bagging classifier Gradient boosting classifier Random forest classifier
Gaussian processes	Gaussian process classifier
Linear models	Logistic regression Logistic regression CV Perceptron Ridge classifier
Naive Bayes	GaussianNB
Nearest neighbors	KNeighbors classifier Nearest centroid
Neural network	MLP classifier
Support vector machine	SVC
Tree	Decision tree classifier

Synthetic data

In order to rigorously test the impact of uninformative features, we generate synthetic datasets with known properties. In these synthetic datasets, the patterns to be learned are modeled as two-dimensional Gaussian clusters with a covariance matrix,

$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{1}$$

where the separation between the centers of adjacent clusters equals d (normalized by the standard deviations $\sigma \equiv 1$). The uninformative features are modeled using uniform distributions $U(0,1)$ [Fig. 1(a)]. The optimal decision boundaries and theoretical accuracy can be calculated exactly in this case. As shown in Fig. 1, the data used for the analyses have four major parameters: the number of uninformative features in the dataset, the number of patterns (clusters) to be learned, the separation between adjacent clusters (d), and the number of training instances (sample size).

Relative loss of predictability

The robustness of a learning algorithm is in its ability to learn the desired pattern in the presence of noise. José *et al.* proposed the relative loss of accuracy (RLA) as a metric to measure the robustness of learning algorithms in the presence of class and feature noise.[23] RLA is easily interpretable and allows the comparison of performances of a single algorithm in the presence of different noise levels. However, RLA is not well suited for comparisons across

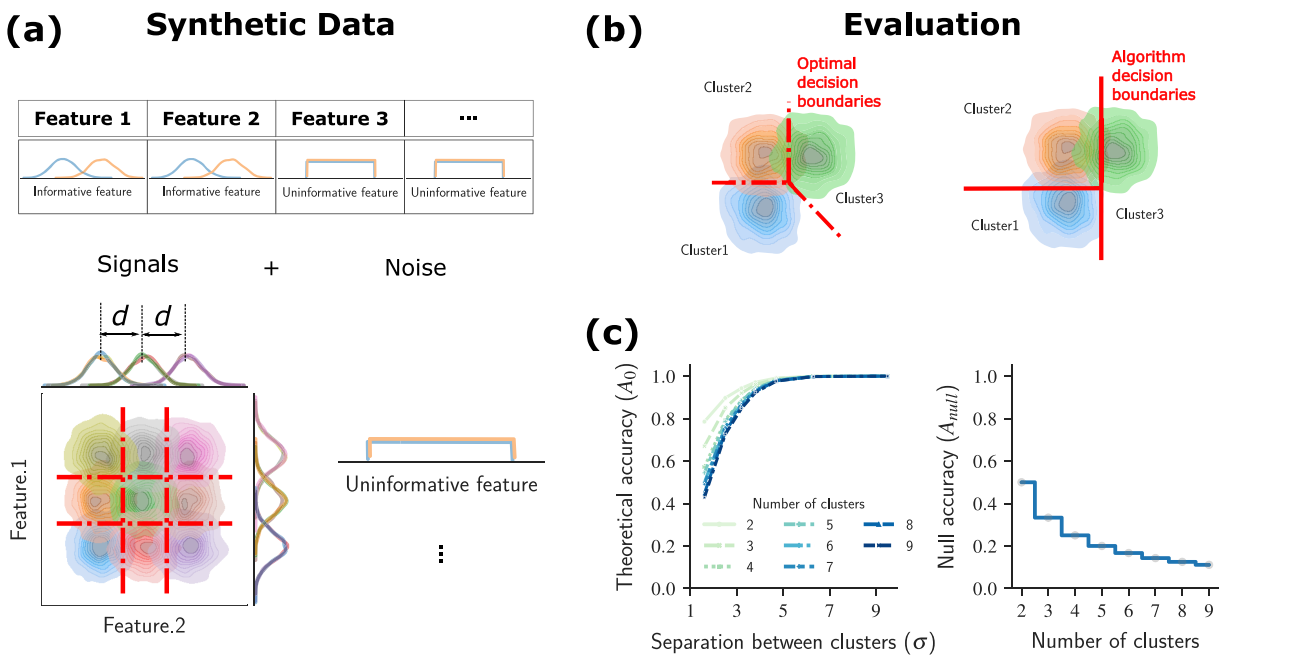

FIG. 1. Illustration of the experimental setup used to evaluate the sensitivity of machine learning (ML) classifiers to uninformative features. (a) We generate corpora of synthetic datasets in which two features contain information that enables us to identify which two-dimensional Gaussian cluster a data instance belongs to, while a specified number of uninformative features take values drawn from a uniform distribution over the entire support. (b) Comparison of decision lines obtained with a specific ML learning algorithm (red full lines) vs the optimal theoretical decision boundaries (red dotted-dashed lines). (c) To enable comparisons across datasets with different numbers of clusters and different theoretical accuracy expectations, we introduce a new metric in which we denote the relative loss of predictability; see Eq. (2). The relative loss of predictability normalizes the measured accuracy by the maximum theoretical accuracy A_0 and the null accuracy A_{null}, which is the baseline's accuracy where instances' labels are randomly assigned.

models or for data encoding a different number of patterns to be learned.

To address these caveats, we build on the RLA here. We note that more important than model accuracy in the absence of noise is the theoretical upper limit of accuracy, which we denote A_0. For example, with a minimum separation between clusters placed on a 2D grid of $d = 3.2\sigma$, the theoretical accuracy is 0.94 for two clusters and 0.85 for nine clusters. Additionally, we take account of task difficulty by computing the baseline accuracy A_{null} where each instance class is randomly assigned, which equals the inverse of the number of clusters for equally populated clusters. By introducing A_{null} into the denominator, we allow fair comparisons of learning tasks of different difficulties (e.g., different numbers of clusters to be detected). We define the relative loss of predictability as

$$L_k(m,p) = 100 \cdot \frac{A_0 - A_k(m,p)}{A_0 - A_{null}}, \tag{2}$$

where $A_k(m,p)$ is the accuracy of algorithm k trained on a dataset that has m uninformative features and p patterns to be detected. This metric can be extended to imbalanced datasets by replacing accuracy with balanced accuracy.

The impact of uninformative features

To evaluate the robustness of learning algorithms against the presence of uninformative features, we generated multiple datasets for each set of patterns. The number of clusters varies from 2 to 9, and the number of uninformative features varies from 0 to 8000. We assign 1000 data instances to each cluster and set the separation between neighboring clusters to $d = 3.2\sigma$, ensuring that the expected cluster overlapping is less than 0.1%. We thus generate datasets for $8 \times 8 = 64$ sets of parameters that can be used to benchmark the performance of 15 classification algorithms (listed in Table I) with fivefold cross-validation.

Inspired by our visual inspection, we define the following criteria to cluster algorithms based on their sensitivity to the number of uninformative features and to the number of patterns to be learned:

U: high sensitivity to the number of features: $L(100,2) - L(0,2) > 25\%$,

u: intermediate sensitivity to the number of features: $L(8000,2) - L(0,2) > 10\%$, and

P: high sensitivity to the number of patterns to be learned: $L(0,9) - L(0,2) > 25\%$.

Application of these criteria clusters the 15 algorithms into four groups [Figs. 2(a) and S1]. Algorithms in group **"U"** are very sensitive to uninformative features, and the relative loss of predictability

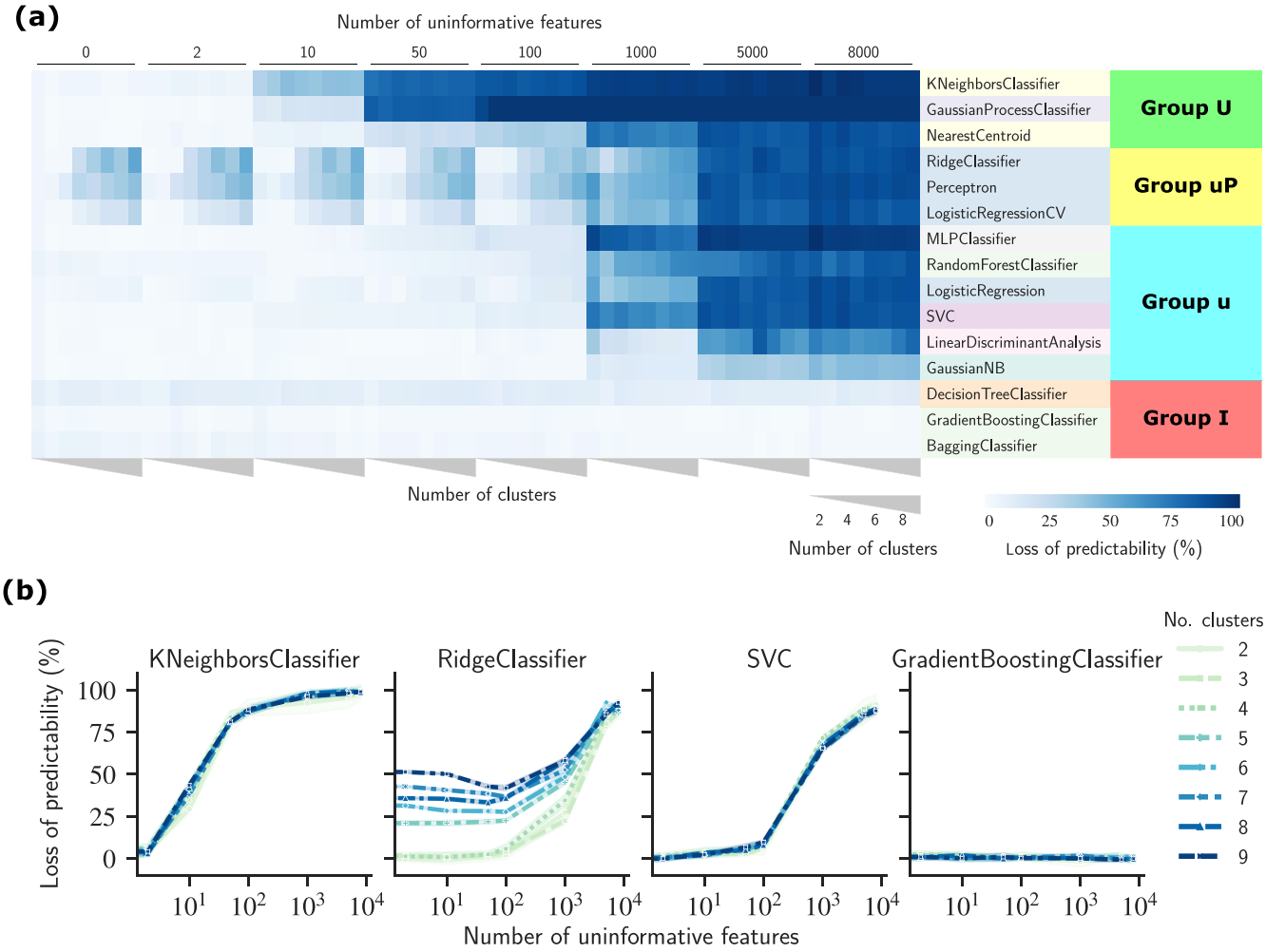

FIG. 2. The patterns of loss of predictability under an increasing number of uninformative features and clusters enable us to organize the algorithms into four groups. We generate datasets for 64 different pairs of the number of clusters and the number of uninformative features. (a) The heatmap displays the relative loss of predictability for 15 machine learning classifiers as a function of increasing numbers of clusters and uninformative features. A comparison of the impact of these two parameters enables us to organize these algorithms into four groups. Algorithms in group **U** are highly sensitive to the number of uninformative features. Algorithms in group **uP** are less sensitive than those in group **U** to the number of uninformative features but are highly sensitive to the number of clusters to be learned. Algorithms in group **u** are less sensitive to the number of uninformative features and insensitive to the number of clusters. Finally, algorithms in group **I** are insensitive to both the number of uninformative features and the number of patterns to be learned. (b) Examples of dependence of relative loss of predictability on the number of uninformative features and the number of clusters for one algorithm in each of the four groups. The lines show the average relative loss of predictability, and the shaded bands show one standard deviation.

starts increasing for even a small number of uninformative features. Algorithms in group **"uP"** are less sensitive to uninformative features but highly sensitive to the number of patterns to be learned. Algorithms in group **"u"** are less sensitive to uninformative features and mostly insensitive to the number of patterns to be learned. Finally, algorithms in group **"I"** are mostly insensitive to both uninformative features and the number of patterns to be learned. We display in Fig. 2(b) the relative loss of predictability against the number of uninformative features for one algorithm from each group. We show the full experimental results in Figs. S2, S3, and S4.

Changing signal-to-noise ratio

The impact of the number of uninformative features can be formulated in the context of the signal-to-noise ratio. As the separation between the clusters increases, the patterns in the data become more distinguishable. Similarly, as the training set size increases, one could expect that the likelihood that the pattern is learned also increases.

First, we investigate how the separation between clusters changes classification performance for datasets with two clusters. We generate ten synthetic datasets for each set of parameters. For

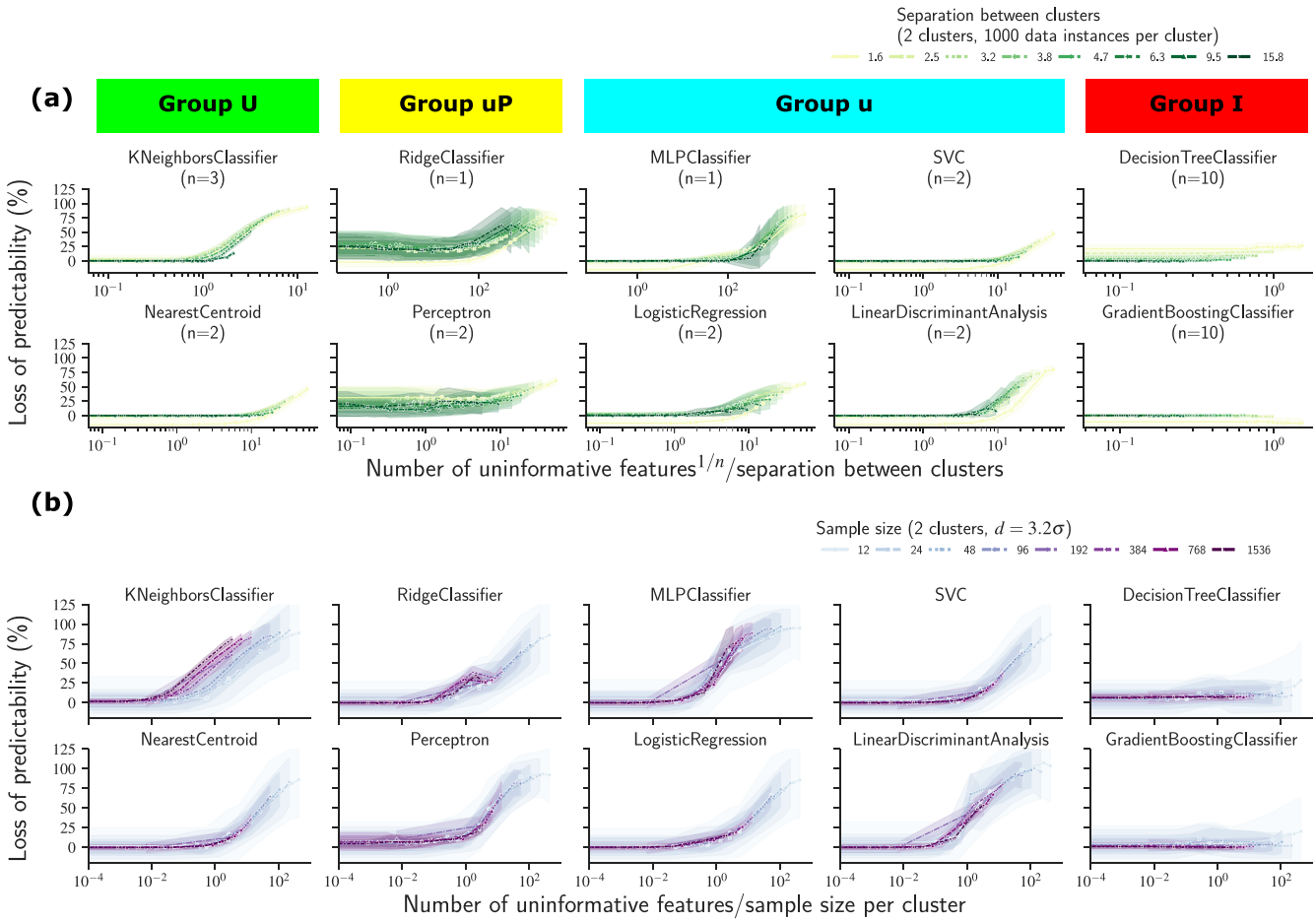

FIG. 3. Impact of larger separations between clusters and a higher number of data instances per cluster on the loss of predictability due to uninformative features. (a) We generate synthetic datasets with two Gaussian clusters and 1000 data instances per cluster for several values of the separation between clusters. The loss of predictability in learning algorithms can be mitigated by increasing the separations between clusters for some classifiers. By scaling the nth root of the number of uninformative features by the separation between clusters, we are able to collapse the relative loss of predictability for all different values of the separation between clusters. A larger root index represents a smaller minimal distance for distinguishing two clusters at the same dimensions. (b) We generate synthetic datasets with two Gaussian clusters and separation between clusters of 3.2σ for several numbers of data instances per cluster. We find that the loss of predictability in learning algorithms from uninformative features can be mitigated by increasing the number of data instances in each cluster. By scaling the number of uninformative features by the number of data instances per cluster, we are able to collapse the relative loss of predictability for all different values of the sample size for all algorithms except the KNeighbors classifier.

each dataset, we estimate accuracy using fivefold cross-validation. As shown in Fig. S5, while increasing the separation between clusters can mitigate the loss in relative predictability, a significant number of uninformative features still cause many algorithms to suffer from a substantial loss in their predictability even when clusters are well separated ($d > 5\sigma$). The MLP and RandomForest classifiers, in particular, do not significantly improve performance as the separation between clusters increases when the number of uninformative features exceeds 1000.

Inspired by the results on methods for separating high-dimensional Gaussian mixtures,[24–26] we investigate the impact of scaled separation between clusters for classification algorithms. Specifically, we scale the distance by $\sqrt[n]{m}$. As shown in Fig. 3(a), we are able to collapse the relative loss of predictability for all different values of separation between clusters when using the correct value of n (see Fig. S6 for lines under different root indices and different numbers of samples). A larger root index represents a smaller minimal distance for correctly classifying two clusters for the same value of m.

Our results suggest that tested algorithms have an n in the range of 1 to 3, except algorithms in group **"I"** that have a much larger root index. Note that the root index is estimated by visually inspecting data collapse, that is, it does not result from a theoretical analysis. Note also that the estimate of $n = 10$ for group **"I"** is quite inexact since the loss of predictability is very small and data collapse equally well under different values of n.

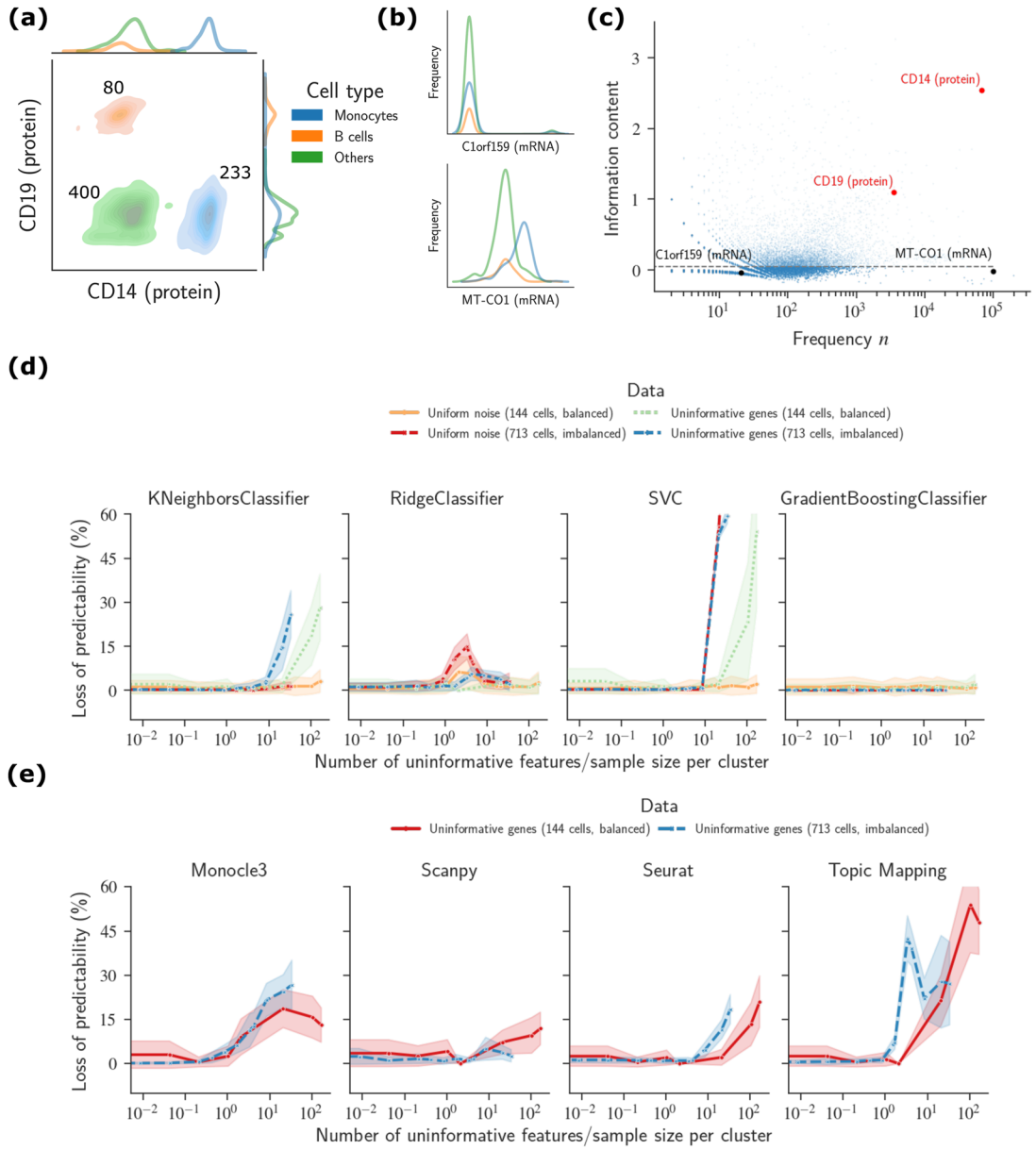

FIG. 4. Loss of predictability under an increasing number of uninformative genes in the CITE-seq dataset. The CITE-seq dataset contains (a) two informative features (the expression levels of surface proteins CD14 and CD19) that can separate monocytes (233 samples), B cells (80 samples), and other cells (400 samples). (b) The expression level of two selected uninformative/less-informative genes (C1orf159 and MT-CO1) from scRNA-seq data. (c) The information content of 17 467 genes and two surface proteins as a function of the frequency of genes or proteins. 8405 genes with an information content less than 0.05 are identified as uninformative/less-informative. (d) The loss of predictability on datasets with increasing uninformative genes for four classification algorithms. We generate four sets of datasets (cell balanced, cell imbalanced, synthetic balanced, and synthetic imbalanced) with increasing numbers of uninformative features from CITE-seq data. The lines show the average relative loss of predictability, and the shaded bands show one standard deviation. Cell datasets contain two informative features (expression levels of protein CD14 and CD19) and increasing numbers of uninformative genes as features. Synthetic datasets contain protein CD14, protein CD19, and increasing numbers of uninformative uniform features (as in Fig. 1). Balanced datasets use under-sampling to obtain 48 samples for each class. (e) The loss of predictability on datasets with increasing uninformative genes for four additional classification algorithms mainly used for biological sequencing data. We use the same two sets of datasets as above (cell balanced and cell imbalanced), which contain two informative features (expression levels of protein CD14 and CD19) and increasing numbers of uninformative genes as features from CITE-seq data. The lines also show the average relative loss of predictability, and the shaded bands show one standard deviation.

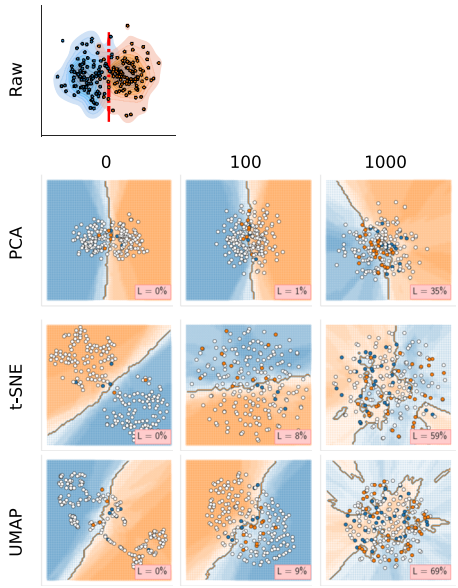

FIG. 5. Representational artifacts and loss of information of dimensionality reduction approaches in synthetic data. Dimensionality reduction is frequently used for the visualization of high-dimensional data and for feature reduction. However, some of these methods apply a nonlinear transformation that changes the data in uncontrolled ways. In the top panel, we show the raw data projected along the two truly informative features for a synthetic dataset, with 100 data instances per cluster and a separation of 3.2σ. The plots in the rows below compare the impact of a different number of uninformative features (across columns) for three-dimensional reduction approaches (across rows). Data points are shown in white if they are classified correctly and in the original color if they are classified incorrectly. The background color shows the decision boundary using a K-nearest neighbor classifier, with a more saturated color indicating a higher confidence. The corresponding loss of predictability is shown on the bottom right of each panel. It is visually apparent that PCA preserves more of the information in the discriminatory features than t-SNE or UMAP as the number of uninformative features increases. While for a low number of uninformative features, t-SNE and UMAP do not affect performance, they distort the representation in such a manner that they artificially increase viewers' confidence in how accurate the cluster classification is. That is, even though the two clusters appear highly separated in the t-SNE and UMAP representations, their degree of separation is not actually real.

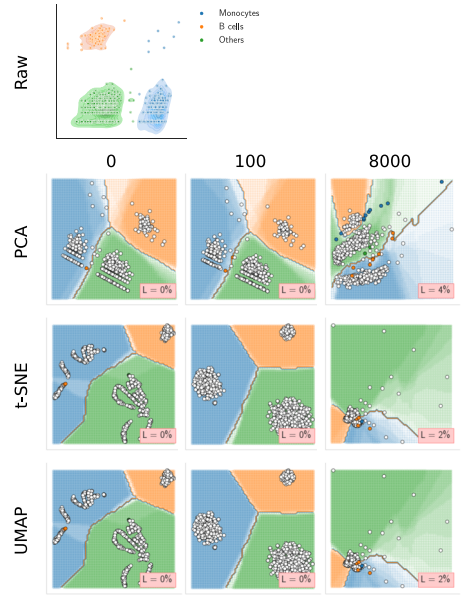

FIG. 6. Representational artifacts and loss of information of dimensionality reduction approach in CITE-seq data. In the top panel, we show the distribution of protein expression data. The plots in the rows below compare the impact of a different number of uninformative genes (0, 100, and 8000) for three-dimensional reduction approaches (PCA, t-SNE, and UMAP). Data points are shown in white if they are classified correctly and in the original color if they are classified incorrectly. The background color shows the decision boundary using a K-nearest neighbor classifier, with a more saturated color indicating a higher confidence. The corresponding loss of predictability is shown on the bottom right of each panel. While the loss of predictability is low, results from t-SNE and UMAP hint at the presence of spurious clusters when datasets have no uninformative features. This suggests the likelihood of false discovery when the ground truth is unknown. That is, even though the data are not separable, they appear to have multiple clusters.

The impact of the degree of separation among clusters depends on both n and the estimation of the threshold at which loss of predictability occurs. For example, if the separation between clusters is less than $\sqrt[4]{m}/10$, then SVC starts to lose predictability.

Next, we investigate how sample size impacts the performance of a learning algorithm when different numbers of uninformative features are present in the data. The required training sample size for particular learning algorithms is often unknown. Although a larger sample size can lead to a better estimate of model parameters, it also increases the computational cost and may be costly to obtain.

As shown in Fig. S7, the impact of uninformative features can be mitigated by sample size for most algorithms. We generate synthetic datasets with sample sizes ranging from 12 to 1536 data instances per cluster. To obtain more precise estimates of $L_k(m)$ for cases with small sample sizes, we generate 30 synthetic datasets with two clusters with a separation $d = 3.2\sigma$ for each set of parameter values. For each dataset, we estimate algorithm accuracies using five-fold cross-validation. We find that learning algorithms can benefit from increasing sample sizes except for the KNeighbors and GaussianProcess classifiers. For these two algorithms, the drop in relative predictability is less significant as the sample size increases, especially when large numbers of uninformative features are present in the training dataset.

To better understand this interplay between the sample size and the number of uninformative features, we scale the number of uninformative features by sample size. As shown in Fig. 3(b), the relative loss of predictability in Fig. S7 collapsed for all sample sizes for all algorithms, except the KNeighbors classifier. Our results suggest that the average number of data instances clustered in the training data must exceed the number of uninformative features to ensure a model with a loss of predictability smaller than 10%. This bound is lower by a factor of 100 for learning algorithms from group **"I."**

Generalizability to real-world data

To show the loss of predictability beyond synthetic data, we apply the same set of supervised algorithms and four additional unsupervised algorithms specifically designed for single-cell RNA sequencing data to a curated CITE-seq dataset (details of the data are provided in the "Methods" section) obtained from peripheral blood mononuclear cells (PBMCs). The dataset comprises expression levels of genes and of surface proteins for each cell. Using the surface protein data, we can rigorously assign each cell to one of the following three types: monocytes, B cells, and other cells. As shown in Fig. 4(a), these three classes of cells can be well separated by the expression levels of two surface proteins (CD14 and CD19). These classifications provide a ground truth for the results of the previously considered supervised classification algorithms. To investigate the impact of uninformative features, we create separate datasets by systematically adding data from genes with a low information content (defined as information content <0.05; see Ref. 27 for details) as additional features. Figure 4(b) shows the distinction across cells of the expression levels of two such uninformative genes (C1orf159 and MT-CO1). It is visually apparent that the information content of the two surface proteins is much higher [Fig. 4(c)].

The original dataset contains 713 cells with imbalanced classes (233 monocytes, 80 B cells, and 400 others). For a better comparison with the results from synthetic data, we also create balanced datasets with 48 instances from each class. Figures 4(d) and 4(e) show the loss of predictability as the number of uninformative features increases for the four representative algorithms (see Fig. S8 for all 15 algorithms and Fig. S9 for additional 11 algorithms) and the four additional unsupervised algorithms (Monocle3, Scanpy, Seurat, and Topic Mapping). Our analysis shows that the KNeighbors classifier, SVC, and Topic Mapping suffer significant losses in predictability as increasing numbers of uninformative genes are used as features.

We also create referencing synthetic datasets, where the signals are the expression levels of the same surface proteins and uninformative features are drawn from a uniform distribution as shown in Fig. 1. As shown in Figs. 4(d), S8, and S9, depending on the algorithm, datasets with uninformative genes have a better performance compared with their synthetic counterparts, likely due to the presence of some residual information in some of those genes. However, the small amount of information does not compensate for the impacts of extra dimensions. This leads to algorithms having a larger loss of predictability for datasets built with a large number of uninformative features.

Dimensionality reduction

Finally, we investigate how uninformative features impact the three most used dimensionality reduction approaches:[17,28] principal component analysis (PCA),[29] t-distributed stochastic neighbor embedding (t-SNE),[30,31] and uniform manifold approximation and projection (UMAP).[32] Figure 5 shows the two-dimensional embedding of datasets of two clusters with a separation $d = 3.2\sigma$ and 100 data instances per cluster for different numbers of uninformative features (see Fig. S10 for an investigation of synthetic data with 3–6 clusters). Our analysis reveals important insights. In the absence of uninformative features, t-SNE and UMAP produce artifacts that create an artificial separation of the two clusters that obscures the fact that some points cannot possibly be classified correctly. In contrast, PCA is able to maintain the reality of the actual separation between the clusters and the difficulty in correctly classifying some data instances.

The most remarkable result, however, is the impact of a large number of uninformative features. While PCA is mostly insensitive to even very large numbers of uninformative features, UMAP and, in particular, t-SNE create a great deal of intermixing of data instances belonging to different clusters. If one would use the KNeighbors classifier to identify clusters in the projected space, then the classification would yield large losses of predictability.

We also applied dimensionality reduction approaches to the CITE-seq data. Figure 6 shows the two-dimensional embedding of CITE-seq data with different numbers of uninformative genes. The results suggest that t-SNE and UMAP again hint at the presence of spurious clusters even when datasets have no uninformative genes due to the nature of distribution of signals. However, a striking observation is that spurious clusters merged into the ground truth clusters when about 100 uninformative features are added. Since the separations between the three clusters are large (see the "Methods" section for the statistics of clusters), we do not see a large loss in predictability when using the KNeighbors classifier to identify clusters in the projected space. However, clusters are visually indistinguishable when there are 8000 uninformative genes.

DISCUSSION

A key advantage of machine learning algorithms is their ability to learn the complex correlations between high-dimensional data that minimize desired loss functions. While this is a powerful feature in many contexts, one must contend with concerns about using machine learning for very high-dimensional data with potentially many uninformative features retained "for the sake of convenience." Here, we presented systematic experiments that benchmark a set of machine learning algorithms to investigate their limitations in handling uninformative features. Our study provides a reference for the user of machine learning algorithms when selecting suitable algorithms for different tasks. Our work highlights that algorithms relying on Euclidean distance are particularly ill-suited for tasks where one cannot filter out large numbers of uninformative features space.[21,33]

We recognize that our study has several limitations, which might reduce the generality of our conclusions. First, we do not cover deep learning approaches in this work. We believe that this is not a critical limitation as several studies have shown that they are not well-suited for panel data.[34,35] Second, we do not consider the computational efficiency of algorithms a significant factor when benchmarking because all the algorithms considered require modest computational resources and can be applied to data with the characteristics considered using a standard modern laptop. Third, some algorithms, including the Ridge classifier, have hyper-parameters that can be tuned to penalize less relevant features. In such cases, a grid search for a set of optimal hyper-parameters could be undertaken, but it is computationally costly. While hyper-parameter tuning using Bayesian optimization could reduce the computational

cost,[36] as shown in Fig. S11, tuned models do not appear to—at least for the KNeighbors classifier and Ridge classifier—significantly mitigate the loss in performance when the number of uninformative features is high. Fourth, we do not consider the possibility of correlations across independent variables. Studying such a case would require a very large number of additional computations. We believe that this is not a critical limitation as it is plausible to assume that, if anything, cross correlations would make the task of classifying data instances into clusters even more challenging. In such cases, the loss of predictability that we already identify due to uninformative features would likely become even more significant. Finally, we build all of the synthetic datasets using simple 2D Gaussian clusters with uninformative features drawn from uniform distributions. In real data, the distributions of values of different features are likely to be much more varied than this. However, one could easily hypothesize that the distribution of the values for uninformative features could be skewed and have low decaying tails, and under those conditions, fluctuations would be much more likely to result in the model over-fitting that occurs when the number of uninformative features increases. Indeed, our analysis of the CITE-seq dataset demonstrates that nonuniform, less informative features still cause algorithms to lose predictability.

Indeed, our study shows that uninformative features have the ability to introduce a sort of "phase transition" between a regime where an algorithm displays good predictability and a regime where predictability is lost. While such limits may appear unimportant for large datasets, they can nonetheless become a problem if there are a large number of clusters to be identified or if, instead of all clusters being equally represented in the data, some clusters are dominating.

METHODS

CITE-seq data

CITE-seq data are single-cell data comprising transcriptome measurements (gene expression) for each cell as well as surface protein expression level.[37] Our data on peripheral blood mononuclear cells (PBMCs) are openly available from 10x Genomics at https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/pbmc_1k_protein_v3.

TABLE II. Statistics of CITE-seq clusters.

First GMM using CD 14			
	Mean	Weight	Variance
B cells and other cells	2.77	0.67	0.62
Monocytes	7.20	0.33	0.23
Second GMM using CD 19			
	Mean	Weight	Variance
B cells	6.53	0.17	0.21
Other cells	1.10	0.83	0.44

For the purpose of our experiments, we focus on the expression level of two surface proteins (CD14 and CD19). These two proteins enable us to separate monocytes and B cells from other cells (T cells, NK cells, dendritic cells, etc.). We first use a Gaussian mixture model (GMM) on CD14 to separate monocytes (233 samples), and then, a second GMM is used to separate the rest of the cells into B cells (80 samples) and other cells (400 samples). Table II shows the statistics of the two GMMs. The table and Fig. 4 show that the clusters are well separated.

To investigate the impacts of uninformative features, we selected genes based on their information content.[27] Among 17 467 curated genes, we identify 8405 genes with an information content less than 0.05. The expression levels of those genes are then used as uninformative features. Finally, since the counts of genes and proteins are highly skewed, expression levels are represented by $\log(1 + count)$.[38]

SUPPLEMENTARY MATERIAL

The supplementary online material contains additional figures and tables.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation Grant No. 1937123.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

W.L., C.Z., and L.A.N.A. conceived and designed the study. W.L., C.Z., and Z.E.H. performed the numerical simulations. W.L., C.Z., Z.E.H., and L.A.N.A. performed the data analysis. W.L., C.Z., Z.E.H., and L.A.N.A. created the figures. W.L., C.Z., and L.A.N.A. wrote the first draft of the paper. W.L., C.Z., Z.E.H., and L.A.N.A. wrote, read, and approved the final version of the paper.

Weihua Lei: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Cleber Zanchettin**: Conceptualization (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Zoey E. Ho**: Conceptualization (supporting); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – review & editing (equal). **Luís A. Nunes Amaral**: Conceptualization (equal); Funding acquisition (lead); Methodology (equal); Project administration (lead); Resources (equal); Supervision (lead); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in https://github.com/amarallab/benchmarkml at http://doi.org/10.5281/zenodo.10055176.[39]

REFERENCES

[1]Y. Su *et al.*, "Multiple early factors anticipate post-acute COVID-19 sequelae," Cell **185**, 881–895.e20 (2022).

[2]J. Vamathevan *et al.*, "Applications of machine learning in drug discovery and development," Nat. Rev. Drug Discovery **18**, 463–477 (2019).

[3]C. Gao *et al.*, "Innovative materials science via machine learning," Adv. Funct. Mater. **32**, 2108044 (2022).

[4]D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, 2011), pp. 1100–1108.

[5]W. Lei, L. G. Alves, and L. A. N. Amaral, "Forecasting the evolution of fast-changing transportation networks using machine learning," Nat. Commun. **13**, 4252 (2022).

[6]T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," Simul. Modell. Pract. Theory **55**, 1–9 (2015).

[7]X. Zhu and X. Wu, "Class noise vs attribute noise: A quantitative study," Artif. Intell. Rev. **22**, 177–210 (2004).

[8]X. Wu and X. Zhu, "Mining with noise knowledge: Error-aware data mining," IEEE Trans. Syst. Man Cybern. A **38**, 917–932 (2008).

[9]D. F. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," Artif. Intell. Rev. **33**, 275–306 (2010).

[10]B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," IEEE Trans. Neural Netw. Learn. Syst. **25**, 845–869 (2014).

[11]A. Shanthini, G. Vinodhini, R. Chandrasekaran, and P. Supraja, "A taxonomy on impact of label noise and feature noise using machine learning techniques," Soft Comput. **23**, 8597–8607 (2019).

[12]X. Wang and A. Kaban, "Finding uninformative features in binary data," in *International Conference on Intelligent Data Engineering and Automated Learning* (Springer, 2005), pp. 40–47.

[13]C. Matsoukas *et al.*, "Adding seemingly uninformative labels helps in low data regimes," in *International Conference on Machine Learning* (PMLR, 2020), pp. 6775–6784.

[14]P. van Galen *et al.*, "Single-cell RNA-seq reveals AML hierarchies relevant to disease progression and immunity," Cell **176**, 1265–1281.e24 (2019).

[15]Z. Ren, M. Gerlach, H. Shi, G. S. Budinger, and L. A. Nunes Amaral, "Information-theory-based benchmarking and feature selection algorithm improve cell type annotation and reproducibility of single cell RNA-seq data analysis pipelines," bioRxiv:02.365510 (2020).

[16]M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," Nat. Rev. Genet. **16**, 321–332 (2015).

[17]Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," Bioinformatics **23**, 2507–2517 (2007).

[18]A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics* (Croatian Society for Information and Communication Technology, Electronics and Microelectronics, Rijeka, Croatia, 2015), pp. 1200–1205.

[19]L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," J. Mach. Learn. Res. **5**, 1205–1224 (2004).

[20]S. Baek, C.-A. Tsai, and J. J. Chen, "Development of biomarker classifiers from high-dimensional data," Briefings Bioinf. **10**, 537–546 (2009).

[21]R. Bellman, "Dynamic programming," Science **153**, 34–37 (1966).

[22]F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res. **12**, 2825–2830 (2011).

[23]J. A. Sáez, J. Luengo, and F. Herrera, "Fuzzy rule based classification systems versus crisp robust learners trained in presence of class noise's effects: A case of study," in *2011 11th International Conference on Intelligent Systems Design and Applications* (IEEE, 2011), pp. 1229–1234.

[24]S. Dasgupta, "Learning mixtures of Gaussians," in *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)* (IEEE Comput. Society, New York, 1999), pp. 634–644.

[25]S. Dasgupta and L. Schulman, "A two-round variant of EM for Gaussian mixtures," UAI'00: *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., San Francisco, CA, 2000) , pp. 152–159.

[26]A. Sanjeev and R. Kannan, "Learning mixtures of arbitrary Gaussians," in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing* (ACM, Hersonissos, Greece, 2001), pp. 247–257.

[27]M. Gerlach, H. Shi, and L. A. N. Amaral, "A universal information theoretic approach to the identification of stopwords," Nat. Mach. Intell. **1**, 606–612 (2019).

[28]C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," arXiv:1403.2877 (2014).

[29]S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," Chemom. Intell. Lab. Syst. **2**, 37–52 (1987).

[30]G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," Advances in Neural Information Processing Systems **15**, 833–840 (2002).

[31]L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," J. Mach. Learn. Res. **9**, 2579–2605 (2008).

[32]L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," J. Open Source Software **3**(29), 861 (2018).

[33]M. Sanderson, P. Raghavan, and H. Schütze, "Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University press. 2008. ISBN-13 978-0-521-86571-5, xxi + 482 pages," Nat. Lang. Eng. **16**, 100–103 (2010).

[34]R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," Inf. Fusion **81**, 84–90 (2022).

[35]L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," in Thirty-Sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.

[36]T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. O. Koyama, "A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Association for Computing Machinery, New York, 2019), pp. 2623–2631.

[37]M. Stoeckius *et al.*, "Simultaneous epitope and transcriptome measurement in single cells," Nat. Methods **14**, 865–868 (2017).

[38]I. Zwiener, B. Frisch, and H. Binder, "Transforming RNA-seq data to improve the performance of prognostic gene signatures," PLoS One **9**, e85150 (2014).

[39]W. Lei, C. Zanchettin, Z. Ho, and L. Amaral (2023). "Code and data for quantifying the impact of uninformative features on the performance of supervised classification and dimensionality reduction algorithms," Zenodo, v1.0.0. https://doi.org/10.5281/zenodo.10055176