

Feature subspace learning based on local point processes patterns

Yuting Ma  | Yuejing Ding | Tian Zheng

Department of Statistics, Columbia University,
New York, New York,

Correspondence

Tian Zheng, Department of Statistics, Columbia
University, New York, NY 10027.
Email: tzheng@stat.columbia.edu

High-dimensional data have provided vast amounts of information for scientific research and learning. However, in most cases, such information is buried in noise from noninformative features. Learning informative feature subspaces has become a necessary step for supervised learning tasks in high dimensions in order to improve accuracy and interpretability. The learning methods should also consider possible interactions among the features that may carry significant signals and reveal important scientific findings. In this paper, we develop a nonparametric measure of association between class label and continuous-valued feature subspaces using local point processes (LPP) patterns. A backward elimination algorithm based on random subspaces is used to identify informative feature subspaces according to this measure. Through simulations and real data applications, the proposed method demonstrates effectiveness in identifying patterns that are informative about the class difference not only marginally but also with higher-order interactions among features. As a result, the proposed method outperforms other popular feature selection methods with better generalizability and robustness.

KEYWORDS

feature subspace learning, flexible learning, local point processes, variable interaction

1 | INTRODUCTION

Current technological developments in many scientific fields have enabled researchers to generate data to explore scientific problems in more detail, on a larger scale, involving more possible factors, and in huge dimensions simultaneously. Among these exciting quests, 2 of the most frequently studied questions are [26] how to predict an outcome of interests well using the data collected (features, variables, inputs, etc.) and [7] which features, variables, or inputs contain information about the outcome or are “associated” with the outcome of scientific interest. In this paper, we are primarily interested in the cases where the outcome is a binary variable, that is, the binary classification problem in statistical learning.

High-dimensional data produced by scientific studies, such as that of gene expression in high throughput biological studies, provide researchers with a vast amount of information to study the sciences. However, at the same time, the large number of features also creates challenges for the statistical learning task at hand. It is partly due to the fact that true signals of class differences are embedded within

low-dimensional feature subspaces and are masked by a plethora of irrelevant and noisy features. The presence of irrelevant features hinders the performance of classifiers and hides the true relevant features. Moreover, for data of a complex nature, it is often the case that an individual feature that appears to be irrelevant when considered alone is influential only when jointly considered with other features. Evaluating features 1 at a time risks ignoring useful information carried by the features beyond the marginal distributional differences between classes. For instance, the threshold max problem introduced in [26] lays out the data structures that are often observed in real-world complex data that involve interactions, such as the coexpressions of genes. In such scenarios, each of the informative features provides very little information, but the problem is completely separable when the features are used together. To improve classification performance and discover important feature interactions, we devised an efficient method for identifying informative feature subspaces that take joint effects into account.

While traditional feature selection methods mostly fail when marginal information is weak, efficient and intuitive

feature selection methods for nonlinear and local classifiers are lacking in the literature [7]. In particular, the k -nearest neighbor (k NN) classifier, despite its simplicity, is widely used when data distributions are unknown and complex. The advantages of k NN include its model-free nature and universal consistency [28]. However, there are fundamental reasons why a traditional k NN classifier cannot be used to carry out feature selection. Firstly, k NN uses neighborhood majority votes rather than a discriminant function. It is therefore hard to introduce a notion of *margin*, as in the support vector machine (SVM) classifier, to achieve better generalization performance. Secondly, an ordinary k NN classifier can only be assessed using 0-1 loss. With the 0-1 loss, there is no incentive to move the decision boundary further away from training samples as in other loss functions, such as the exponential loss and the hinge loss. Hence, it is well known that k NN with the 0-1 loss is prone to overfitting. This limitation hinders k NN classifiers from being used as wrapper feature selection methods. Therefore, in order to carry out feature selection to exploit the flexibility of k NN classifiers, we need to introduce a feature selection approach that extends from the k NN classifier and incorporates a notion of “margin” to encourage maximum class separation.

In this paper, we present a flexible subspace feature learning framework based on point processes patterns in local neighborhoods that is capable of considering higher-order interactions among continuous-valued features while maintaining high interpretability. It aims at identifying a nonlinear complex association between a multidimensional input vector and a class label. Specifically, our approach leverages a well-known result that random observations from a probability distribution can be locally treated as homogeneous/heterogeneous *Poisson point processes* [32]. First, we define a novel measure of association between feature subspace and class label based on *local point processes* patterns, which is constructed in a nonparametric fashion without distributional assumptions. By recognizing the ordinal property of local point processes patterns in relation to the class label, we construct a notion of “margin” using the agreement between 2 ordinal variables. According to this measure, automatic searching algorithms based on random subspaces, such as a backward elimination algorithm, are then used to search for the most informative feature subspaces. The proposed method can be considered a *filter* method that takes higher-order interactions into consideration. Using information on local structures, the proposed feature selection framework is particularly versatile for high-dimensional data of a complex nature. Moreover, the computational efficiency can be further improved by evaluating feature subspaces in a distributed manner, making it favorable for large high-dimensional datasets.

The rest of the paper is organized as follows. Section 2 offers a brief review on the feature selection literature. In section 3, we define the local point processes (LPP) patterns and the association measure LPP_τ . The framework of feature

subspace learning based on LPP_τ is outlined in section 3.3. Section 4 presents some practical remarks for the efficient and scalable implementations of the proposed framework. In section 5, we compare the performance of the proposed method on simulated scenarios and real datasets with several popular feature selection methods in combinations with different classifiers. The results show that the proposed method based on LPP_τ outperforms the comparison methods and excels in cases that involve nonlinearity and higher-order interactions. Section 6 concludes with a discussion.

2 | BACKGROUND

There are 2 general types of approaches in feature selection: the *filters* approach and the *wrappers* approach. A filters approach focuses on the intrinsic characteristics of the data regardless of the classifiers used [20]. For instance, t -statistic-, or similar t -type statistic-, based algorithms are 1 popular category of filter methods, such as the Golub’s weighted voting method [14] and the Significance Analysis of Microarray (SAM) [30]. Dudoit et al. proposed a filter method based on F -ratio statistics of between-group to within-group sums of squares [11]. Information theory-/entropy-based methods [10] can also be viewed as filters approaches. The common characteristic of filter methods is that the evaluation criteria of features depend only on the marginal information of features. A wrappers approach, on the other hand, is related to a specific classification algorithm using classification performance as part of the feature evaluation criteria [23]. The most commonly used wrappers approaches are the SVM-based [31] feature elimination algorithms, such as *SVM-RFE* by Guyon et al. [15] and *R-SVM* by Zhang et al. [33], and the *Random Forests*-based feature selection algorithm by Breiman [5].

In high-dimensional data analysis, wrappers approaches generally enjoy an improved performance with their corresponding classifiers, but the selected features usually lack direct interpretability in terms of their relevancy to class labels. Filters approaches, on the other hand, may not improve classification accuracy as they are not designed for a specific classifier but boast better interpretability for the selection results and better generalization properties [10]. For example, in [33], *R-SVM* is compared with Golub’s weighted voting method on many simulated datasets based on known outcome variables and relevant features. The results show that *R-SVM* outperforms Golub’s method in prediction accuracy but is inferior to Golub’s method in finding relevant features.

Among various feature selection algorithms, neighborhood-based methods build up a special class by themselves. They use association structure in local neighborhoods, which is an intrinsic characteristic of the data as the criterion for feature evaluation. They are also closely related to the k NN classifier. Therefore, this family of methods can be regarded as a combination of both filters and wrappers approaches. One popular algorithm in this family

is the RELIEF algorithm based on the nearest neighbor distances [22]. In a binary classification problem, RELIEF defines a margin for the observation \mathbf{x}_i in feature j as $|x_{ij} - M(x_{ij})| - |x_{ij} - H(x_{ij})|$, where $M(x_{ij})$ is \mathbf{x}_i 's nearest neighbor with the same class label as \mathbf{x}_i , and $H(x_{ij})$ is its nearest neighbor of the same class. The goal is to find an optimal weight \mathbf{w} for all features to maximize the total margin:

$$\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathbb{R}^p} \sum_{i=1}^n \sum_{j=1}^p w_j (|x_{ij} - M(x_{ij})| - |x_{ij} - H(x_{ij})|),$$

where p is the number of features, and n is the sample size. Feature selection is therefore based on the optimal weight \mathbf{w}^* of each feature, which is regarded as a measure of feature importance. By treating each feature individually, however, the RELIEF algorithm does not consider the potential interactions among features and thus limits its usefulness regarding data of a complex nature.

3 | ASSOCIATION MEASURE BASED ON LPP PATTERNS

3.1 | Definitions

In this paper, we consider a binary classification problem with response variable $Y \in \{0, 1\}$, which is treated as an *ordinal variable* with “class 0” < “class 1.” The training data are given as $\mathcal{T} = \{(\mathbf{x}_1 y_1), (\mathbf{x}_2 y_2), \dots, (\mathbf{x}_n y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^p$ is the feature vector, and $y_i \in \{0, 1\}$ is the class label. Let $n_1 = \sum_{i=1}^n \mathbf{1}(y_i = 1)$ and $n_0 = \sum_{i=1}^n \mathbf{1}(y_i = 0)$ be the number of observations in class 1 and class 0, respectively. The total sample size is $n = n_1 + n_0$.

We denote a feature subspace as a set of features S , $\{x_s, s \in \{1, 2, \dots, p\}\}$. In a feature subspace S , the Euclidean distance is computed based on the subset of features, that is, by projecting the original feature space onto the feature subspace S :

$$d_S(\mathbf{x}_i \mathbf{x}_{i'}) = \sqrt{\sum_{j \in S} (x_{ij} - x_{i'j})^2}. \quad (1)$$

In the following, the nearest neighbors for an arbitrary point \mathbf{x}_i in S are obtained using $d_S(\mathbf{x}_i \cdot)$ correspondingly.

We consider the sequence of the class labels of its k NNs (from the nearest to the k th nearest neighbor) of an arbitrary observation \mathbf{x}_i . We define the ordered class label sequence of these neighbors as the LPP pattern of this observation.

$$\begin{aligned} \Phi^{(k)} &\triangleq \{\phi_1^{(k)} \phi_2^{(k)} \dots \phi_{2^k}^{(k)}\} \\ &= \{(11\dots 1)(11\dots 0) \dots (00\dots 0)\}. \end{aligned} \quad (2)$$

As the neighborhood size k is fixed throughout the discussions and examples in this paper, we omit the superscript k in the LPP pattern notations in the following paragraphs and write it as Φ . The j th LPP pattern in Φ , $\phi_j \triangleq (\phi_{j,1}, \phi_{j,2}, \dots, \phi_{j,k})$, is a binary vector of length k , where

$\phi_{j,l} \in \{0, 1\}$ is the class label of the l th nearest neighbor in ϕ_j , with $l = 1, 2, \dots, k$ and $j = 1, 2, \dots, 2^k$.

Given an observation \mathbf{x}_i and the feature subspace S , we define the mapping from \mathbf{x}_i to its LPP pattern Φ as $F_\Phi(\mathbf{x}_i S; \mathcal{T}) : \mathcal{X} \rightarrow \Phi$. Let $n_{1,j}$ and $n_{0,j}$ be the count of observations in class 1 and class 0 in S that has LPP pattern ϕ_j , $j = 1, 2, \dots, 2^k$, respectively:

$$n_{1,j}(S) = \sum_{i: y_i=1} \mathbf{1}(F_\Phi(\mathbf{x}_i S; \mathcal{T}) = \phi_j), \quad (3)$$

$$n_{0,j}(S) = \sum_{i: y_i=0} \mathbf{1}(F_\Phi(\mathbf{x}_i S; \mathcal{T}) = \phi_j). \quad (4)$$

We then define the *LPP profiles* of class 1 in feature subspace S , $V_1(S)$, as the proportion of each pattern in the training sample with class 1 and $V_0(S)$ for class 0:

$$V_1(S) = \frac{1}{n_1} (n_{1,1}(S), n_{1,2}(S), \dots, n_{1,2^k}(S))^T, \quad (5)$$

$$V_0(S) = \frac{1}{n_0} (n_{0,1}(S), n_{0,2}(S), \dots, n_{0,2^k}(S))^T. \quad (6)$$

And $V(S) = \frac{1}{n} (n_{+1}(S), n_{+2}(S), \dots, n_{+2^k}(S))^T$ is the *pooled LPP profile*, where $n_{+j} = n_{0,j}(S) + n_{1,j}(S)$, indicating the proportion of each pattern in both classes.

To understand the properties of LPP, we consider each LPP pattern as a vector of *rank statistics*. Among all sample points, let $(d_{1,(1)} \dots d_{1,(n_1)})$ and $(d_{0,(1)} \dots d_{0,(n_0)})$ be the ordered distances of class 1 points and class 0 points to \mathbf{x} , respectively—that is, $d_{1,(1)} \leq d_{1,(2)} \leq \dots \leq d_{1,(n_1)}$ —and similarly for $d_{0,(j)}$ s. Accordingly, for sample point \mathbf{x} with LPP pattern ϕ_j , its *rank representation* $\tilde{\phi}_j = (\tilde{\phi}_{j,1} \dots \tilde{\phi}_{j,n_1})$ is defined as the vector of rank values (in increasing order) of $d_{1,(j)}$ s among the pooled set of $d_{1,(j)}$ s and $d_{0,(j)}$ s censored at $k+1$, which is referred to as a *LPP rank pattern*. It has a bijective mapping with ϕ_j and is always of length n_1 with maximum value $k+1$. For instance, for a LPP pattern $\{11010\}$ with $k=5$, the corresponding rank representation is

$$\underbrace{\{1, 2, 4, k+1, \dots, k+1\}}_{\text{length of } n_1},$$

where $k=5$.

The advantage of using this alternative rank representation is 2-fold. Firstly, unlike the binary vector, the difference of 2 rank representations of LPP patterns contains more information about class difference, which can be exploited when determining the ordering of LPP patterns in section 3.2. Secondly, the fixed length-of-rank representation facilitates more efficient computation and storage allocation for better implementation.

An informative feature subspace is 1 where the distributions of 2 classes are notably different. The best possible classification performance of a classifier is the *Bayes rate*, that is, the error rate of the Bayes classifier that classifies a sample \mathbf{x} using the class densities at \mathbf{x} . The k NN classifier mimics the Bayes classifier by using the observed class labels of \mathbf{x} 's k NNs [17]. Therefore, it does not assume any model for

the class distributions and has the greatest advantage when the distributions are unknown and complex. The important underlying assumption of the k NN classifier is that, when n is large and k is relatively small, one can assume that the class densities are approximately constant within \mathbf{x} 's local neighborhood. It is easy to see that when $n \rightarrow \infty$, $k \rightarrow \infty$, and $k/n \rightarrow \infty$, the k NN classifier attains the Bayes rate. There is a well-established result that the observed samples in the neighborhood of a fixed \mathbf{x} can be regarded as being from a stationary *Poisson point process* in terms of their distances to \mathbf{x} [32]. However, in practice, where the sample size n is usually small and the dimension is high, distance is no longer informative about the closeness of sample points due to the curse of dimensionality [2]. Even the nearest neighbor of \mathbf{x} is far away from \mathbf{x} , and thus, a neighborhood of \mathbf{x} needs to be relatively large to contain its k -closest neighbors. As a result, the assumptions of constant density and stationary Poisson point processes do not hold.

Instead, we assume that the nearest neighbors are from inhomogeneous Poisson processes. Let the ordered distances $(d_{1,(1)} \dots d_{1,(n_1)})$ and $(d_{0,(1)} \dots d_{0,(n_0)})$ follow unknown continuous distributions F and G , respectively, that depend on the class distributions. Under mild conditions, the spacings of the ordered distances $d_{1,(i)} - d_{1,(i-1)}$, $i = 1, \dots, n_1$, with $d_{1,(0)} = 0$, are asymptotically independent and follow exponential distributions with rates $nf\left(\frac{i}{n}\right)$, where f is the density function of F [9,27]. Therefore, we can model the local points in \mathbf{x} 's neighborhood using the following nonstationary point processes defined by the distributions of the intervals between subsequent points:

$$\begin{aligned} \Delta_{1,i} &= d_{1,(i)} - d_{1,(i-1)} \sim \text{Exp}(\lambda_i^{(1)}), \\ \Delta_{0,i'} &= d_{0,(i')} - d_{0,(i'-1)} \sim \text{Exp}(\lambda_{i'}^{(0)}), \end{aligned} \quad (7)$$

both independently with $\lambda_i^{(1)} > 0$ and $\lambda_{i'}^{(0)}$ for $i = 1, \dots, n_1$ and $i' = 1, \dots, n_0$, respectively, and $d_{1,(0)} = d_{0,(0)} = 0$.

To see how LPP provides more information about class differences than the classical k NN counts, one can calculate that the probability of observing a specific LPP rank pattern $\tilde{\phi}_j$ at \mathbf{x} , or equivalently an LPP pattern ϕ_j , as follows

$$\begin{aligned} P(\phi = \phi_j | \mathbf{x}) &= \left(\prod_{i=1}^{k_1} \lambda_i^{(1)} \right) \left(\prod_{i'=1}^{k_0} \lambda_{i'}^{(0)} \right) \\ &\cdot \prod_{i=1}^{k_1+1} \left[\prod_{j=\tilde{\phi}_{j,i-1}+1}^{k_1+1} \frac{1}{\lambda_i^{(1)} + \lambda_{j-i+1}^{(0)}} \right], \end{aligned} \quad (8)$$

where $\tilde{\phi}_{j,0}$ is defined to be 0. In (8), the values of $\lambda_i^{(1)}$ and $\lambda_{i'}^{(0)}$ depend on the local densities of the 2 classes. If class 1 points cluster at \mathbf{x} (ie, have higher density at \mathbf{x} than class 0 points), $\lambda_i^{(1)}$ would be greater for smaller values of i than for larger values of i . Consequently, certain LPP patterns are observed more frequently where class 1 points cluster than others where class 0 points cluster. From (8), it is also easy to see that when $\lambda_i^{(1)} \equiv \lambda^{(1)}$ and $\lambda_{i'}^{(0)} \equiv \lambda^{(0)}$, as in stationary

Poisson point processes, different rank patterns with the same values of k_1 and k_0 will have the same probability. In such a case, the k NN counts used in the traditional k NN classifier are just as informative as the LPP rank patterns. When stationarity cannot be assumed, as is usually the case in practice, the LPP patterns (distance or rank) provide more information than the k NN counts when measuring the association between feature subspace and class label.

3.2 | Stochastic ordering of LPP patterns

Based on the rank representation of LPP patterns, we make the following assumption on the stochastic ordering of LPP patterns:

Assumption 1 (Stochastic ordering of LPP patterns). *For a feature subspace S that is “informative” about the class label, the following properties should hold:*

1. *For a point \mathbf{x} in this feature subspace, having an observed pattern with all k neighbors in class 1 is the strongest evidence in favor of \mathbf{x} being in class 1, and the pattern with all k neighbors in class 0 is the strongest evidence in favor of class 0.*
2. *Let $\delta_{jj'} \triangleq \tilde{\phi}_j - \tilde{\phi}_{j'}$ be the element-wise difference between 2 patterns $\tilde{\phi}_j$ and $\tilde{\phi}_{j'}$ in rank representation. If $\delta_{jj',k} \leq 0$ for all $k = 1, 2, \dots, n_1$ with at least 1 strict inequality, then ϕ_j is of stronger evidence for class 1 than pattern $\phi_{j'}$, denoted as $\phi_j \triangleleft \phi_{j'}$. If the elements in $\delta_{jj'}$ have difference signs, then the ordering between ϕ_j and $\phi_{j'}$ is undecided.*

Assumption indicates that, in an informative feature subspace, the class 1 points in the neighborhood centered at a class 1 point should be relatively closer to the “center” than in the neighborhood centered at a class 0 point. Under Assumption, if the feature subspace S is informative about the class label, LPP pattern ϕ_j s are *partially ordinal categories*, with $\{11 \dots 1\}$ most favoring class 1, and $\{00 \dots 0\}$ most favoring class 0. For some LPP patterns, however, the assumption does not provide a definite ordering. For example, when $k = 3$, the pair of patterns $\{100\}$ vs $\{011\}$ does not have a defined order under Assumption. For such cases, we compare the patterns using an empirical estimate of the class ratio R_j for pattern ϕ_j :

$$\hat{R}_j \triangleq \frac{n_{1,j}/n_1}{n_{0,j}/n_0} = \frac{n_{1,j}n_0}{n_{0,j}n_1}. \quad (9)$$

Then, $\phi_{j'} \triangleleft \phi_j$ if $R_{j'} < R_j$. However, if $R_{j'} = R_j$ or if the number of observations falling in either pattern is too small—that is, $(n_{1,j} + n_{0,j}) \leq 2$ or $(n_{1,j'} + n_{0,j'}) \leq 2$ — ϕ_j and $\phi_{j'}$ are treated as a “tie” ($\phi_j' = \phi_j$).

It is easy to see that the difference between the LPP profiles of the 2 classes implies a difference between class distributions in local neighborhoods. The greater the difference in LPP profiles, the more separable the class labels are in the given feature subspace. Therefore, the association between the profiles (or the distributions) of LPP patterns and the class label can be considered a measure of difference between

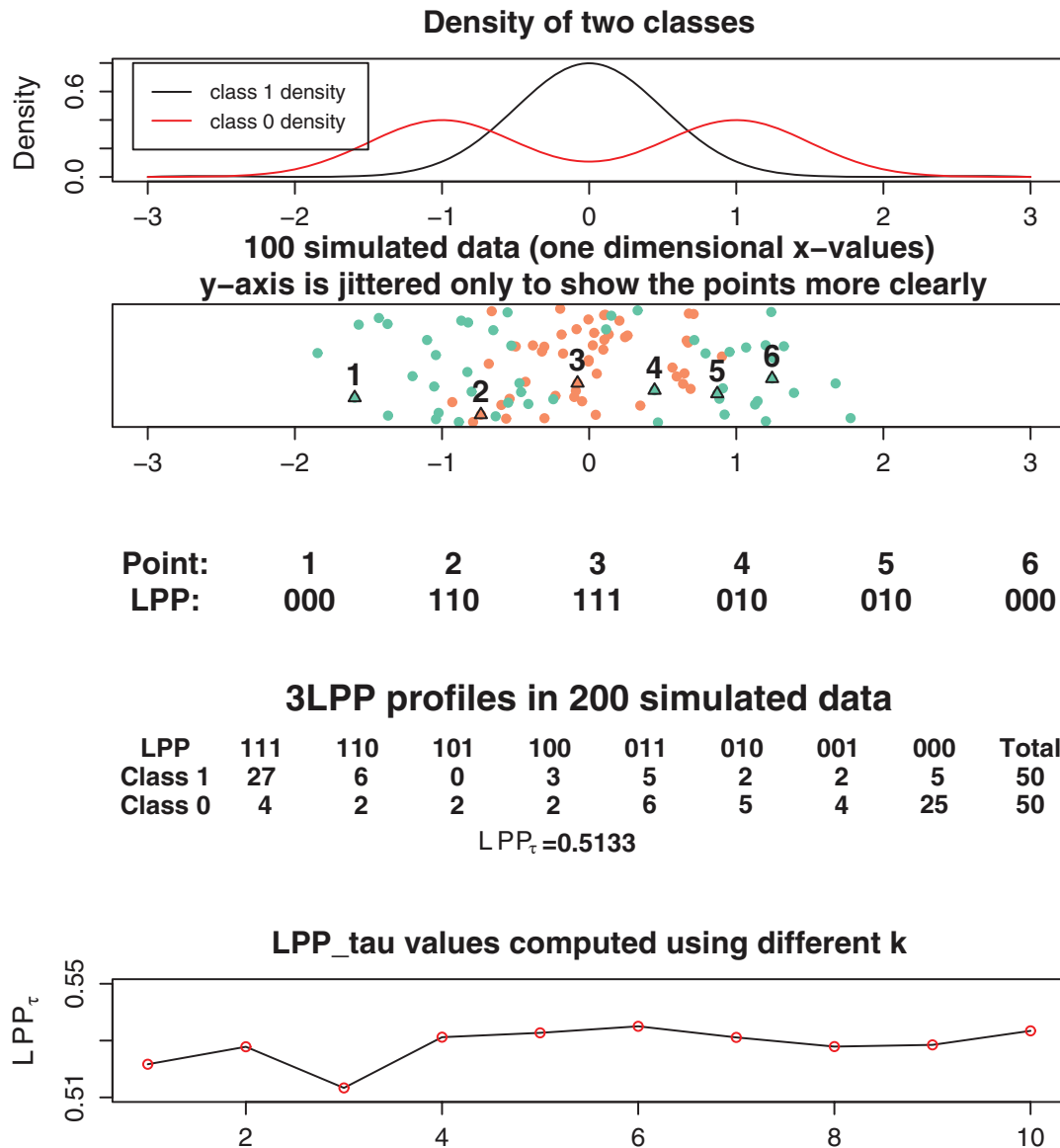


FIGURE 1 Illustration of different LPP profiles as evidence for different distribution between 2 classes

class distributions within the feature subspace. Moreover, it is worth noting that the ordinal property of LPP patterns provides more information about the class distribution in local neighborhoods than the ordinary counts in k -nearest neighborhood without sequencing.

In Figure 1, we illustrate the idea of LPP patterns. It shows that different LPP patterns with the same number of class 1 neighbors have different strengths of evidence in favor of class 1. In a 1-dimensional feature subspace, 50 points are simulated from 2 different but overlapping distributions of each class. In the neighborhood of observation 3 of class 1, the density of class 1 (orange points) is much higher than that of class 0 (green points), which is captured by the 3-LPP {111} of observation 3. In the neighborhood of observations 1 and 6 of class 0, the density of class 0 is much higher, and these 2 observations have their 3 closest neighbors in class 0. In the neighborhoods where the densities of the 2 classes are close, class impurity usually exists in the LPP pattern of the observations, and different LPP patterns

indicate different degrees of evidence in favor of class 1 or 0 (observations 2, 4, and 5). The 3-LPP profiles in Figure 1 summarize the 3-LPP for all 100 observations and show that the 3-LPP distributions of class 1 and 0 are different, indicating that this 1-dimensional feature is informative about the class difference.

From the 3-LPP table in Figure 1, we can see that {100} shows weak evidence in favor of class 1, while {010} and {001} both have relatively stronger evidence for class 0. If we use 3-nearest-neighbor counts instead, all of this evidence will be aggregated and weakened.

To show the advantage of using LPP patterns over ordinary k NN counts, in Appendix A, we provide a formal examination of Assumption under the assumption that points in local neighborhoods follow Poisson point processes, as suggested by well-known results in [32]. The stochastic ordering of LPP patterns in Assumption is demonstrated to be based on the probability of observing certain patterns under a mild and natural distributional condition.

3.3 | LPP_τ: a measure of class relevancy for a feature subspace

The properties and assumption of LPP patterns discussed in section 3.1 and 3.2 imply that the correlation between 2 *ordinal* variables, the class label $Y \in \{0, 1\}$ and the LPP pattern $\phi \in \{\phi_1, \phi_2, \dots, \phi_{2^k}\}$ in the feature subspace S , can be used as a measure of association between the feature subspace S and the class label. To this end, we choose to use Kendall's τ_b [21] to calculate the correlation between LPP patterns and the class label.

In feature subspace S , we define the pairwise comparison matrix of LPP patterns as $C = \{c_{ij}\}_{i,j=1,\dots,2^k}$ and formulate the absolute pairwise comparison matrix of LPP patterns as $D = \{d_{ij}\}_{i,j=1,\dots,2^k} = \{|c_{ij}|\}_{i,j=1,\dots,2^k}$, where

$$c_{ij} = \begin{cases} 1 & \text{if } \phi_i < \phi_j \\ -1 & \text{if } \phi_i > \phi_j \\ -0 & \text{if } \phi_i = \phi_j \end{cases}.$$

As defined in section 3.1, we assign the natural order of class label Y such that “class 1” > “class 0.” With the LPP profiles V_1 , V_0 , and V in feature subspace S computed according to (3) and (5), the measure of association between feature subspace S and the class labels with neighborhood size k , denoted as $\text{LPP}_\tau^{(k)}$, is defined by simple algebra as:

$$\text{LPP}_\tau^{(k)}(S) \triangleq \frac{V_1^T(S) C V_0(S) \sqrt{2n_1 n_0 / n^2}}{\sqrt{V^T(S) D V(S)}}. \quad (10)$$

As the neighborhood size k is fixed throughout the method, it is neglected in the following notations of LPP_τ .

By recognizing the ordinal properties of LPP patterns in relation with the class label, the LPP_τ measure in (10) serves as a notion of “margin” by quantifying the agreement between these 2 ordinal variables. Hence, in comparison with the traditional k NN classifier, the underlying classification using LPP_τ is better crafted to avoid overfitting by maximizing the “margin” between class boundary and sample points.

Similar to the k NN method, the choice of k (the number of neighbors to use) is a crucial question for methods based on LPP patterns. Kendall's τ_b is shown, in Argenti [1], to be robust in the choice of categories, corresponding to the choice of k in the LPP framework. To evaluate the effects of k , we compute and plot the values of LPP_τ for $k = 1, 2, \dots, 10$ on the simulated data in Figure 1. As shown, LPP_τ turns out to be quite stable to the choice of k . Part of this robustness stems from the partial order of LPP patterns. The rank statistics are expected to be more immune to fluctuations and noise from neighbors that are far away from \mathbf{x} when k is large.

For a d -dimensional feature subspace S , LPP_τ is a measure of the association between S and the ordinal class label. The larger value indicates stronger association. Therefore, we devise a heuristic feature selection algorithm based on LPP_τ

that searches for the feature subspaces with large LPP_τ values. For an arbitrary feature s in S , if it is not informative about the class difference, the $(d-1)$ -dimensional feature subspace $S/\{s\}$ will demonstrate a stronger association via a higher LPP_τ value. We prove in Appendix B that, under some distributional assumption, if feature s is not informative about the class label, LPP_τ will increase by deleting s from S . On the other hand, if feature s is informative about the class label, LPP_τ will decrease by deleting s . Therefore, a heuristic feature selection algorithm would be to select feature subspaces with LPP_τ values after evaluating all possible subsets of features. However, such an exhaustive search is computationally prohibitive when the number of features is usually in the thousands to tens of thousands.

In practice, a backward stepwise search or recursive elimination has been a popular algorithmic choice for feature ranking or selection, such as R-SVM in [33]. However, it was found that 1 backward search on all features is not efficient in identifying the most important variables that are associated with the outcome. This is due to several reasons. First, the search will not be informative initially due to the predominantly large number of noisy features in the evaluation set. Second, as the associated important features subspace may contain overlapping information about the class difference, the “optimal” subset with the best generalizability does not necessarily contain all important features. Therefore, we develop a random subset backward elimination algorithm for learning feature subspaces based on LPP_τ .

Within a large number of candidate d -dimensional feature subspaces, denoted as B , each feature subspace is compared with a subset of it in terms of LPP_τ values sequentially. If a subset of it achieves the same or an even higher LPP_τ value, there is redundancy in the feature subspace, and thus, the feature subspace should be substituted by the more concise subset while maintaining the same level of association with class label. The elimination procedure proceeds until no feature can be eliminated without a compromise on the LPP_τ value. The subspace spanned by the remaining features is called the LPP_τ *irreducible subspace*. It can be shown that this step is likely to delete noninformative features and only retain the informative ones. Finally, B_0 feature subspaces with the highest LPP_τ values are selected as the informative feature subspaces. To avoid the curse of dimensionality, we suggest a small value of d . The selection of parameters k , d , B , and B_0 are discussed in section 4. The greedy backward screening algorithm is elaborated in Algorithm 1.

However, for data with a large number of features, the greedy backward elimination algorithm becomes either computationally intractable or incapable of covering all features in random feature subspace selection with insufficiently large B . Therefore, in practice, instead of backward elimination, we examine all 2-dimensional feature subspaces using LPP_τ to capture important pairwise interactions. The pairwise interaction algorithm is summarized in Algorithm 2.

Algorithm 1 Feature Selection Algorithm based on LPP_τ: Greedy Backward Screening

```

1: Input Parameters:  $k, d, B$  and  $B_0$ .
2: Initialization: LPP patterns  $\Phi$  with corresponding  $k$ .
3: for  $h = 1$  to  $B$  do
4:   Randomly select a  $d$ -dimensional feature subspace  $\mathcal{S}_h$  and compute its LPPτ.
   Backward Elimination
5:   for each  $s \in \mathcal{S}_h$  do
6:     Let  $\tilde{\mathcal{S}}_h = \mathcal{S}_h / \{s\}$ , that is, remove feature  $s$  from  $\mathcal{S}_h$ .
7:     if LPPτ( $\mathcal{S}_h$ ) < LPPτ( $\tilde{\mathcal{S}}_h$ ), then
8:        $\mathcal{S}_h \rightarrow \tilde{\mathcal{S}}_h$ .
9:     end if
10:  end for
11:  Continue the backward elimination until no features can be deleted from  $\mathcal{S}_h$  without
  decreasing LPPτ( $\mathcal{S}_h$ ).
12: end for
13: return Select  $B_0$  feature subspaces with top LPPτ values as the informative feature subspaces
 $\Xi^* \triangleq \{\mathcal{S}_1^*, \mathcal{S}_2^*, \dots, \mathcal{S}_{B_0}^*\}$ .

```

Algorithm 2 Feature Selection Algorithm based on LPP_τ: Pairwise Interaction

```

1: Input Parameters:  $k, B$  and  $B_0$ .
2: Initialization: LPP patterns  $\Phi$  with corresponding  $k$ .
3: for  $h = 1, 2, \dots, p$  do
4:   Compute the LPPτ value for each univariate feature subspace  $\mathcal{S}_h = \{h\}$ .
5: end for
6: for  $h_1, h_2 = 1, 2, \dots, p$  do
7:   Compute the LPPτ value for each pairwise feature subspace  $\mathcal{S}_h = \{h_1, h_2\}$ .
8:   if LPPτ( $\mathcal{S}_h$ ) ≤ max(LPPτ( $\mathcal{S}_{h_1}$ ), LPPτ( $\mathcal{S}_{h_2}$ )), then
9:     Discard  $\mathcal{S}_h$ .
10:  end if
11: end for
12: return Select  $B_0$  feature subspaces with top LPPτ values as the informative feature subspaces
 $\Xi^* \triangleq \{\mathcal{S}_1^*, \mathcal{S}_2^*, \dots, \mathcal{S}_{B_0}^*\}$ .

```

4 | PRACTICAL REMARKS**4.1 | Tuning parameters**

Algorithms 1 and 2 are controlled by 4 major parameters corresponding to the given data. We elaborate the strategy for tuning these parameters in practical implementations as follows.

4.1.1 | Nearest neighborhood size k

As discussed in section 3.3, LPP_τ is quite robust to the value of k . In practice, we suggest relatively small values of k to avoid sparseness in the counts of LPP patterns with limited sample sizes. As shown in Tables 4 and 5, the performance of the proposed method is quite insensitive to the choice of k .

4.1.2 | Number of candidate subspaces B

The number of repeats B in Algorithms 1 and 2 is a trade-off between computational costs and a better chance to capture important subspaces. It should be large enough to capture at least all 2-dimensional feature subspaces with very high probability. This will ensure a high probability for the algorithm to capture at least 2-way interaction information:

P(features i and j are selected by Algorithm 1)

$$= 1 - \left(1 - \frac{d(d-1)}{M(M-1)}\right)^B > p,$$

for any i and j , where d is the dimension of the random subspaces, and M is the total number of features. This implies

$$B > \log(1-p) / \log\left(1 - \frac{d(d-1)}{M(M-1)}\right).$$

For example, if the starting number of dimensions is $d = 5$, the number of features is $M = 200$, and $p = 0.9999$, then $B \approx 20\,000$ should satisfy the above condition.

4.1.3 | Number of informative subspaces B_0

The number of B_0 sets the total maximum number of feature subspaces selected with fixed d , which is the most important parameter for tuning the proposed method. Increasing B_0 does not linearly increase the model complexity due to the averaging effect of the aggregated classifier and the overlap among significant feature subspaces. By aggregating feature subspaces, which is discussed in section 5.2, redundant feature subspaces will be assigned a weight close to 0. Hence, our algorithm is relatively robust to overselection, that is, to large B_0 . A more accurate tuning can be achieved by cross-validation. B_0 can also be determined using a stopping criterion within the learning process [6].

4.1.4 | Dimensionality of candidate subspaces d

The main factors contributing to choosing the dimension of candidate subspaces, d , are the data's sample size, the order of interactions among features that one wish to consider, and the computational cost. For data of a moderate sample size, sparsity of data points in high dimensions will seriously impair the informativeness of "nearest neighbors," and the computed LPP_τ becomes an unreliable assessment of the association between a subspace and the class label. In addition, the computational cost grows exponentially with d . In practice, for a dataset with a large number of candidate features, we suggest using $d = 2$ without backward elimination for controlling computational complexity, as in Algorithm 2. In the ordinary setting of Algorithm 1, d is constrained by the limited sample size. When the sample size n is small, large d will lead to the curse of dimensionality, rendering the k NNs noninformative about the class differences. Analytically, the sampling density of a single point in a p -dimensional unit ball is proportional to $N^{1/p}$. Thus, for an appropriate neighborhood of size k , d should be, at most, in the order of $\log(N)$ to avoid scarcity in feature spaces [17]. Within the upper bound, d determines the maximum order of interaction among features. Greater values of d exchange computational complexity for better classification accuracy if high-order interactions are truly informative about the class label.

4.2 | Scalable computation

The feature selection framework proposed in Algorithms 1 and 2 enables fast and scalable computation via parallelization. As each candidate feature subspace is evaluated independently without expensive communication, the backward eliminations based on LPP_τ on different candidate feature subspaces can be implemented on multiple cores of a high-performance computer simultaneously. In comparison, popular recursive methods, such as R-SVM, sequentially

learn the importance of features, which cannot be segregated in a distributed fashion for computational acceleration. This isolated structure among each computation unit makes the proposed algorithm highly efficient and favorable for high-dimensional data.

Furthermore, when computing the value of LPP_τ at each low-dimensional feature subspace, the search for k NNs can be performed efficiently with a kd -tree data structure [3]. In a low-dimensional candidate feature subspace, a kd -tree stores sample points in a tree structure. For each center point, it only takes $O(\log n)$ to search for nearest neighbors on average, instead of $O(n)$ with an ordinary search. For applications with large sample sizes, using k -d trees help expedite the feature selection procedure concurrently with parallelization.

5 | RESULTS

In this section, we evaluate the performance of the proposed methods on multiple simulation scenarios and well-known classification datasets. We compare it with several popular feature selection methods introduced in section 2 with different classifiers, including logistic regression, linear SVM, and SVM with the radial basis function (RBF) kernel. To conduct fair and consistent comparison of selected feature subspaces, we apply the feature subspaces aggregation via boosting to obtain classification results based on the framework of [13]. The results show that the proposed method based on LPP_τ performs favorably over the comparing methods in simulation studies. Our method even outperforms feature selection methods based on flexible nonlinear and local classifiers, such as KNN and RBF-SVM, especially when the marginal information is scarce and the noise level is high. Meanwhile, the resulting feature subspaces maintain good interpretability and computational efficiency. With real data, LPP_τ provides substantial improvement for a widely known hard dataset and more stable results for a relatively easy dataset.

We compare the proposed method with several well-established feature selection methods reviewed in section 1: the Golub's weighted voting method [14], F-ratio-based feature selection [11], RELIEF [22], and R-SVM [33]. Specifically, the Golub's weighted voting method and F-ratio-based feature selection, although using different measures, are both representatives of filter methods in which each input feature is evaluated independently without considering the joint effects among features. We choose RELIEF because it is based on distance information within local neighbors. Thus, it is particularly comparable to the proposed method based on LPP_τ . As a wrapper method, R-SVM is an example of the class of nonlinear feature selection methods. With these contrasting methods, we aim at evaluating the proposed LPP_τ not only on classification accuracy but on different aspects, including stability and the ability of exploiting local information and of extracting nonlinearity.

5.1 | Computational details

The computations are implemented using R* on x86_64 Red-hat Linux GNU system. In the simulation study, we use $k=3$ to construct LPP_τ in all examples. $B=5000$ candidate feature subspaces of $d=5$ were screened using Algorithms 1. B_0 was set to 1 to choose the most important feature subspace. In real data applications, in addition to the comparison with other feature selection methods, we also evaluate the performance of LPP_τ under different values of k , where $k \in \{3, 5, 7\}$, and different values of B_0 , where $B_0 \in \{10, 50, 100\}$. For computational efficiency, in real data applications, we implement Algorithm 2, which searches for informative pairwise interactions, that is, the dimension of features subspace was fixed at $d=2$, and $B = p = \frac{p(p-1)}{2}$ for exhaustive search.

To compare feature selection methods, we directly implement the F -ratio feature selection method and Golub's weight voting method based on the algorithm in the papers. For RELIEF, we use the R package CORElearn. We adapt R-SVM from the original code written by Zhang et al.[†] For classifiers, we use the R package wsvm to implement the weighted SVM algorithm with both linear and RBF kernels and use the R function glm to implement logistic regression.

5.2 | Feature subspaces aggregation

After informative feature subspaces[‡] have been selected using various feature selection methods, classifiers based on these feature subspaces need to be trained and aggregated to achieve the final class label prediction. The rationale behind classifier-aggregating algorithms is that if the classifiers have various association levels with the class label that do not complete overlap, aggregating should improve the prediction accuracy. Aggregating using equal weights (average vote) is only reasonable when the classifiers are independent and contain approximately equal amounts of information. However, due to overlapping features among selected subspaces and correlation among feature subspaces, classifiers based on these selected feature subspaces and trained on the same data are dependent. In addition, different subspaces usually contain heterogeneous amounts of information. Therefore, unequal weights for aggregating classifiers should be used to account for dependency among different subspaces, as well as the different credibility of each subspace, in order to achieve better classification performance than simply averaging votes from the classifiers.

Aggregating algorithms have been proposed in recent years for high-dimensional data classification to overcome overfitting and to improve prediction accuracy. These methods, such as bagging [4] and boosting [12], have enjoyed great popularity because of their good performance. However, their basic

frameworks are constructed based on the entire feature space so that each classifier is trained with all features, instead of on a feature subspace. On the contrary, the random subspace method introduced by Ho [18] represents a distinct aggregating method in which classifiers are learned by projecting the original feature space onto different low-dimensional feature subspaces. Using the AdaBoost algorithm [12] together with feature subspace projection in [18], we aggregate classifiers on selected feature subspaces by adapting the idea of boosting feature subspaces in [13]. At each boosting step with given weights, the algorithm searches for an optimal feature subspace that achieves the lowest error rate within the selected set of B_0 informative feature subspaces. Using this subspace, a classifier is trained and added to the aggregated classifier. The steps are elucidated in Algorithm 3.

The total number of boosting steps M can be tuned using cross-validation or other advanced methods, such as proposed in [6]. Nevertheless, our focus in this section is to achieve a fair comparison of different feature selection methods under the same conditions. As it is empirically shown that the performance of the boosting algorithm is considerably robust to large values of M , we set M to be 20 times the size of the set of feature subspaces B_0 in the following experiments for all feature selection methods and classifiers.

5.3 | Simulation study

In each simulation example, 2 informative features and 48 noisy features are generated. Figure 2 summarizes the simulation settings of the 3 examples (each with 3 noise levels). In total, 100 training data points (70 samples in the blue class, 30 samples in the red class) and 200 independent test samples (100 samples in blue class, 100 samples in red class) are drawn from bivariate normal distributions or bivariate normal mixtures, as illustrated by the contour plots in Figure 2; 100 simulations were carried out under each specification to evaluate the feature selection probabilities for individual informative features and the 2-dimensional informative subspace under different feature selection methods. The parameters of the feature selection methods are listed in Table 1.

As shown in Table 2, in the "easy example" (linearly separable case), Golub's weighted voting method and R-SVM perform equally well in all levels of noise. Neighborhood-based methods, RELIEF, and the proposed LPP_τ successfully pick up the first 2 dimensions in low-noise and median-noise scenarios but fail in the high-noise scenario. This is due to the unnecessary complexity caused by the noisy features. The results indicate that when no interaction is present and the marginal information is dominant, there is little to be gained by considering interaction information such as LPP_τ .

In the "little harder example," which is not linearly separable with only modest marginal information, LPP_τ achieves better performance than all the other comparing methods in the cases of low and median levels of noise. It is capable of identifying the first 2 informative dimensions and taking the

*<https://cran.r-project.org/>

[†]<https://web.stanford.edu/group/wonglab/RSVMpage/R-SVM.html>

[‡]In comparing feature selection methods, each feature subspace contains a single selected feature.

Algorithm 3 Boosting classifiers on feature subspaces

Input: Training data \mathcal{T} , a base classifier $f(x; \mathbf{w}, \mathcal{S})$ that is trained with sample weights $\mathbf{w} \in \mathbb{R}^n$ and training data \mathcal{T} on feature subspace \mathcal{S} and takes input \mathbf{x} , and a set of feature subspaces $\Xi = \{\mathcal{S}_1, \dots, \mathcal{S}_D\}$.

Initialization: $w_i = \frac{1}{n}$ for $i = 1, \dots, n$.

for $m = 1, 2, \dots, M$ **do**

1) Obtain the classifier $f_m(x) = f_{h^*}(x; \mathbf{w}, \mathcal{S}_{h^*})$, where

$$h^* = \arg \min_{h \in \{1, \dots, D\}} \text{err}_h$$

with

$$\text{err}_h = \frac{\sum_{i=1}^n w_i \mathbf{1}(f(x_i; \mathbf{w}, \mathcal{S}_h) \neq y_i)}{\sum_{i=1}^n w_i}.$$

2) Compute $\alpha_m = \log \left(\frac{1 - \text{err}_{h^*}}{\text{err}_{h^*}} \right)$.

3) Update weights $w_i \leftarrow w_i \exp(\alpha_m \mathbf{1}(f_m(x_i) \neq y_i))$ for $i = 1, \dots, n$.

end for

return The aggregated classifier $F(x) = \sum_{m=1}^M \alpha_m f_m(x)$.

interaction between them into consideration. However, when the marginal information is modest, LPP_τ masks the true signal when the noise level is high. In comparison, Golub's weighted voting method maintains its high accuracy in selecting informative features with all levels of noise. The eligibility of RELIEF decays when the noise level increases. R-SVM fails to pick up the second informative feature, which implies its lack of stability when the marginal information is weak.

In the “hardest example,” which is lack of any marginal information, LPP_τ significantly outperforms all the other comparing methods in all noise levels owing to its efficacy in capturing interaction information. The results also suggest the robustness of LPP_τ to increasing noise levels. None of the other comparing methods selects either informative dimension due to the absence of marginal information.

5.4 | Real data applications

The performance of LPP_τ is also evaluated on 2 public datasets, with data information presented in Table 3. For each dataset, the generalization performance is quantified using 5-fold cross-validation with 20 randomized partitions. We use 3 different values of k in the implementation of LPP_τ , $k \in \{3, 5, 7\}$. All comparing feature selection methods are integrated with 3 classifiers: logistic regression, linear SVM, and RBF SVM, with the number of selected features or feature subspaces B_0 varying from 10, 50, and 100. In training the SVM classifiers, the cost parameter is tuned using cross-validation from $\{2^{-5}, 2^{-3}, \dots, 2^3, 2^5\}$,

and the scale parameter of RBF kernel is tuned from $\{2^{-15}, 2^{-13}, \dots, 2^{-7}, 2^{-5}\}$. Finally, the classifiers are aggregated using Algorithm 3 to obtain the ensemble classifier. The cross-validation errors are shown in Tables 4 and 5 for MALDEON and SECOM datasets, respectively.

MADELON is an artificial dataset used in the NIPS 2003 challenge on feature selection[§] [16,25]. It contains sample points with binary labels that are clustered on the vertices of a 5-dimensional hypercube, in which these 5 dimensions constitute 5 informative variables. Fifteen linear combinations of these 5 variables were added to form a set of 20 (redundant) informative variables, while the other 480 variables have no predictive power on class label. This dataset is widely used for experiments in feature selection literature. It has been shown to be particular hard due to the weak marginal signal, which requires both the correct selection of relevant features and the ability of considering interaction features. Table 4 shows that the proposed feature selection method based on LPP_τ achieves much better classification accuracy than all the other comparing methods inasmuch as the higher-order interactions considered in feature subspace evaluations. In comparison, LPP_τ performs the best when B_0 is larger, in which case, there is better coverage of informative features. For LPP_τ , better results are found when integrated with the linear SVM classifier. Worse results

[§]The data are available at <https://archive.ics.uci.edu/ml/datasets/Madelon>. We use both the train data and the validation data. The 5-fold cross-validation is performed on the combined dataset.

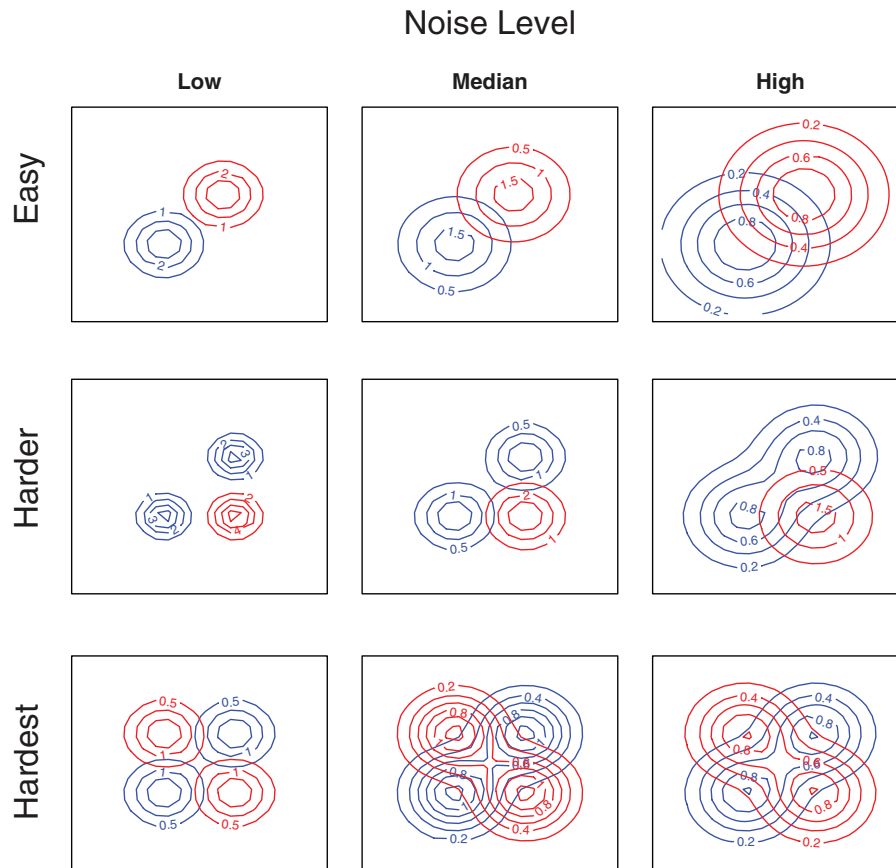


FIGURE 2 Simulation setup: the distribution of the 2 classes in the subspace of the 2 informative features

TABLE 1 The numbers of top subspaces (or features) selected by the feature selection methods in the simulation examples

Feature selection method	Number of top subspaces (or features) selected
Golub's method	10
R-SVM	10
RELIEF	10
LPP _r	1

TABLE 3 Data statistics

Data statistics	MADELON	SECOM
Input dimension	500	590
Sample size	2600	1567

are found with the RBF-SVM classifier as the unnecessary nonlinear structure introduced by the RBF kernel function may led to overfitting. In comparison, F -ratio and RELIEF

TABLE 2 Feature selection probabilities for individual informative features and the 2-dimensional informative subspace using different feature selection methods

Example	Noise Methods	Low		Median		High	
		Ind.	Both	Ind.	Both	Ind.	Both
Easy	Golub	1	1	1	1	1	1
	RELIEF	0.99	0.99	0.51	0.23	0.13	0.01
	R-SVM	1	1	1	1	1	1
	LPP _r	0.91	0.82	0.8	0.59	0.67	0.37
Hard	Golub	1	1	1	1	0.98	0.96
	RELIEF	0.98	0.97	0.66	0.41	0.11	0.02
	R-SVM	0.5	0	0.5	0	0.5	0
	LPP _r	1	1	1	1	0.74	0.62
Hardest	Golub	0	0	0.02	0	0.04	0
	RELIEF	0.42	0.12	0.12	0.01	0.07	0.01
	R-SVM	0.03	0.01	0.1	0.01	0.13	0.03
	LPP _r	0.99	0.99	0.79	0.79	0.43	0.42

The probabilities are estimated using 100 simulations.

TABLE 4 Classification results of MADELON data with different feature selection methods and the proposed method under different classifiers using 5-fold cross-validation

	B_0	Golub	F -ratio	RELIEF	LPP-7	LPP-5	LPP-3
Logistic	10	35	30.69	30.86	24.63	24.67	25.08
		(0.43)	(0.58)	(0.7)	(1.22)	(1.18)	(1.23)
	50	35.39	30.93	29.27	22.04	22.06	22.04
		(0.41)	(0.55)	(0.72)	(0.45)	(0.42)	(0.42)
	100	35.46	30.86	29.37	21.11	21.09	21.08
		(0.47)	(0.61)	(0.69)	(0.46)	(0.44)	(0.36)
Linear SVM	10	35.19	30.64	31.3	24.82	24.99	25.35
		(0.64)	(0.48)	(0.62)	(1.24)	(1.16)	(1.22)
	50	35.72	31.01	32.45	22.36	22.38	22.12
		(0.66)	(0.54)	(0.65)	(0.41)	(0.35)	(0.51)
	100	36.13	31.29	32.21	21.71	21.79	21.61
		(0.66)	(0.71)	(0.78)	(0.35)	(0.37)	(0.48)
RBF SVM	10	37.22	34.34	35.72	24.71	25.97	24.89
		(2.13)	(3.27)	(2.34)	(2.47)	(3.16)	(2.43)
	50	37.46	34.61	34.09	24.43	23.38	24.5
		(2.47)	(2.31)	(2.5)	(3.58)	(3.38)	(3.54)
	100	37.61	34.4	35.06	23.12	23.42	24.54
		(1.49)	(2.25)	(2.92)	(2.54)	(2.95)	(2.98)

The bold numbers indicate the lowest cross-validation error under each classification method and B_0 combination.

We show the mean percentage (standard deviation) of cross-validation errors with 20 randomized partitions. In this dataset, the feature selection method RSVM is omitted in the comparison due to its prohibitively high computational cost with large sample size.

TABLE 5 Classification results of SECOM data with different feature selection methods and the proposed method under different classifiers using 5-fold cross-validation

	B_0	Golub	F -ratio	RELIEF	R-SVM	LPP-7	LPP-5	LPP-3
Logistic	10	7	7.53	6.98	6.99	6.9	6.93	6.94
		(0.07)	(0.21)	(0.04)	(0.06)	(0.07)	(0.07)	(0.07)
	50	7.41	8.45	7.64	8.44	7.36	7.77	7.74
		(0.2)	(0.35)	(0.15)	(0.13)	(0)	(0.02)	(0.02)
	100	8.85	10.33	9.44	8.7	8.77	7.79	8.29
		(0.05)	(0.09)	(0.03)	(0.14)	(0.08)	(0.08)	(0.08)
Linear SVM	10	6.96	7.01	6.96	6.98	6.83	6.83	6.83
		(0.13)	(0.09)	(0.13)	(0.04)	(0.04)	(0.04)	(0.04)
	50	6.98	7.03	6.97	6.82	6.87	6.88	6.86
		(0.03)	(0.11)	(0.09)	(0.24)	(0.1)	(0.12)	(0.12)
	100	6.96	7.16	7	6.96	6.87	6.87	6.87
		(0.07)	(0.13)	(0.07)	(0.12)	(0.08)	(0.08)	(0.08)
RBF SVM	10	6.97	7.03	6.94	6.98	6.83	6.87	6.87
		(0.03)	(0.09)	(0.07)	(0.04)	(0.08)	(0.09)	(0.06)
	50	6.97	7.09	7	6.75	6.88	6.88	6.82
		(0.39)	(0.9)	(0.35)	(0.5)	(0.03)	(0.01)	(0.02)
	100	6.97	7.1	7.07	6.98	6.86	6.85	6.68
		(0.02)	(0.15)	(0.08)	(0.04)	(0.04)	(0.04)	(0.04)

The bold numbers indicate the lowest cross-validation error under each classification method and B_0 combination.

We show the mean percentage (standard deviation) of cross-validation errors with 20 randomized partitions.

perform slightly better than Golub's weighted voting method, which is solely based on marginal information. The test error achieved by LPP_r also outperforms state-of-the-art methods beyond the listed comparable methods on the MADELON dataset [24,29].

SECOM[¶] contains measurements from sensors for monitoring the function of a modern semiconductor manufacturing

[¶]The data are available at <https://archive.ics.uci.edu/ml/datasets/SECOM>. The original data are trimmed by taking out variables with constant values

process [25]. This dataset is a representative real-data application in which not all input variables are equally informative. The measured signals from the sensors contain irrelevant information and high noise, which mask the true information from being discovered. Under such a scenario, accurate feature selection methods are proven to be effective in reducing the test error significantly as well as in identifying the most relevant signals [25]. As shown in Table 5, LPP_τ achieves slightly lower mean cross-validation error and lower variance, suggesting better stability of LPP_τ . In the cross-comparison of classifiers, all feature selection methods perform badly with logistic regression, which may be due to the nonlinear class boundary inherit in the true low-dimensional feature subspaces. This observation is also consistent with the overall better performance with the RBF SVM classifier. As compared to other feature selection methods, LPP_τ is able to identify nonlinearity in the data by exploiting information within local neighborhoods. As B_0 increases, it also exhibits better control of overfitting.

In summary, the proposed feature selection method with LPP_τ mostly outperforms comparable methods. With different sizes of local neighborhoods, both datasets demonstrated that LPP_τ is robust with different choices of k , consistent with the example in Figure 1. By using local information and taking higher-order interactions into consideration, LPP_τ -based feature selection has been demonstrated to be favorable to datasets with complex nonlinear structures.

6 | CONCLUSION

In this paper, we proposed a novel and heuristic feature selection algorithm based on a nonparametric measure of association between class label and feature subspace, LPP_τ . The feature selection is carried out using an automatic backward searching algorithm. With both simulation study and real data applications, the proposed method has demonstrated its ability to identify informative low-dimensional, continuous-valued feature subspaces in data with complex structures, owing to its merits in exploiting local information and in capturing higher-order interactions. Furthermore, our method is robust to different values of parameters, the neighborhood size, and is more stable under different classifiers than other popular feature selection methods. Overfitting is also well controlled under the framework of LPP_τ -based feature selection.

Our proposed LPP_τ measure is currently defined using Euclidean distance on continuous-valued features. The general result of this paper also applies to other continuous distance norms. For discrete variables, the proposed frame-

work does not work efficiently as there are too many ties among the observations.

An assumption we made in this paper is that, for high-dimensional data of moderate sample size, data points in a local neighborhood are distributed following inhomogeneous Poisson processes. This is different from the conventional assumption for KNN methods that points in a small neighborhood approximately follow a homogeneous Poisson point process, which only holds in data-rich situations. Our method is motivated by deviation from this ideal data-rich scenario, which is often observed in practice. If the assumption of inhomogeneous Poisson processes does not hold, that is, the arrivals of nearest neighbors instead follow a homogeneous Poisson process, our proposed method will be approximately equivalent to the original KNN classifier.

So far, we have restricted our attention to the binary classification problem, but the LPP_τ -based algorithm can be extended to the multiclass problem as well. In a binary classification problem, there are 2^k LPP patterns in total. A direct generalization of LPP to an m -class problem results in a total of m^k LPP patterns. One concern for such direct generalization is that the number of observations in each LPP pattern will be very small when m is large and sample size n is small, rendering sparse LPP profiles. Although the LPP_τ value will not be affected too much, the sparseness is not good for classifiers based on LPP_τ . An alternative generalization method is to integrate binary classifiers into a multiclass problem. Possible solutions include the all-pairs (AP) algorithm, one-vs-all algorithm (OVA), or the tree-based integration of the *one-vs-some* classifier (OVS) [34].

ORCID

Yuting Ma  <http://orcid.org/0000-0003-4514-6639>

REFERENCES

1. A. Agresti, *The effect of category choice on some ordinal measures of association*, J. Am. Stat. Assoc. 71 (1976), 49–55.
2. R. Bellman and R. Bellman, *Adaptive control processes: a guided tour*, Princeton Univ. Press, Princeton Legacy Library, 1961.
3. J. L. Bentley, *Multidimensional binary search trees used for associative searching*, Commun. ACM 18 (1975), 509–517.
4. L. Breiman, *Bagging predictors*, Mach. Learn. 24 (1996), 123–140.
5. L. Breiman, *Random forests*, Mach. Learn. 45 (2001), 5–32.
6. P. Bühlmann and B. Yu, *Sparse boosting*, J. Mach. Learn. Res. 7 (2006), 1001–1024. MR2274395
7. G. Chandrashekar and F. Sahin, *A survey on feature selection methods*, Comput. Electr. Eng. 40 (2014), 16–28.
8. S. N. Chiu et al., *Stochastic geometry and its applications*, John Wiley & Sons, 2013. MR3236788
9. D. R. Cox and D. V. Hinkley, *Theoretical statistics*, CRC Press, 1979. MR0518594
10. C. Ding and H. Peng, *Minimum redundancy feature selection from microarray gene expression data*, J. Bioinform. Comput. Biol. 3 (2005), 185–205.
11. S. Dudoit, J. Fridlyand, and T. P. Speed, *Comparison of discrimination methods for the classification of tumors using gene expression data*, J. Am. Stat. Assoc. 97 (2002), 77–87. MR1963389
12. Y. Freund, R. E. Schapire, et al., *Experiments with a new boosting algorithm*, ICML 96 (1996), 148–156.
13. N. García-Pedrajas and D. Ortiz-Boyer, *Boosting random subspace method*, Neural Netw. 21 (2008), 1344–1362.

and variables with more than 10% of missing values so that the dimension is reduced from 591 to 414. Observations with missing values after the trimming of variables are discarded in this experiment, which reduces the sample size to 1436.

14. T. R. Golub et al., *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*, Science 286 (1999), 531–537.
15. I. Guyon et al., *Gene selection for cancer classification using support vector machines*, Mach. Learn. 46 (2002), 389–422. MR2765052
16. I. Guyon, J. Li, T. Mader, P. A. Pletscher, G. Schneider and M. Uhr, *Feature selection with the clop package*, 2006, Technical report, available at <http://clopinet.com/isabelle/Projects/ETH/TM-fextract-class.pdf>.
17. T. Hastie et al., *The elements of statistical learning: data mining, inference and prediction*, Math. Intell. 27 (2005), 83–85. MR2722294
18. T. K. Ho, *The random subspace method for constructing decision forests*, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998), 832–844.
19. J. S. Huang, *Characterizations of the exponential distribution by order statistics*, J. Appl. Prob. 11 (1974), 605–608. MR0381071
20. I. Inza et al., *Filter versus wrapper gene selection approaches in dna microarray domains*, Artif. Intell. Med. 31 (2004), 91–103.
21. M. G. Kendall, *The treatment of ties in ranking problems*, Biometrika 33 (1945), 239–251. MR0016601
22. K. Kira and L. A. Rendell, *A practical approach to feature selection*, In *Proc. Ninth Int. Workshop Mach. Learn.*, 1992, 249–256.
23. R. Kohavi and G. H. John, *Wrappers for feature subset selection*, Artif. Intell. 97 (1997), 273–324.
24. M. B. Kursa, W. R. Rudnicki, et al., *Feature selection with the boruta package*, J. Stat. Softw. 36 (2010).
25. M. Lichman, *UCI machine learning repository*, Univ. of California, School of Information and Computer Sciences, Irvine, 2013, available at <http://archive.ics.uci.edu/ml>.
26. S. Perkins, K. Lackner, and J. Theiler, *Grafting: fast, incremental feature selection by gradient descent in function space*, Journal of Machine Learning Research Vol. 3 Mar (2003), 1333–1356.
27. R. Pyke, *Spacings*, J. R. Stat. Soc. B. Methodol. (1965), 395–449.
28. C. J. Stone, *Consistent nonparametric regression*, Ann. Stat. (1977), 595–620. MR0216622
29. T. Turki and U. Roshan, *Weighted maximum variance dimensionality reduction*, In *Pattern Recognition*, Springer, 2014, 11–20.
30. V. G. Tusher, R. Tibshirani, and G. Chu, *Significance analysis of microarrays applied to the ionizing radiation response*, Proc. Natl. Acad. Sci. U. S. A. 98 (2001), 5116–5121.
31. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, 2013. MR1719582
32. X. Yan and T. Zheng, *Selecting informative genes for discriminant analysis using multigene expression profiles*, BMC Genomics 9 (2008), S14.
33. X. Zhang et al., *Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data*, BMC Bioinformatics 7 (2006), 1.
34. T. Zheng and Y. Ding, *Tree-based integration of one-versus-some classifiers for multiclass classification*, In *Proc. Joint Stat. Meet.* 2006, 2007.

How to cite this article: Ma Y, Ding Y, Zheng T. Feature subspace learning based on local point processes patterns. *Stat Anal Data Min: The ASA Data Sci Journal*, 2018;11:32–50. <https://doi.org/10.1002/sam.11368>.

APPENDIX A: EXAMINATION OF ASSUMPTION 1

A1 | MODEL ASSUMPTION

To justify the Assumption I in section 3.2 and to show the advantage of using LPP patterns over ordinal k -Nearest Neighbor (k NN) counts, we consider the following model.

Given a d -dimensional subspace S , consider a class 1 point \mathbf{x} . In the local neighborhood, $b(\mathbf{x})$, of \mathbf{x} , the class 1 points

are assumed to have an *inhomogeneous Poisson point process* where the d th power of the distance between \mathbf{x} and a class 1 point in $b(\mathbf{x})$ follows an *exponential distribution* $\exp(\lambda_1)$. From the classic result of exponential distribution [19,20], if X_1, \dots, X_n follow exponential distribution with rate λ_1 (denoted as $\exp(\lambda_1)$) and denote $X_{(1)}, \dots, X_{(n)}$ as the order statistic of X_1, \dots, X_n with $X_{(0)} \triangleq 0$, we have $\delta_i = (n - i + 1)(X_{(i)} - X_{(i-1)})$, $i = 1, \dots, n$ i.i.d distributed as $\exp(\lambda_1)$. In other words,

$$D_i \triangleq X_{(i)} - X_{(i-1)} \sim \exp((n - i + 1)\lambda_1), \text{ for } i = 1, \dots, n,$$

and D_1, \dots, D_n are independent.

For class 0 points in $b(\mathbf{x})$, we assume a *homogeneous Poisson point process* with intensity rate λ_0/w_d , where $w_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)}$ is the volume of a unit ball in \mathbb{R}^d . Denote the distance between the fixed point \mathbf{x} and its nearest neighbor in class 0 as R . According to the classic results on spatial homogeneous Poisson point process [8], R has the cumulative distribution function

$$F_R(r) = 1 - \exp\left(-\left(\frac{\lambda_0}{w_d} \cdot w_d\right) R^d\right) = 1 - \exp(-\lambda_0 R^d).$$

In other words, $Z = R^d$ follows exponential distribution with rate λ_0 . Denote the distance of \mathbf{x} 's 1st nearest neighbor, 2nd nearest neighbor, \dots , m th nearest neighbor in class 0 to \mathbf{x} as R_1, \dots, R_m , and $Z_i = R_i^d$. From [8], we also have

$$D_i = Z_i - Z_{i-1} \stackrel{\text{i.i.d}}{\sim} \exp(\lambda_0).$$

Given the fixed neighborhood and the fixed number of points in the neighborhood m , Z_1, \dots, Z_m follow uniform distribution.

This model is reasonable for cases where \mathbf{x} is located at the place where the density of class 1 points is high and decreases as distance to \mathbf{x} increases, and where the density of class 0 points is low and more stable. Figure A1 is a graphical illustration of this model assumption under 2 bivariate normal density settings.

Assume the points in class 1 and class 0 follow 2 bivariate normal distribution, respectively, as shown in Figure A1. The neighborhood is chosen to be in the area where the density of class 1 is high, while the density of class 0 is low. The neighborhood is chosen slightly away from the center of class 1 so that the result can have more generalizability. Fixing the number of observations from class 1 in the neighborhood (shown in Figure A1) to be 100 and the number of observations from class 0 in the neighborhood to be 30, we simulated the 100 samples from class 1 and 30 samples from class 0 within the neighborhood. From Figure A1, we can see that the histogram of the squared distances of the class 1 points to the fixed point \mathbf{x} distribute close to an exponential distribution, and those of the class 0 points seem uniformly distributed.

Theoretically, assume that the densities of the 2 classes (1 and 0) are multivariate normal distributions. Considering the fixed point \mathbf{x} to be the center of class 1, we have the normalized squared distance R^2/σ^2 from any point of class 1 to its

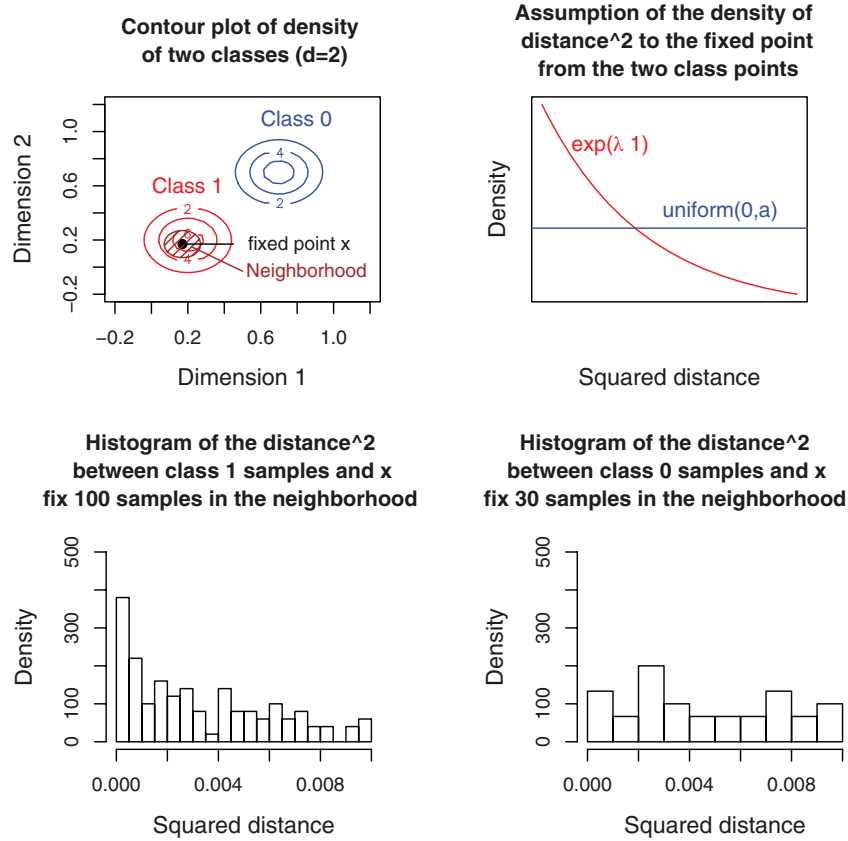


FIGURE A1 Illustration of Assumption using bivariate normal distributions

mean following Chi-square distribution $\chi^2(d)$. Denote $Z = R^d$ as the d th power of the distance from a class 1 point to \mathbf{x} , then

$$\begin{aligned} P(Z < z) &= P(R^2 < z^{2/d}) \\ &= P\left(\frac{R^2}{\sigma^2} < \frac{z^{2/d}}{\sigma^2}\right) \\ &= \int_0^{\frac{z^{2/d}}{\sigma^2}} \frac{1}{2^{d/2}\Gamma\left(\frac{d}{2}\right)} r^{d/2-1} e^{-r/2} dr. \end{aligned}$$

So, the density function of Z is

$$\begin{aligned} f_Z(z) &= \frac{2}{d\sigma^2} z^{2/d-1} \frac{1}{2^{d/2}\Gamma\left(\frac{d}{2}\right)} \left(\frac{z^{2/d}}{\sigma^2}\right)^{d/2-1} e^{-\frac{z^{2/d}}{2\sigma^2}} \\ &= \frac{1}{d2^{d/2-1}\sigma^d\Gamma\left(\frac{d}{2}\right)} e^{-\frac{z^{2/d}}{2\sigma^2}}. \end{aligned}$$

This pattern would also be approximately true for most class 1 points that are close to the class 1 center, as we have shown in Figure A1.

When $d=2$ as in Figure A1, $f_Z(z) = \frac{1}{2\sigma^2} e^{-\frac{z}{2\sigma^2}}$, which is the exponential distribution with rate $\lambda_1 = \frac{1}{2\sigma^2}$. When $d > 2$, the curvature of the density function is steeper than the exponential distribution, and the conclusion we proved using exponential distribution in Appendix A.1.1 is also valid.

On the other hand, when the center of class 0 is far away from point \mathbf{x} , then in a small neighborhood of \mathbf{x} , the density of any observing class 0 point is low and relatively

flat; therefore, a homogeneous Poisson process assumption is reasonable.

A.1.1 | Proof of Assumption I

Assume that there are $n_1^{(b)}$ class 1 points in $b(\mathbf{x})$, and denote the d th power of their distances to \mathbf{x} as $S_1, \dots, S_{n_1^{(b)}}$. Similarly, assume that there are $n_0^{(b)}$ class 0 points in $b(\mathbf{x})$ and denote the d th power of their distances to \mathbf{x} as $T_1, \dots, T_{n_0^{(b)}}$. Denote $S_{(1)}, \dots, S_{(n_1^{(b)})}$, $T_{(1)}, \dots, T_{(n_0^{(b)})}$ as the corresponding order statistics of $S_1, \dots, S_{n_1^{(b)}}$ and $T_1, \dots, T_{n_0^{(b)}}$, respectively. Define ϕ as the LPP Patterns. Using the results above, we can derive the following:

When $\phi = \{1\}$,

$$\begin{aligned} P(\phi) &= \int_{S_{(1)} < T_{(1)}} dP(S_{(1)}, T_{(1)}) \\ &= \int_0^\infty n_1^{(b)} \lambda_1 e^{-n_1^{(b)} \lambda_1 s} \left(\int_{S_{(1)}}^\infty \lambda_0 e^{-\lambda_0 t} dt \right) ds \\ &= \int_0^\infty n_1^{(b)} \lambda_1 e^{-n_1^{(b)} \lambda_1 s} e^{-\lambda_0 s} ds \\ &= \frac{n_1^{(b)} \lambda_1}{n_1^{(b)} \lambda_1 + \lambda_0}. \end{aligned} \quad (\text{A.1})$$

Similarly, we can derive that when $\phi = \{0\}$,

$$P(\phi) = \frac{\lambda_0}{n_1^{(b)} \lambda_1 + \lambda_0}. \quad (\text{A.2})$$

Now assume ϕ is an LPP Patterns with m_1 1s and m_0 0s in $b(\mathbf{x})$, with its farthest neighbor being in class 1, and $m_1 + m_0 = k$, $m_1 < n_1^{(b)}$, $m_0 < n_0^{(b)}$. $S_{(1)}, S_{(2)}, \dots, S_{(m_1)}$ are the ordered d th power of distance of samples in class 1 to the fixed observation \mathbf{x} . $T_{(1)}, T_{(2)}, \dots, T_{(m_0)}$ are the ordered d th power of distance of samples in class 0 to the fixed observation \mathbf{x} . Then, ϕ corresponds to the set

$$C_\phi = \{S_{(1)} < S_{(2)} < \dots < T_{(1)} < \dots < T_{(m_0)} < S_{(m_1)} < T_{(m_0+1)}\}.$$

Define

$$C_\phi^* = \{S_{(1)} < S_{(2)} < \dots < T_{(1)} < \dots < T_{(m_0)} < S_{(m_1)}\},$$

$$\begin{aligned} P(\phi) &= \int_{C_\phi} dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}, T_{(m_0+1)}) \\ &= \int_{C_\phi^*} P(T_{(m_0+1)} > S_{(m_1)} | S_{(m_1)}, T_{(m_0)}) dP \\ &\quad \times (S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \int_{C_\phi^*} P(T_{(m_0+1)} - T_{(m_0)} > S_{(m_1)} - T_{(m_0)} | T_{(m_0)}, S_{(m_1)}) dP \\ &\quad \times (S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \int_{C_\phi^*} e^{-\lambda_0(S_{(m_1)} - T_{(m_0)})} dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}). \end{aligned}$$

Denote $\{\phi_1\}$ as the LPP patterns that are composed of $\{\phi\}$ followed by a class 1 neighbor and $\{\phi_0\}$ as the LPP patterns that are composed of $\{\phi\}$ followed by a class 0 neighbor. Therefore, $\{\phi_1\}$ corresponds to the set

$$C_{\phi_1} = \{S_{(1)} < S_{(2)} < \dots < T_{(1)} < \dots < T_{(m_0)} < S_{(m_1)} < S_{(m_1+1)} < T_{(m_0+1)}\},$$

and $\{\phi_0\}$ corresponds to the set

$$C_{\phi_0} = \{S_{(1)} < S_{(2)} < \dots < T_{(1)} < \dots < T_{(m_0)} < S_{(m_1)} < T_{(m_0+1)} < S_{(m_1+1)}\}.$$

Define $\Delta S = S_{(m_1+1)} - S_{(m_1)}$, $\Delta T = T_{(m_0+1)} - T_{(m_0)}$,

$$\begin{aligned} P(\phi_1) &= \int_{C_{\phi_1}} dP(S_{(1)}, \dots, S_{(m_1)}, S_{(m_1+1)}, T_{(1)}, \dots, T_{(m_0)}, T_{(m_0+1)}) \\ &= \int_{C_\phi^*} P(T_{(m_0)} < S_{(m_1)} < S_{(m_1+1)} < T_{(m_0+1)} | T_{(m_0)}, S_{(m_1)}) \cdot \\ &\quad dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \int_{C_\phi^*} P(T_{(m_0)} < S_{(m_1)} < \epsilon S + S_{(m_1)} \\ &\quad < \epsilon T + T_{(m_0)} | T_{(m_0)}, S_{(m_1)}) \cdot \\ &\quad dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \int_{C_\phi^*} \left(\int_0^\infty \left(\int_{s+S_{(m_1)}-T_{(m_0)}}^\infty \lambda_0 e^{-\lambda_0 t} dt \right) \right. \\ &\quad \times (n_1^{(b)} - m_1) \lambda_1 e^{-(n_1^{(b)} - m_1) \lambda_1 s} ds \Big) \cdot \\ &\quad dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \end{aligned}$$

$$\begin{aligned} &= \int_{C_\phi^*} \frac{(n_1^{(b)} - m_1) \lambda_1}{(n_1^{(b)} - m_1) \lambda_1 + \lambda_0} e^{-\lambda_0(S_{(m_1)} - T_{(m_0)})} \\ &\quad dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \frac{(n_1^{(b)} - m_1) \lambda_1}{(n_1^{(b)} - m_1) \lambda_1 + \lambda_0} P(\phi). \end{aligned} \quad (\text{A.3})$$

Similarly, we can derive

$$\begin{aligned} P(\phi_0) &= \int_{C_{\phi_0}} dP(S_{(1)}, \dots, S_{(m_1)}, S_{(m_1+1)}, T_{(1)}, \\ &\quad \dots, T_{(m_0)}, T_{(m_0+1)}) \\ &= \int_{C_\phi^*} P(T_{(m_0)} < S_{(m_1)} < T_{(m_0+1)} < S_{(m_1+1)} | T_{(m_0)}, S_{(m_1)}) \cdot \\ &\quad dP(S_{(1)}, \dots, S_{(m_1)}, T_{(1)}, \dots, T_{(m_0)}) \\ &= \frac{\lambda_0}{(n_1^{(b)} - m_1) \lambda_1 + \lambda_{01}} P(\phi). \end{aligned} \quad (\text{A.4})$$

From Equations (A.1), (A.2), (A.3), and (A.4), we can derive the probability of any LPP patterns $\phi = \{i_1, \dots, i_{m_1}, k+1, \dots, k+1\} \in \mathbb{R}^{n_1}$ in rank representation, $m_1 \leq k$, recursively:

$$P(\phi) = \frac{\prod_{l=1}^{m_1} (n_1^{(b)} - l + 1) \lambda_1^{m_1} \lambda_0^{k-m_1}}{\prod_{l=1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l - i_{l-1}} ((n_1^{(b)} - m_1) \lambda_1 + \lambda_0)^{k - i_{m_1}}}, \quad (\text{A.5})$$

where $i_0 = 0$.

Rewrite Equation (A.5) as the following:

$$\begin{aligned} P(\phi) &= \frac{\prod_{l=1}^{m_1} (n_1^{(b)} - l + 1) \lambda_1^{m_1} \lambda_0^{k-m_1} \prod_{l=1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l - i_{l-1}}}{\prod_{l=1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l} ((n_1^{(b)} - m_1) \lambda_1 + \lambda_0)^{k - i_{m_1}}} \\ &= \frac{\prod_{l=1}^{m_1} (n_1^{(b)} - l + 1) \lambda_1^{m_1} \lambda_0^{k-m_1} \prod_{l=0}^{m_1-1} ((n_1^{(b)} - l) \lambda_1 + \lambda_0)}{\prod_{l=1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l} ((n_1^{(b)} - m_1) \lambda_1 + \lambda_0)^{k - i_{m_1}}} \\ &= \frac{\prod_{l=1}^{m_1} (n_1^{(b)} - l + 1) \lambda_1^{m_1} \lambda_0^{k-m_1}}{((n_1^{(b)} - m_1) \lambda_1 + \lambda_0)^k} \\ &\quad \times \prod_{l=1}^{m_1} \left(\frac{(n_1^{(b)} - l) \lambda_1 + \lambda_0}{(n_1^{(b)} - l + 1) \lambda_1 + \lambda_0} \right)^{i_l}. \end{aligned} \quad (\text{A.6})$$

Now, we are ready to demonstrate the following theorem:

Theorem 1. Under this model, for 2 LPP Patterns ϕ_a and ϕ_b ,

1. If they only differ by 1 neighbor, that is, ϕ_a 's j th neighbor is class 1 and ϕ_b 's j th neighbor is class 0. In other words, using LPP patterns rank representation,

$$\phi_a = \{i_1, \dots, i_r, j, i_{r+1}, \dots, i_{m_1}\},$$

$$\phi_b = \{i_1, \dots, i_r, i_{r+1}, \dots, i_{m_1}\},$$

then, when $\lambda_1 > \frac{\lambda_0}{n_1^{(b)} - m_1}$, $\phi_a \triangleright \phi_b$ as defined in Assumption.

2. If their number of neighbors in class 1 are identical (m_1 1's), but we have LPP patterns

$$\phi_a = \{i_1, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_b = \{j_1, \dots, j_{m_1}, k+1, \dots, k+1\},$$

and $i_l \leq j_l$, for $l = 1, \dots, m_1$, with at least 1 strict inequality, then, we also have $\phi_a \triangleright \phi_b$.

We notice that the difference between the d th power of \mathbf{x} 's m_1 th nearest neighbor in class 1 and the d th power of \mathbf{x} 's $(m_1 + 1)$ th nearest neighbor in class 1 follows exponential distribution with intensity rate $(n_1^{(b)} - m_1)\lambda_1$:

$$\Delta S = S_{(m_1+1)} - S_{(m_1)} \sim \exp((n_1^{(b)} - m_1)\lambda_1).$$

The difference between the d th power of \mathbf{x} 's m_0 th nearest neighbor in class 0 and the d th power of \mathbf{x} 's $(m_0 + 1)$ th nearest neighbor in class 0 follows exponential distribution with intensity rate λ_0 :

$$\Delta T = T_{(m_0+1)} - T_{(m_0)} \sim \exp(\lambda_0).$$

The condition $\lambda_1 > \frac{\lambda_0}{n_1^{(b)} - m_1}$ in Theorem means that the distribution of ΔS has a higher intensity rate than that of ΔT .

Proof.

1. If they only differ by 1 neighbor,

$$\phi_a = \{i_1, \dots, i_r, j, i_{r+1}, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_b = \{i_1, \dots, i_r, i_{r+1}, \dots, i_{m_1}, k+1, \dots, k+1\},$$

(ϕ_b changes ϕ_a 's j th neighbor to 0), from Equation (A.5),

$$P(\phi_a) = \frac{\prod_{l=1}^{m_1+1} (n_1^{(b)} - l + 1) \lambda_1^{m_1+1} \lambda_0^{k-m_1-1}}{\prod_{l=1}^r ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l - i_{l-1}} ((n_1^{(b)} - r) \lambda_1 + \lambda_0)^{j - i_r}} \cdot \frac{1}{((n_1^{(b)} - r - 1) \lambda_1 + \lambda_0)^{i_{r+1} - j} \prod_{l=r+2}^{m_1} ((n_1^{(b)} - l) \lambda_1 + \lambda_0)^{i_l - i_{l-1}}} \cdot \frac{1}{((n_1^{(b)} - m_1 - 1) \lambda_1 + \lambda_0)^{k - i_{m_1}}},$$

$$P(\phi_b) = \frac{\prod_{l=1}^{m_1} (n_1^{(b)} - l + 1) \lambda_1^{m_1} \lambda_0^{k-m_1}}{\prod_{l=1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l - i_{l-1}} ((n_1^{(b)} - m_1) \lambda_1 + \lambda_0)^{k - i_{m_1}}}.$$

So

$$\frac{P(\phi_a)}{P(\phi_b)} = \frac{(n_1^{(b)} - m_1) \lambda_1}{\lambda_0} \cdot \left(\frac{1}{\prod_{l=r+2}^{m_1} ((n_1^{(b)} - l) \lambda_1 + \lambda_0)^{i_l - i_{l-1}}} \right) \times \left(\frac{\prod_{l=r+1}^{m_1} ((n_1^{(b)} - l + 1) \lambda_1 + \lambda_0)^{i_l - i_{l-1}}}{((n_1^{(b)} - r) \lambda_1 + \lambda_0)^{j - i_r} ((n_1^{(b)} - r - 1) \lambda_1 + \lambda_0)^{i_{r+1} - j}} \right) \times \left(\frac{(n_1^{(b)} - m_1) \lambda_1 + \lambda_0}{(n_1^{(b)} - m_1 - 1) \lambda_1 + \lambda_0} \right)^{k - i_{m_1}} = \frac{(n_1^{(b)} - m_1) \lambda_1}{\lambda_0} \prod_{l=r+2}^{m_1} \left(\frac{(n_1^{(b)} - l + 1) \lambda_1 + \lambda_0}{(n_1^{(b)} - l) \lambda_1 + \lambda_0} \right)^{i_l - i_{l-1}} \times \left(\frac{(n_1^{(b)} - r) \lambda_1 + \lambda_0}{(n_1^{(b)} - r - 1) \lambda_1 + \lambda_0} \right)^{i_{r+1} - j} \times \left(\frac{(n_1^{(b)} - m_1) \lambda_1 + \lambda_0}{(n_1^{(b)} - m_1 - 1) \lambda_1 + \lambda_0} \right)^{k - i_{m_1}} > \frac{(n_1^{(b)} - m_1) \lambda_1}{\lambda_0}.$$

So when $\lambda_1 > \frac{\lambda_0}{n_1^{(b)} - m_1}$,

$$P(\phi_a) > P(\phi_b).$$

If their number of neighbors in class 1 are identical, and LPP patterns

$$\phi_a = \{i_1, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_b = \{j_1, \dots, j_{m_1}, k+1, \dots, k+1\}.$$

and $i_l \leq j_l$, for $l = 1, \dots, m_1$, then from Equation (A.6),

$$\frac{P(\phi_a)}{P(\phi_b)} = \prod_{l=1}^{m_1} \left(\frac{(n_1^{(b)} - l + 1) \lambda_1 + \lambda_0}{(n_1^{(b)} - l) \lambda_1 + \lambda_0} \right)^{j_l - i_l} > 1.$$

So, we also have

$$P(\phi_a) > P(\phi_b)$$

□

Now, we can prove Assumption under this model:

Proof. For 2 patterns

$$\phi_i = \{i_1, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_j = \{j_1, \dots, j_{m_2}, k+1, \dots, k+1\}, m_1 \geq m_2.$$

Consider a "middle pattern"

$$\phi'_i = \{i_1, \dots, i_{m_2}\}.$$

Therefore, from 1 in Theorem,

$$\phi_i \triangleright \phi'_i,$$

and from 2 in Theorem ,

$$\phi'_i \triangleright \phi_j,$$

which imply

$$\phi_i \triangleright \phi_j.$$

□

Under this model, the probability of observing different LPP patterns with same numbers of class 1 neighbors and class 0 neighbors can be different. In other words, LPP patterns in this case are strictly more informative than k NN counts.

For example, $k = 3$,

$$P(110) = \frac{n_1^{(b)}(n_1^{(b)} - 1)\lambda_1^2\lambda_0}{(\lambda_0 + n_1^{(b)}\lambda_1)(\lambda_0 + (n_1^{(b)} - 1)\lambda_1)(\lambda_0 + (n_1^{(b)} - 2)\lambda_1)}.$$

$$P(011) = \frac{n_1^{(b)}(n_1^{(b)} - 1)\lambda_1^2\lambda_0}{(\lambda_0 + n_1^{(b)}\lambda_1)^2(\lambda_0 + (n_1^{(b)} - 1)\lambda_1)}.$$

$$P(101) = \frac{n_1^{(b)}(n_1^{(b)} - 1)\lambda_1^2\lambda_0}{(\lambda_0 + n_1^{(b)}\lambda_1)(\lambda_0 + (n_1^{(b)} - 1)\lambda_1)^2}.$$

Therefore,

$$P(110) > P(101) > P(011).$$

The position of the neighbor in class 0 plays an informative role in this model.

APPENDIX B: PROPERTY OF LPP _{τ}

In this section, we discuss the property of LPP _{τ} —that it increases when a noisy feature is deleted and decreases when an informative feature is deleted. This property is the basis of the backward feature selection algorithm based on LPP _{τ} in section ??.

Suppose the current feature subspace is $S = \{s_1, \dots, s_d\}$. Under the model in Section ??, suppose in a small neighborhood, the intensity rate of class 1 and 0 are λ_1 and λ_0 , respectively. Denote the current intensity ratio as $v = \frac{\lambda_1}{\lambda_0}$. Denote S^{-i} to be the new feature subspace after deleting feature s_i in S . If feature s_i is informative about the class label, the intensity ratio v will decrease to a new value $v_i < v$ by deleting s_i . On the other hand, if s_i is not informative about the class label, the intensity ratio R will increase to a new value $v_i < v$ by deleting s_i .

If 2 LPP patterns only differ by 1 neighbor,

$$\phi_a = \{i_1, \dots, i_r, j, i_{r+1}, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_b = \{i_1, \dots, i_r, i_{r+1}, \dots, i_{m_1}, k+1, \dots, k+1\}.$$

From Appendix A, we have

$$\begin{aligned} \frac{P(\phi_a)}{P(\phi_b)} &= \frac{(n_1^{(b)} - m_1)\lambda_1}{\lambda_0} \prod_{l=r+2}^{m_1} \left(\frac{(n_1^{(b)} - l + 1)\lambda_1 + \lambda_0}{(n_1^{(b)} - l)\lambda_1 + \lambda_0} \right)^{i_l - i_{l-1}} \\ &\quad \times \left(\frac{(n_1^{(b)} - r)\lambda_1 + \lambda_0}{(n_1^{(b)} - r - 1)\lambda_1 + \lambda_0} \right)^{i_{r+1} - j + 1} \\ &\quad \times \left(\frac{(n_1^{(b)} - m_1)\lambda_1 + \lambda_0}{(n_1^{(b)} - m_1 - 1)\lambda_1 + \lambda_0} \right)^{k - i_{m_1}} \\ &= (n_1^{(b)} - m_1)v \prod_{l=r+2}^{m_1} \left(\frac{(n_1^{(b)} - l + 1)v + 1}{(n_1^{(b)} - l)v + 1} \right)^{i_l - i_{l-1}} \\ &\quad \times \left(\frac{(n_1^{(b)} - r)v + 1}{(n_1^{(b)} - r - 1)v + 1} \right)^{i_{r+1} - j + 1} \left(\frac{(n_1^{(b)} - m_1)v + 1}{(n_1^{(b)} - m_1 - 1)v + 1} \right)^{k - i_{m_1}} \\ &= (n_1^{(b)} - m_1)v \prod_{l=r+2}^{m_1} \left(1 + \frac{1}{(n_1^{(b)} - l) + \frac{1}{v}} \right)^{i_l - i_{l-1}} \\ &\quad \times \left(1 + \frac{1}{(n_1^{(b)} - r - 1) + \frac{1}{v}} \right)^{i_{r+1} - j + 1} \\ &\quad \times \left(1 + \frac{1}{(n_1^{(b)} - m_1 - 1) + \frac{1}{v}} \right)^{k - i_{m_1}}. \end{aligned}$$

which is an increasing function of v .

If 2 LPP patterns have the same number of neighbors in class 1, and LPP patterns

$$\phi_a = \{i_1, \dots, i_{m_1}, k+1, \dots, k+1\},$$

$$\phi_b = \{j_1, \dots, j_{m_1}, k+1, \dots, k+1\},$$

and $i_l \leq j_l$, for $l = 1, \dots, m_1$, with at least 1 strict inequality, then we have

$$\begin{aligned} \frac{P(\phi_a)}{P(\phi_b)} &= \prod_{l=1}^{m_1} \left(\frac{(n_1^{(b)} - l)\lambda_1 + \lambda_0}{(n_1^{(b)} - l + 1)\lambda_1 + \lambda_0} \right)^{i_l - j_l} \\ &= \prod_{l=1}^{m_1} \left(1 + \frac{1}{(n_1^{(b)} - l) + \frac{1}{v}} \right)^{j_l - i_l}, \end{aligned}$$

which is also an increasing function of v .

In both cases, if dimension s_i is informative about the class label, the intensity ratio v will decrease to a new value $v_i < v$ by deleting s_i ; therefore, $\frac{P(\phi_a)}{P(\phi_b)}$ will decrease. On the other hand, if s_i is not informative about the class label, the intensity ratio R will increase to a new value $v_i < v$ by deleting s_i , and $\frac{P(\phi_a)}{P(\phi_b)}$ will increase. The effect of deleting an informative dimension is to reduce the probability ratio between the ordered LPP patterns pair, and the effect of deleting a noninformative dimension is to increase the probability ratio between the ordered LPP patterns pair.

From the formula of LPP _{τ} , if we substitute the observed LPP patterns profile

$$V_1^T = \left(\frac{n_{1,1}}{n_1}, \frac{n_{1,2}}{n_1}, \dots, \frac{n_{1,2^k}}{n_1} \right),$$

$$V_0^T = \left(\frac{n_{0,1}}{n_0}, \frac{n_{0,2}}{n_0}, \dots, \frac{n_{0,2^k}}{n_0} \right),$$

$$V^T = \left(\frac{n_{+1}}{n}, \frac{n_{+2}}{n}, \dots, \frac{n_{+2^k}}{n} \right),$$

with its theoretical counterparts

$$W_1^T = (p_{1,1}, p_{1,2}, \dots, p_{1,2^k}),$$

$$W_0^T = (p_{0,1}, p_{0,2}, \dots, p_{0,2^k}),$$

$$W^T = (p_{+1}, p_{+2}, \dots, p_{+2^k}),$$

and also substitute LPP_τ with its theoretical counterpart:

$$LPP_\tau^* \triangleq \frac{W_1^T C W_0 \sqrt{2p_{1+}p_{0+}}}{\sqrt{W^T D W}}.$$

It is easy to see that increasing the probability ratio between ordered LPP patterns pair will increase the LPP_τ , and vice versa.