2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

A cheaper Rectified-Nearest-Feature-Line-Segment classifier based on safe points

Mauricio Orozco-Alzate
Universidad Nacional de Colombia – Sede Manizales
Departamento de Informática y Computación
km 7 vía al Magdalena, Manizales 170003, Colombia

Manuele Bicego
Università degli Studi di Verona
Dipartimento di Informatica
Strada le Grazie 15, Verona 37134, Italy

Abstract*—The Rectified Nearest Feature Line Segment (RNFLS) classifier is an improved version of the Nearest Feature Line (NFL) classification rule. RNFLS corrects two drawbacks of NFL, namely the interpolation and extrapolation inaccuracies, by applying two consecutive processes —segmentation and rectification— to the initial set of feature lines. The main drawbacks of this technique, occurring in both training and test phases, are the high computational cost of the rectification procedure and the exponential explosion of the number of lines. We propose a cheaper version of RNFLS, based on a characterization of the points that should form good lines. The characterization relies on a recent neighborhood-based principle that categorizes objects into four types: safe, borderline, rare and outliers, depending on the position of each point with respect to the other classes. The proposed approach represents a variant of RNFLS in the sense that it only considers lines between safe points. This allows a drastic reduction in the computational burden imposed by RNFLS. We carried out an empirical and thorough analysis based on different public data sets, showing that our proposed approach, in general, is not significantly different from RNFLS, but cheaper since the consideration of likely irrelevant feature line segments is avoided.

I. INTRODUCTION

The task of automatic classification consists in applying a function f that assigns a class label $\hat{\theta}$ to a test feature vector $\boldsymbol{t} \in \mathbb{R}^D$, where $\hat{\theta} \in \Omega = \{\omega_1, \ldots, \omega_K\}$ in a K-class problem. The function $f : \boldsymbol{t} \to \hat{\theta}$, typically called the *classifier*, is learned from a set of labeled training feature vectors $\mathcal{T} = \{(\boldsymbol{t}_1, \theta_1), \ldots, (\boldsymbol{t}_N, \theta_N)\}$, where $\theta_k \in \Omega$. Such a process of learning f from data is known as *training*, for which it is typically assumed that elements of $\mathcal{T}$ must be normalized by using a technique such as the well-known z-score standardization [1]. The data normalization, prior to training, aims at avoiding the influence of differences in the dynamic ranges of the features and must also be applied to the test feature vector $\boldsymbol{t}$. Hereafter we will assume normalized versions of $\mathcal{T}$ and $\boldsymbol{t}$ and denote them as $\mathcal{X} = \{(\boldsymbol{x}_1, \theta_1), \ldots, (\boldsymbol{x}_N, \theta_N)\}$ and $\boldsymbol{x}$, respectively.

A plethora of classifiers has been proposed in the literature whose decisions are typically based on either posterior probabilities or distances. Among the latter, a commonly used baseline algorithm is the *Nearest Neighbor* (1-NN) rule [2], which assigns to $\boldsymbol{x}$ the class label of its closest $(\boldsymbol{x}_n, y_n) \in \mathcal{X}$. In spite of the simplicity of 1-NN, its classification performance is often satisfactory; besides, it provides a straightforward understanding for non-experts about the motivation of the class label assignment. The first advantage, however, only holds when a sufficiently large training set $\mathcal{X}$ is provided; moreover, due to the local character of the 1-NN decisions, this technique is very sensitive to noise in the data.

Several proposals, aimed at solving the weaknesses of 1-NN, can be found in the literature; some of them are aimed at removing either noisy or superfluous data by applying, for example, condensing and editing methods [3]. Other alternatives are aimed at better exploiting the information from $\mathcal{X}$ by building a continuum between either pairs or triplets of training feature vectors belonging to the same class [4], [5]. The earliest and most popular representative of the latter methods is the *Nearest Feature Line* (NFL) rule [6], in which continua correspond to lines in the feature space: the so-called *feature lines*. NFL labels a test feature vector by assigning it to the class label associated to its nearest feature line. Even though NFL was originally proposed for face recognition, it exhibits a good classification performance in several application scenarios but suffers from three drawbacks: *extrapolation inaccuracy* because feature lines extend indefinitely beyond their endpoints in both directions, *interpolation inaccuracy* due to potential invasion of the interpolating segment of the feature lines —typically for multimodal classes— to the territory of other classes, and *high computational cost*, due to the quadratic time complexity induced by the number of generated feature lines.

The first two above-mentioned drawbacks of NFL —that tend to occur for complicated distributions and in low-dimensional feature spaces— have been overcome by a variant of NFL called the *Rectified Nearest Feature Line Segment* (RNFLS) [7], particularly by applying two consecutive processes to the feature lines, namely a *segmentation* followed by a *rectification*. Less attention has been paid to the third drawback (high computational cost), which still remains an issue for the practical application of the NFL family of classifiers. In particular, this problem occurs in the testing phase since all possible lines should be analyzed, and also in the training phase since the rectification procedure (a rather high cost procedure) is applied to all possible lines. Some strategies have been proposed to reduce the computational cost in the testing case, including selections based on the length of the lines or supervised criteria [8], [9]. In addition, RNFLS

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

itself also alleviates the cost of NFL during the testing stage, since the rectification process reduces the cardinality of the set of feature lines by removing those that cross the territory of other classes. However, in general, how to reduce the number of lines to be considered remains an open issue: in this paper we propose a viable solution to this problem, based on a characterization of points which should form good lines. In this way we move the reasoning from the line-level (which is quadratic in the number of points) to the point level (linear in the number of points).

In our approach we started from the observation that the rectification procedure of RNFLS is indeed very useful, since it generates more concentrated distributions which, thereby, leads to better classification rates – as shown by the proponents in their paper [7]. Then we analyzed the characteristics of the points composing the feature line segments that are preserved after the rectification, together with the points composing the lines which are decisive, i.e. the feature lines that, at the end, are responsible for the class label assignments. We characterized such points according to a neighborhood-based principle recently proposed in [10], [11] that categorizes objects into four types: *safe*, *borderline*, *rare* and *outlier* examples, depending on the position of the point with respect to the other classes. Inspired by a thorough empirical analysis, which showed that in the RNFLS procedure most of the points belonging to the relevant lines (i.e. lines which are decisive for the classification) belong to the *safe* category, here we propose the safeRNFLS, a variant of the RNFLS which only considers lines between *safe* points. This permits a drastic reduction in the computational burden needed by RNFLS. We carried out an empirical and thorough analysis based on different public data sets, showing that our proposed approach, in general, is not significantly different from RNFLS, but cheaper since the consideration of likely irrelevant feature line segments is avoided. A counterexample is also provided, in order to illustrate situations in which the proposed approach is not recommended.

II. NEAREST FEATURE LINE CLASSIFIERS

A. The Nearest Feature Line (NFL) classifier

The NFL classifier [6] was originally proposed for face recognition and, by extension, for pixel-based problems where a relatively high-dimensional representation (either the pixels themselves or a transform of them) is derived. It is defined by considering all possible pairs of training feature vectors —belonging to the same class and without repetition— as endpoints of the interpolating lines that connect the pairs. Such lines, called *feature lines*, are used as a subspace to project a test point onto them. Subsequently, the test point is assigned with the class label of the nearest subspace. More formally, let $\overleftrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j}$ denote a feature line connecting training points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from a particular class, that is $\theta_i = \theta_j$ and $i \neq j$. The orthogonal projection of a test feature vector $\boldsymbol{x}$ onto $\overleftrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j}$ is given by:

$$\boldsymbol{p} = (1-\mu)\boldsymbol{x}_i + \mu\boldsymbol{x}_j, \qquad \mu = \frac{(\boldsymbol{x}-\boldsymbol{x}_i)\cdot(\boldsymbol{x}_j-\boldsymbol{x}_i)}{||\boldsymbol{x}_j-\boldsymbol{x}_i||^2}.$$

Notice that $\boldsymbol{p}$ might end up either on the interpolating or on the extrapolating part of $\overleftrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j}$: the former represents the segment connecting the two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, while the latter corresponds to the rest of the line. The distance from $\boldsymbol{x}$ to $\overleftrightarrow{\boldsymbol{x}_i\boldsymbol{x}_j}$ is then the distance from $\boldsymbol{x}$ to $\boldsymbol{p}$.

B. The Rectified Nearest Feature Line Segment (RNFLS) classifier

As mentioned above, NFL is prone to fail due to a two-fold reason: on the one hand, NFL suffers from the invasion of feature lines to regions belonging to other classes, this being very likely to occur in low-dimensional spaces; on the other hand, NFL is prone to fail because $\boldsymbol{p}$ might be inconveniently located when the projection of $\boldsymbol{x}$ is on the extrapolating part of the feature line. RNFLS [7] is also attractive —as NFL— in small sample size situations but, in order to solve the above-mentioned drawbacks, it differs from NFL in the following three aspects:

- *Segmentation:* Distances from $\boldsymbol{x}$ to $\boldsymbol{p}$ are replaced by the distance from $\boldsymbol{x}$ to the nearest endpoint when $\boldsymbol{p}$ is placed on the extrapolating part of the feature line, therefore, feature lines are now called feature lines segments, denoted hereafter as $\overset{\vdash\!\dashv}{\boldsymbol{x}_i\boldsymbol{x}_j}$. An illustrative example of the segmentation process is shown in Figure 1; different situations are presented for three test feature vectors: $\boldsymbol{x}_a$, $\boldsymbol{x}_b$ and $\boldsymbol{x}_c$, respectively.

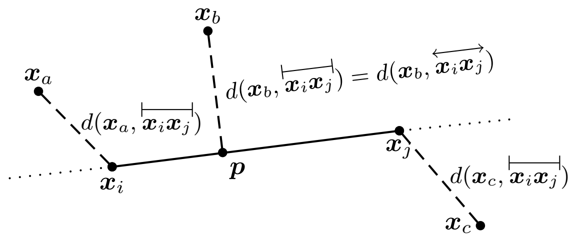


Fig. 1: Illustrative example of the distances from three test feature vectors to a feature line segment.

- *Rectification:* The set of feature line segments is examined in order to remove those that cross regions of other classes. The criterion of invasion is the following: Let r_k be the largest ball, centered at the training feature vector $\boldsymbol{x}_k$ that only includes other training vectors of the same class of $\boldsymbol{x}_k$. Then, $\overset{\vdash\!\dashv}{\boldsymbol{x}_i\boldsymbol{x}_j}$ crosses the territory of $\boldsymbol{x}_k$ if $d(\boldsymbol{x}_k, \overset{\vdash\!\dashv}{\boldsymbol{x}_i\boldsymbol{x}_j}) < r_k$ and provided that $\theta_i \neq \theta_k$. Feature line segments that are preserved after rectification are those that do not cross any territory of the training feature vectors from other classes.
- *Degenerate Lines:* In contrast with NFL, RNFLS also allows degenerated feature lines (cases where $i = j$); that is, feature line segments that are built with a training feature vector and itself. In practice, this means that RNFLS considers not just the distances to the feature line segments but also the distances to the training feature vectors, thus including 1-NN as a special case.

III. PROPOSED ANALYSES AND METHOD

Napierala and Stefanowski [12], [10] recently proposed a typification of examples according to the class labels of their k nearest neighbors. In particular, they proposed to examine $k = 5$ neighbors as an acceptable compromise between a representative size of the neighborhood and its appropriate locality when considering four types of examples. Given a training feature vector $\boldsymbol{x}_i$, let $A \in \{0, \dots, 5\}$ be the number of neighbors that belong to its same class and let $B \in \{0, \dots, 5\}$ the number of its neighbors that belong to different classes. According to the proportion A:B, $\boldsymbol{x}_i$ is typified as follows:

- 5:0 or 4:1 $\Rightarrow$ $\boldsymbol{x}_i$ is considered a safe example.
- 3:2 or 2:3 $\Rightarrow$ $\boldsymbol{x}_i$ is considered a borderline example.
- 1:4 $\Rightarrow$ $\boldsymbol{x}_i$ is considered a rare example but only if its nearest neighbor from the same class has, in turn, an A:B proportion or either 0:5 or 1:4. Otherwise, $\boldsymbol{x}_i$ is considered a borderline example. This definition, that we adopt here, is the one given by Saez et al. [11] —referring to [10]— where a recursive analysis of the case 1:4 is avoided.
- 0:5 $\Rightarrow$ $\boldsymbol{x}_i$ is considered an outlier.

Figure 2 shows an illustrative case of this typification, as applied to an artificial problem with two Gaussian-distributed classes having identity covariance matrices and class means separated by 1 in the first dimension.

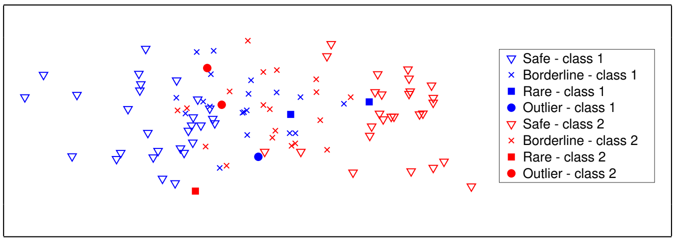


Fig. 2: Illustrative case of typification according to the class labels of the 5 nearest neighbors.

For the sake of simplicity and even though the original neighborhood criterion is based on the so-called heterogeneous value difference metric, we preferred to use the conventional Euclidean distance to judge the proximity between training feature vectors. As a preliminary experiment, we explore the application of this typology as an instance selection procedure for 1-NN, see Sec. IV-B.

Based on the above-described typology, we propose to characterize feature line segments according to the following categories corresponding to all the 2-combinations, with repetition allowed:

s2s:	Feature line segment from safe point to safe point.
s2b:	Feature line segment from safe point to borderline point, or vice versa.
s2r:	Feature line segment from safe point to rare point, or vice versa.
s2o:	Feature line segment from safe point to outlier, or vice versa.
b2b:	Feature line segment from borderline point to borderline point.
b2r:	Feature line segment from borderline point to rare point, or vice versa.
b2o:	Feature line segment from borderline point to outlier, or vice versa.
r2r:	Feature line segment from rare point to rare point.
r2o:	Feature line segment from rare point to outlier, or vice versa.
o2o:	Feature line segment from outlier to outlier.

Starting from this typology, our idea is that the most important lines for NFL (and RNFLS) are those connecting safe points, i.e. only *s2s* lines. These points are placed well inside the classes, so it is highly probable that lines connecting them are not suffering from the problems reported in the previous section (crossing other classes). Our proposal is very straightforward, i.e. firstly selecting only safe points and, afterwards, applying the usual RNFLS procedure. In this sense our proposed method can be understood as a pre-selection of potentially decisive feature line segments followed by the remaining parts of the original RNFLS classifier. The set of pre-selected feature line segments is further refined by the rectification process. In spite of its simplicity, it will be shown in the experimental section that this modified version of RNFLS —that we call heareafter safeRNFLS— is not significantly different from its original version that includes all possible feature line segments. Such a modified version, however, is drastically cheaper since the consideration of likely irrelevant feature line segments is avoided in both training and test phases. The proposed method is schematically depicted in Fig. 3.

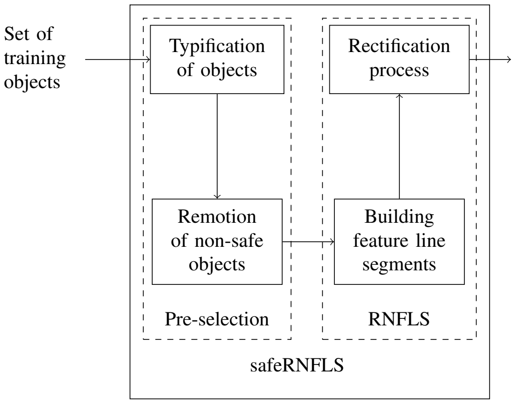


Fig. 3: A block diagram for the training phase of the proposed safeRNFLS method.

IV. RESULTS AND DISCUSSION

In this section the proposed method is evaluated. In particular, we first start by analyzing in which categories fall the most important feature line segments used by RNFLS, in terms of both lines maintained after the rectification procedure and lines used for the final classification (i.e. the nearest feature line segments). This analysis, which justifies our proposal, also provides us with the opportunity of better understanding the local and global effect of the rectification process as well as knowing the typical nature of the feature line segments that, at the end, are the responsible ones for the class label assignments. These analyses are discussed below, followed by a performance evaluation of the proposed safeRNFLS classifier in Sec. IV-D. As a preliminary analysis, we reported the performances of 1-NN when we only select safe points in the training set, which corresponds to seeing the typification introduced in the previous section as a condensing strategy.

A. Experimental setup and data sets

All the reported classification accuracies for the experiments were estimated for a repeated 50%-50% train and test protocol with 20 repetitions. In order to compare the results and decide whether differences are significant, we also computed the corresponding standard errors [13]. It is important, however, to remember in this comparison that there are no unbiased estimators for the variance and, thereby, neither for the standard errors [14]. Mean and standard deviation for the z-score normalization were computed by using just the training set and, afterwards, applying the corresponding normalization to both training and test sets.

For the sake of reproducible research, experiments were performed for almost the same collection of data sets from the UCI Machine Learning repository that were originally used by Du and Chen [7], except that we did not consider Housing data set because its conversion from a regression to a classification problem was unclear to us. In addition, taking into account the well-known “no free luch” theorem [15] which establishes that there is no a universally good learning algorithm, we also report classification accuracies with a data set (called Gastro) for which our proposal does not behave well. This data set is also publicly available at the UCI repository [16] and contains 698-dimensional feature vectors extracted from 76 colonoscopy two-channel videos. We only considered data from the narrow band imaging channel.

Feature vectors having missing values were removed from the data sets. Basic details of the data sets, along with their corresponding leave-one-out (LOO) 1-NN accuracies and composition of types of examples, are given in Table I. It is worth mentioning that, for WPBC, we did not get the same LOO 1-NN accuracy that was reported in [7].

Many observations can be derived by looking at Table I. Firstly, notice that WDBC, Wine and Iris are the most homogeneous data sets, comprising 90% safe examples. These data sets therefore correspond to easy problems as can be verified by their highly accurate LOO 1-NN results which are superior to 90% accuracies. A bit less homogeneous data set is Hepatitis, which also accounts for an LOO 1-NN accuracy greater than 90%. In a descending order of composition complexity, we find Ionosphere data set. Afterwards, we can identify a group of three data sets —Pima, WPBC and Glass— that are composed of approximately 50% safe examples and about 40% borderline ones. The LOO 1-NN accuracy for these data sets is about 70%. Finally, the most heterogeneous and challenging data sets are Liver and Gastro, which are dominated by borderline examples and account for the lowest LOO 1-NN accuracies among the considered data sets. These variety of data sets allows us to perform a comprehensive study under several data complexity scenarios.

B. Typification as instance selection method for 1-NN

The neighborhood-based typology of feature vectors can be used, in a straightforward way, as an instance selection method for the 1-NN classifier. Consider, for example, that we keep safe feature vectors from the training set and remove all the others from it. Hereafter we call the 1-NN rule, preceded by this instance selection method, as *safeNN*. In Table II(a) we compare the accuracies of 1-NN vs. safeNN.

It is interesting to note that for two of the easiest data sets —Iris and Wine— as well as for the most complicated ones (Liver and Gastro), accuracy differences are minimal (this meaning that, by removing non safe points, we observe neither improvements nor deteriorations of the classification performance). For the other data sets, results are either in favor of 1-NN (Hepatitis, Ionosphere and Glass) or in favor of safeNN (Pima, WDBC and WPBC). The most significant performance differences are observed for the two most heterogeneous data sets —WPBC and Glass— since, in their cases, the selection of safe examples reduces the training sets in about a half of its original cardinality. This is somehow expected; however, the fact to be highlighted is that the largest improvement is observed for an heterogeneous, 32-dimension two-class data set (WPBC) while the largest performance deterioration occurs for an heterogeneous, 9-dimensional six-class data set (Glass).

According to these results, it is not certain whether removing the non-safe examples rather leads to an improved or at least not worsened version of 1-NN.

C. Analysis of the types of the feature line segments

In this section we report the characterization of the feature line segments which are more relevant for the RNFLS classifier. To do that, we first performed a classification, following our proposed line of characterization, of the feature line segments before as well as after applying the rectification process, to see which is the most common type of lines kept after the rectification. Such analysis is shown in Fig. 4.

Notice that, for all the data sets except Liver (and Gastro, whose figure is not shown due to space constraints), the most preserved category after the rectification process is represented by the *s2s* lines. Other combinations for safe examples — namely *s2b*, *s2r* and *s2o*— consistently exhibit the opposite behavior: more segments are removed than selected. The only exception for this occurs for Ionosphere, for wich about a half of *s2b* and *s2r* feature line segments are either selected or removed, respectively. In contrast, most of the *b2b* feature line segments are removed by the rectification process, except in the case of Ionosphere, for which 60% of them, as well as the same proportion of *b2r* ones, are maintained. That is a curious fact since Ionosphere is not dominated by these type of examples; in fact, it contains only a 11.4% of

TABLE I: Description of the data sets and their corresponding LOO 1-NN accuracies and composition per type.

					[*Safe*, *Borderline*, *Rare*, *Outlier*]	
Dataset	#Vectors	#Feat.	#Classes	LOO 1-NN acc.	Vectors per type	Percent (%) per type
`Hepatitis`	80	19	2	92.50%	[71, 7, 0, 2]	[88.75, 8.75, 0.0, 2.5]
`Iris`	150	4	3	94.67%	[137, 11, 0, 2]	[91.33, 7.33, 0.0, 1.33]
`Pima`	768	8	2	70.57%	[429, 282, 26, 31]	[55.86, 36.72, 3.39, 4.04]
`Wine`	178	13	3	95.51%	[163, 13, 0, 2]	[91.57, 7.3, 0.0, 1.12]
`Liver`	345	6	2	63.19%	[112, 224, 5, 4]	[32.46, 64.93, 1.45, 1.16]
`Ionosphere`	351	34	2	86.32%	[275, 40, 19, 17]	[78.35, 11.4, 5.41, 4.84]
`WDBC`	569	30	2	95.08%	[537, 25, 3, 4]	[94.38, 4.39, 0.53, 0.7]
`WPBC`	194	32	2	73.20%	[93, 82, 7, 12]	[47.94, 42.27, 3.61, 6.19]
`Glass`	214	9	6	70.09%	[103, 81, 13, 17]	[48.13, 37.85, 6.07, 7.94]
`Gastro`	76	698	3	59.21%	[19, 54, 0, 3]	[25.0, 71.05, 0.0, 3.95]

TABLE II: Classification accuracies, and their corresponding standard errors, estimated for 20 repetitions of 50-50 random training-test for: (a) 1-NN vs. safeNN, (b) RNFLS vs. safeRNFLS, (c) NFL vs. safeNFL.

	(a) 1-NN vs. safeNN		(b) RNFLS vs. safeRNFLS		(c) NFL vs. safeNFL	
Dataset	1-NN	safeNN	RNFLS	safeRNFLS	NFL	safeNFL
`Hepatitis`	92.25±0.95	87.75±1.16	91.50±0.99	91.38±0.99	93.62±0.86	93.00±0.90
`Iris`	93.40±0.64	93.67±0.63	94.87±0.57	94.80±0.57	87.07±0.87	87.47±0.85
`Pima`	70.29±0.52	72.93±0.51	74.14±0.50	74.44±0.50	68.05±0.53	68.31±0.53
`Wine`	94.33±0.55	94.27±0.55	95.45±0.49	95.34±0.50	95.73±0.48	95.62±0.49
`Liver`	59.83±0.83	58.32±0.84	63.67±0.82	62.86±0.82	61.16±0.83	61.04±0.83
`Ionosphere`	84.49±0.61	77.50±0.70	90.43±0.50	89.38±0.52	83.89±0.62	83.38±0.63
`WDBC`	94.88±0.29	95.61±0.27	96.47±0.24	96.53±0.24	94.77±0.29	94.88±0.29
`WPBC`	65.46±1.08	75.36±0.98	72.99±1.01	74.33±0.99	72.16±1.02	71.75±1.02
`Glass`	66.40±1.02	58.41±1.07	68.36±1.01	67.10±1.02	62.90±1.04	60.28±1.06
`Gastro`	52.11±1.81	49.08±1.81	55.66±1.8	45.53±1.81	58.55±1.79	51.97±1.81

borderline feature vectors, cf. Table I. The other combination with borderline examples (*b2o*) are mostly removed but in a more conservative way for Ionosphere: 67% of the feature line segments of this type. Notice also that Iris, Hepatitis and Wine have a minimal number of rare examples in our 50%-50% train and test protocol (or none when considering the entire design set, cf. Table I) and, consequently, all feature line segments involving them are either absent or not significant in comparison with the other types. For other data sets, however, *r2r* and *r2o* feature line segments are not consistently removed; see for instance the case of *r2r* feature line segments for Ionosphere and WDBC. Regarding the *o2o* feature line segments, most of them are kept after rectification for Hepatitis, Iris, Wine, Glass and WDBC while, in contrast, they are mostly removed for the other data sets. Such a remotion is of about a half of them (53%) for Liver.

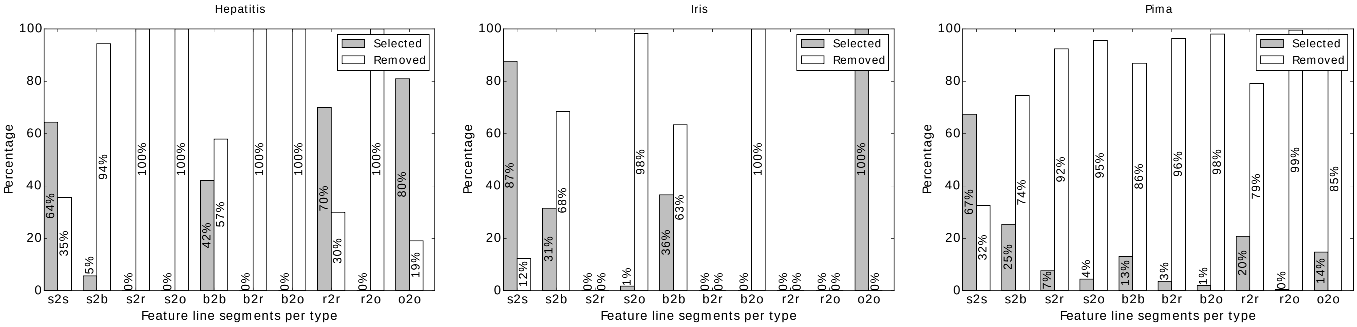
Summarizing, from these experiments it seems evident that the most important feature lines for the rectification procedure (i.e. those which are kept after rectification) are composed by safe points, thus confirming our intuition and proposal.

A complementary analysis which can be done is to investigate the type of the feature line segments that are responsible for the classification, i.e. the feature line segments which assign the class label to the test feature vectors. For the same train and test experiment with 20 repetitions, we inspected all these lines, characterizing them to the type of the feature line segments; see Fig. 5. From there we can see that, for all the considered data sets except for Liver (and Gastro, whose figure is not shown due to space constraints), most of the class labels are assigned by *s2s* feature line segments. This is of course more evident for the four most homogeneous data sets, namely: Hepatitis, Iris, Wine and WDBC. Again, the group of similar data sets composed by Pima, WPBC and Glass present a comparable distributions of class assignments per type. A particular mention is due to Ionosphere, whose class labels are mainly assigned by *s2s* segments in spite that the rectification process is not particularly strong in the remotion of non-safe examples (cf. Fig. 4(f)), probably because this data set has many dimensions where feature lines are less likely to cross each other. Liver, whose composition is dominated by borderline examples, is the only one that exhibits large bars in the number of class assignments for *s2b* and *b2b* feature line segments.

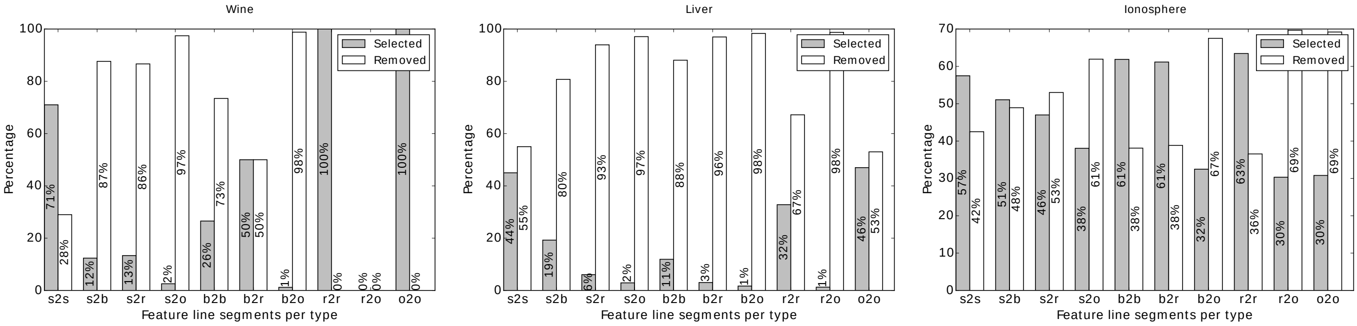
The above-discussed analyses confirm and motivate our proposal, i.e. that the removal of all non-safe examples, prior the building of the feature line segments, would allow us to avoid computations without significantly deteriorating the classification performance of the original RNFLS. In the subsequent section we quantify both issues: (i) the percentage of saved computations in terms of the ratio between *s2s* feature line segments and the total number of them and (ii) the classification accuracies of RNFLS and safeRNFLS and the comparisons between them.

D. Performances of the safeRNFLS classifier

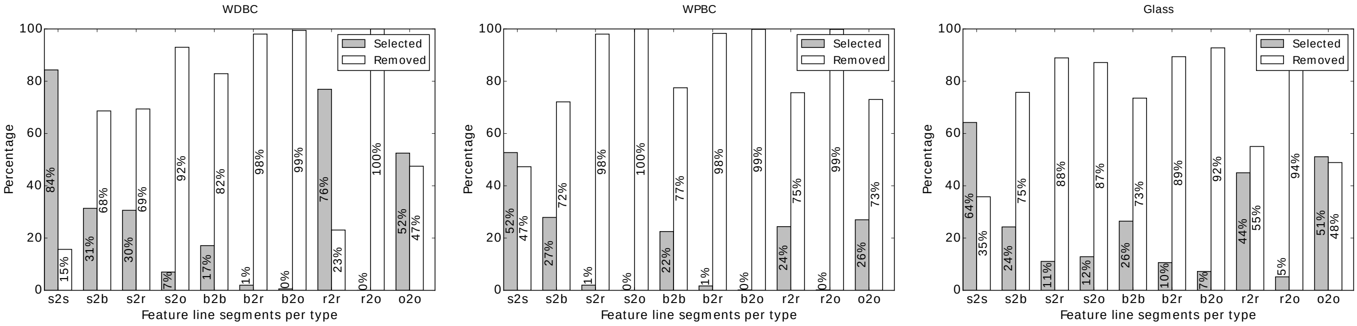
We investigate the first issue (saved computations) by measuring the percentage of *s2s* feature line segments with respect to the total amount of them. Knowing this proportion gives us an idea of the scale of saved computations if only *s2s* feature line segments are taken into account. These percentages are reported in Table III. Notice that the saving of computations after rectification ranges from 2.25% for Hepatitis up to



(a) Hepatitis (b) Iris (c) Pima



(d) Wine (e) Liver (f) Ionosphere



(g) WDBC (h) WPBC (i) Glass

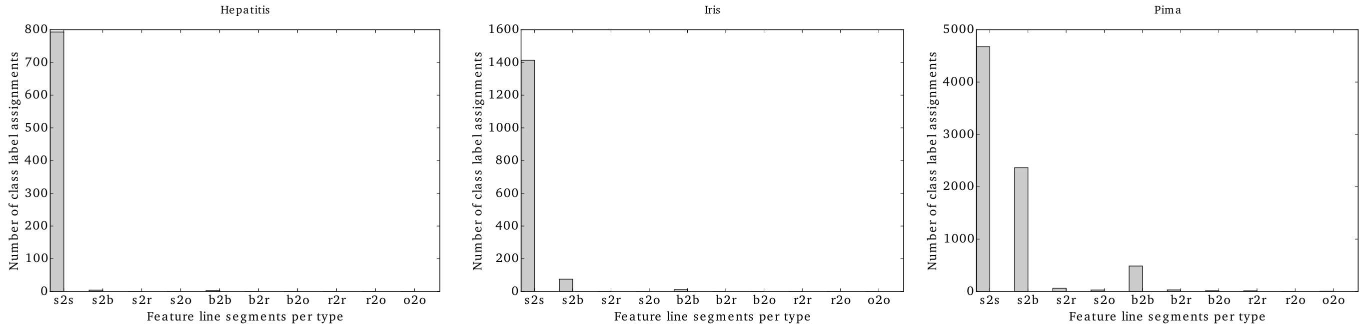
Fig. 4: Percentage of feature line segments per type that, after the rectification process, were either selected or removed.

TABLE III: Percentage of saved computations before and after rectification.

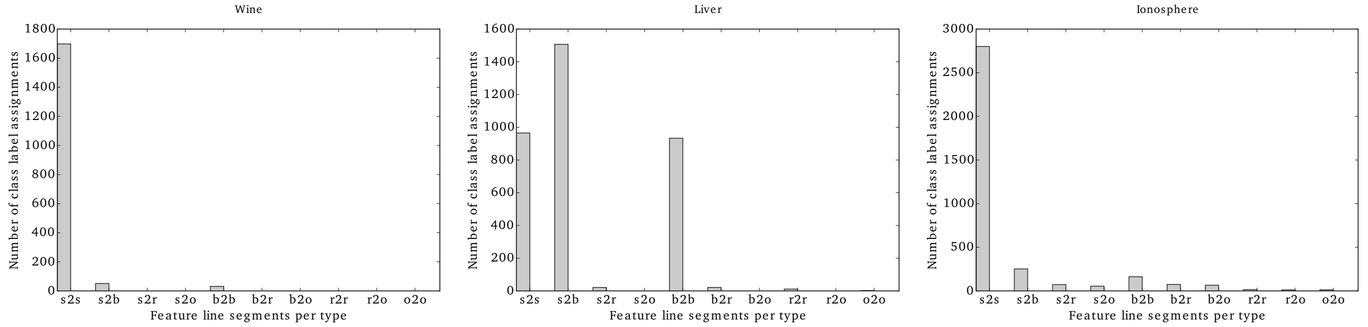
	Before rectification		After rectification	
Dataset	$^{s2s}/_{Total}$	Saving	$^{s2s}/_{Total}$	Saving
`Hepatitis`	$^{10836}/_{11918}$	9.08%	$^{6979}/_{7140}$	2.25%
`Iris`	$^{15517}/_{19643}$	21.00%	$^{13604}/_{14895}$	8.67%
`Pima`	$^{288981}/_{808094}$	64.24%	$^{194865}/_{293991}$	33.72%
`Wine`	$^{22536}/_{28472}$	20.85%	$^{16004}/_{16819}$	4.85%
`Liver`	$^{16359}/_{153369}$	89.33%	$^{7360}/_{26679}$	72.41%
`Ionosphere`	$^{129289}/_{168564}$	23.30%	$^{74332}/_{92985}$	20.06%
`WDBC`	$^{372720}/_{431193}$	13.56%	$^{314294}/_{331124}$	5.08%
`WPBC`	$^{23761}/_{60812}$	60.93%	$^{12525}/_{21417}$	41.52%
`Glass`	$^{8078}/_{31652}$	74.48%	$^{5187}/_{10500}$	50.60%
`Gastro`	$^{300}/_{5040}$	94.05%	$^{300}/_{1780}$	83.15%

83.15% for Gastro. Pima —the largest data set— gets a saving percentage of 33.72% that, even though is not the largest percentage, it does correspond to one of the largest absolute savings of computation time. Even the modest percentage of saving for WDBC is significant, considering that it is the second largest data set.

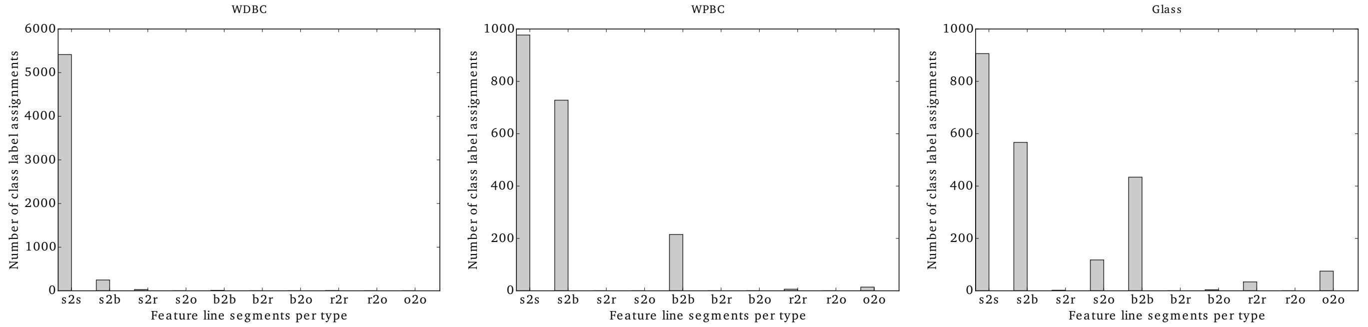
The second aspect (classification accuracies of the safeRNFLS) has been examined by performing a classification test, using exactly the same partitions for 20 repetitions, and comparing the results per data set. Classification accuracies are shown in Table II(b). Notice that there are no highlighted results except for WPBC and Gastro; that means, for most of the considered data sets, our proposal is in practice equally accurate to the original RNFLS classifier. In the particular case of WPBC, safeRNFLS is in fact significantly better than RNFLS. In contrast, for Gastro, safeRNFLS is significantly worse than RNFLS; this meaning that our proposal is not recommended for very complicated data sets (i.e. few safe examples) lying in very sparse high-dimensional feature spaces.



(a) Hepatitis (b) Iris (c) Pima



(d) Wine (e) Liver (f) Ionosphere



(g) WDBC (h) WPBC (i) Glass

Fig. 5: Number of class label assignments according to the type of feature line segments

E. Performance of the safeNFL classifier

As a last experiment, we reported an analysis aimed at studying the effect of removing all non-safe examples before using the original NFL classifier. As explained in Sec. II-A, NFL suffers from two drawbacks that limit its applicability; however, our aim in this last experiment is confirming whether NFL still does almost the same job when building feature lines only between safe points than when using all the available examples. The comparison is shown in Table II(c).

We can see that the performance deterioration for safeNFL against NFL is only significant for Glass and Gastro. For all the remaining data sets, NFL and safeNFL are equivalent but the amount of avoided computations is remarkable in all cases, cf. columns under the "Before rectification" heading in Table III.

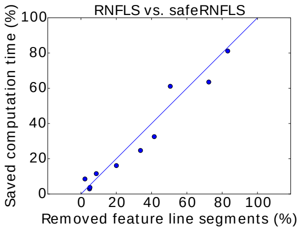
F. Comparison of execution times

The reduction in computational burden —that was reported in Table III as the proportion of feature line segments that are removed from consideration— does not necessarily translate into the number of processor operations or time saved. Therefore, we measured actual timing results when evaluating a test partition in each data set, for 1-NN, RNFLS, NFL and their corresponding "safe" variants; see Table IV. All the experiments were performed on a HP laptop with the following specifications: AMD A9-9420 processor, 3GHz, 8GB RAM and Windows 10. The `time.perf_counter()` function from Python 3.4.1 was used to measure the elapsed times. There were no other applications running on the machine during the experiment.

It can be noticed that the percentages of saved computation time cover a wide range depending on the amount of retained

TABLE IV: Timing results in seconds, and their corresponding percentage of saved computation time, for the evaluation of a test set: (a) 1-NN vs. safeNN, (b) RNFLS vs. safeRNFLS, (c) NFL vs. safeNFL.

	(a) 1-NN vs. safeNN			(b) RNFLS vs. safeRNFLS			(c) NFL vs. safeNFL		
Dataset	1-NN	safeNN	Saving	RNFLS	safeRNFLS	Saving	NFL	safeNFL	Saving
Hepatitis	0.03	0.02	7.27%	0.44	0.40	8.53%	0.49	0.46	6.78%
Iris	0.14	0.10	26.20%	1.56	1.38	11.56%	1.41	1.26	10.56%
Pima	2.03	0.91	55.11%	127.74	96.19	24.70%	334.62	134.60	59.78%
Wine	0.11	0.08	21.17%	1.74	1.69	2.90%	2.59	2.08	19.85%
Liver	0.34	0.11	69.49%	4.71	1.72	63.54%	28.85	2.82	90.22%
Ionosphere	0.35	0.27	21.35%	17.62	14.77	16.12%	33.67	27.08	19.59%
WDBC	0.91	0.87	4.21%	101.81	98.03	3.71%	135.29	117.02	13.51%
WPBC	0.11	0.06	45.63%	2.64	1.78	32.54%	6.24	1.21	80.53%
Glass	0.13	0.05	61.60%	1.08	0.42	61.12%	3.55	0.88	75.31%
Gastro	0.02	0.005	78.90%	0.06	0.01	81.14%	0.34	0.004	98.89%



safe objects. In spite of the individual differences that we observe between the percentages reported in Tables III and Table IV, the relation between the corresponding ones is almost linear, as shown in the scatter plot on the right of Table IV for the case of RNFLS vs. safeRNFLS. Notice also in the figure that, in only three cases (Hepatitis, Iris and Glass), the percentage of saved computation time is slightly larger than the percentage of saved computations. This behavior can be explained by the fact that these data sets —excluding Gastro, which is very high-dimensional in contrast with them— are those with the lower number of of feature line segments after rectification, cf. Table III.

V. CONCLUSION

The RNFLS classifier effectively improves over NFL by solving two important drawbacks of the latter. The cardinality of the set of feature line segments after rectification is, however, still large and time consuming for both training and test phases. In this paper, we proposed a novel variant of RNFLS, called here safeRNFLS, that only considers feature line segments connecting safe points, according to an adopted neighborhood-based definition of safeness. We have experimentally showed that safe-to-safe feature line segments are typically the ones providing the class label assignments for the RNFLS classifier and, therefore, the important ones to be built while avoiding the construction of the other ones. We provided empirical evidence confirming that safeRNFLS is not, in general, significantly different from RNFLS but cheaper to be trained and used once deployed. We also observed that our proposal is not recommended in cases of complicated compositions along with very sparse representations (few examples in very high-dimensional feature spaces). Future work may include the consideration of scores for the degree of safeness, such that a greater control on the size of the set of the feature line segments is obtained.

ACKNOWLEDGMENT

This work was supported by Universidad Nacional de Colombia - Sede Manizales [project Hermes 38489]. We thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., ser. The Morgan Kaufmann Series in Data Management Systems. Boston, USA: Morgan Kaufmann, 2012, ch. 3: Data Preprocessing, pp. 83 – 124.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[3] S. García, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, mar 2012.

[4] J.-T. Chien and C.-C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1644–1649, 2002.

[5] M. Orozco-Alzate and C. G. Castellanos-Domínguez, "Comparison of the nearest feature classifiers for face recognition," *Machine Vision and Applications*, vol. 17, no. 5, pp. 279–285, oct 2006.

[6] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439–443, 1999.

[7] H. Du and Y. Q. Chen, "Rectified nearest feature line segment for pattern classification," *Pattern Recognition*, vol. 40, no. 5, pp. 1486 – 1497, 2007.

[8] Y. Plasencia-Calaña, M. Orozco-Alzate, E. García-Reyes, and R. P. W. Duin, "Selecting feature lines in generalized dissimilarity representations for pattern recognition," *Digital Signal Processing*, vol. 23, no. 3, pp. 902–911, may 2013.

[9] K. Kamaei, "Editing the Nearest Feature Line Classifier," Master's thesis, Eastern Mediterranean University, North Cyprus, 2013.

[10] K. Napierala and J. Stefanowski, "Types of minority class examples and their influence on learning classifiers from imbalanced data," *Journal of Intelligent Information Systems*, vol. 46, no. 3, pp. 563–597, 2016.

[11] J. A. Sáez, B. Krawczyk, and M. Woźniak, "Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets," *Pattern Recognition*, vol. 57, pp. 164 – 178, 2016.

[12] K. Napierala and J. Stefanowski, "Identification of different types of minority class examples in imbalanced data," in *Hybrid Artificial Intelligent Systems: 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part II*, ser. LNCS, E. Corchado, V. Snášel, A. Abraham, M. Woźniak, M. Graña, and S.-B. Cho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7209, pp. 139–150.

[13] M. Bramer, *Principles of Data Mining*, 2nd ed., ser. Undergraduate Topics in Computer Science. London, UK: Springer, 2013, ch. 7: Estimating the Predictive Accuracy of a Classifier, pp. 79–92.

[14] Y. Bengio and Y. Grandvalet, "No unbiased estimator of the variance of k-fold cross-validation," *Journal of Machine Learning Research*, vol. 5, pp. 1089–1105, 2004.

[15] D.H. Wolpert, "The Lack of A Priori Distinctions Between Learning Algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[16] P. Mesejo et al., "Computer-Aided Classification of Gastrointestinal Lesions in Regular Colonoscopy," *IEEE Transactions on Medical Imaging*, vol. 35, no. 9, pp. 2051–2063, 2016. Data set at: https://archive.ics.uci.edu/ml/datasets/Gastrointestinal+Lesions+in+Regular+Colonoscopy