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

Mauricio Orozco-Alzate[1] and Manuele Bicego[2]

[1]Universidad Nacional de Colombia - Sede Manizales, Colombia

[2]University of Verona, Italy

25th International Conference on Pattern Recognition

January 10-15, 2021

Milan, Italy

Introduction

- A commonly used baseline algorithm is the well-known *Nearest Neighbor* (1-NN) rule.
- There are several proposals aimed at solving the weaknesses of 1-NN, e.g. condensing and editing (García et al., 2012).
- Other alternatives build continua between pairs of points belonging to the same class (Chien and Wu, 2002).
- The earliest and most popular representative of the latter methods is the *Nearest Feature Line* (NFL) rule (Li and Lu, 1999), in which continua correspond to lines in the feature space: the so-called *feature lines*.

Introduction

- A commonly used baseline algorithm is the well-known *Nearest Neighbor* (1-NN) rule.
- There are several proposals aimed at solving the weaknesses of 1-NN, e.g. condensing and editing (García et al., 2012).
- Other alternatives build continua between pairs of points belonging to the same class (Chien and Wu, 2002).
- The earliest and most popular representative of the latter methods is the *Nearest Feature Line* (NFL) rule (Li and Lu, 1999), in which continua correspond to lines in the feature space: the so-called *feature lines*.

Introduction

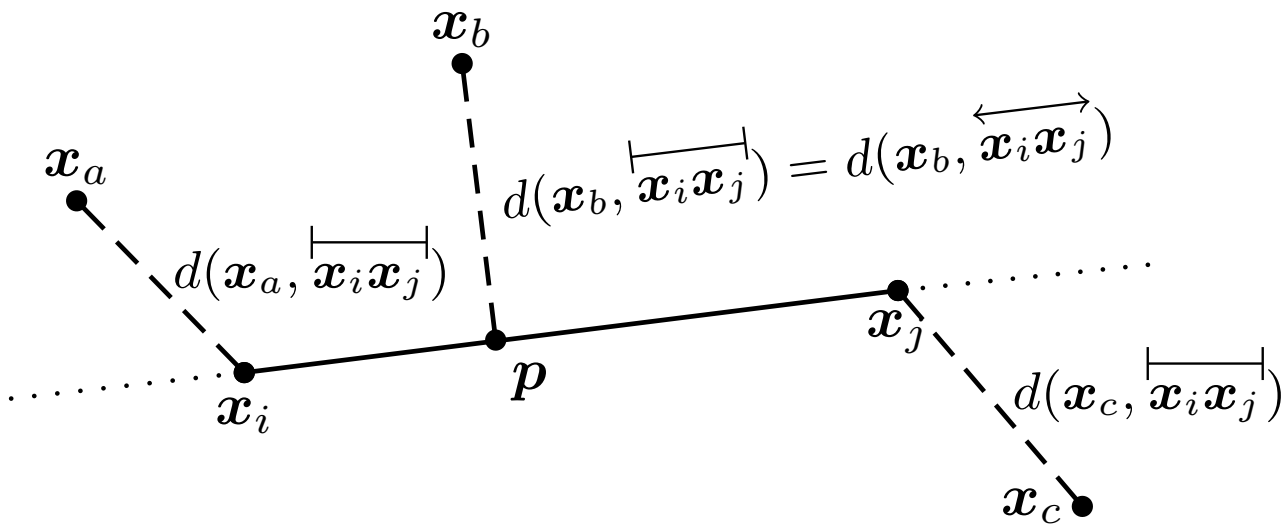
- A commonly used baseline algorithm is the well-known *Nearest Neighbor* (1-NN) rule.
- There are several proposals aimed at solving the weaknesses of 1-NN, e.g. condensing and editing (García et al., 2012).
- Other alternatives build continua between pairs of points belonging to the same class (Chien and Wu, 2002).
- The earliest and most popular representative of the latter methods is the *Nearest Feature Line* (NFL) rule (Li and Lu, 1999), in which continua correspond to lines in the feature space: the so-called *feature lines*.

Introduction

- A commonly used baseline algorithm is the well-known *Nearest Neighbor* (1-NN) rule.
- There are several proposals aimed at solving the weaknesses of 1-NN, e.g. condensing and editing (García et al., 2012).
- Other alternatives build continua between pairs of points belonging to the same class (Chien and Wu, 2002).
- The earliest and most popular representative of the latter methods is the *Nearest Feature Line* (NFL) rule (Li and Lu, 1999), in which continua correspond to lines in the feature space: the so-called *feature lines*.

The RNFLS classifier

The Rectified-Nearest-Feature-Line Segment (RNFLS) classifier (Du and Chen, 2007) improves over NFL (Li and Lu, 1999) by solving two drawbacks of the latter: *interpolation* and *extrapolation* inaccuracies.



The RNFLS classifier

Two processes allow to solve those drawbacks:

- **Segmentation:** Distances on the extrapolating part of the feature line are replaced with the distance to the nearest endpoint.
- **Rectification:** Feature lines segments crossing the territory of other classes are removed. Very costly!
- Degenerated lines are also considered. So, RNFLS includes 1-NN as a special case.

The RNFLS classifier

Two processes allow to solve those drawbacks:

- **Segmentation:** Distances on the extrapolating part of the feature line are replaced with the distance to the nearest endpoint.
- **Rectification:** Feature lines segments crossing the territory of other classes are removed. Very costly!
- Degenerated lines are also considered. So, RNFLS includes 1-NN as a special case.

The RNFLS classifier

Two processes allow to solve those drawbacks:

- **Segmentation:** Distances on the extrapolating part of the feature line are replaced with the distance to the nearest endpoint.
- **Rectification:** Feature lines segments crossing the territory of other classes are removed. Very costly!
- Degenerated lines are also considered. So, RNFLS includes 1-NN as a special case.

The RNFLS classifier

Two processes allow to solve those drawbacks:

- **Segmentation:** Distances on the extrapolating part of the feature line are replaced with the distance to the nearest endpoint.
- **Rectification:** Feature lines segments crossing the territory of other classes are removed. Very costly!
- Degenerated lines are also considered. So, RNFLS includes 1-NN as a special case.

Typification according to the 5-nearest neighbors

According to the proportion *{Same class}:{Different class}* among its 5-nearest neighbors, a point **x** is categorized as (Napierala and Stefanowski, 2016):

- *s* (safe) if 5:0 or 4:1;
- *b* (borderline) if 3:2 or 2:3;
- *r* (rare) if 1:4 but, only if its nearest neighbor from the same class has, in turn, a proportion or either 0:5 or 1:4. Otherwise, **x** is *b* (Sáez et al., 2016);
- *o* (outlier) if 0:5.

Typification according to the 5-nearest neighbors

According to the proportion *{Same class}:{Different class}* among its 5-nearest neighbors, a point **x** is categorized as (Napierala and Stefanowski, 2016):

- *s* (safe) if 5:0 or 4:1;
- *b* (borderline) if 3:2 or 2:3;
- *r* (rare) if 1:4 but, only if its nearest neighbor from the same class has, in turn, a proportion or either 0:5 or 1:4. Otherwise, **x** is *b* (Sáez et al., 2016);
- *o* (outlier) if 0:5.

Typification according to the 5-nearest neighbors

According to the proportion *{Same class}:{Different class}* among its 5-nearest neighbors, a point **x** is categorized as (Napierala and Stefanowski, 2016):

- *s* (safe) if 5:0 or 4:1;
- *b* (borderline) if 3:2 or 2:3;
- *r* (rare) if 1:4 but, only if its nearest neighbor from the same class has, in turn, a proportion or either 0:5 or 1:4. Otherwise, **x** is *b* (Sáez et al., 2016);
- *o* (outlier) if 0:5.

Typification according to the 5-nearest neighbors

According to the proportion *{Same class}:{Different class}* among its 5-nearest neighbors, a point **x** is categorized as (Napierala and Stefanowski, 2016):

- *s* (safe) if 5:0 or 4:1;
- *b* (borderline) if 3:2 or 2:3;
- *r* (rare) if 1:4 but, only if its nearest neighbor from the same class has, in turn, a proportion or either 0:5 or 1:4. Otherwise, **x** is *b* (Sáez et al., 2016);
- *o* (outlier) if 0:5.

Typification according to the 5-nearest neighbors

According to the proportion *{Same class}:{Different class}* among its 5-nearest neighbors, a point **x** is categorized as (Napierala and Stefanowski, 2016):

- *s* (safe) if 5:0 or 4:1;
- *b* (borderline) if 3:2 or 2:3;
- *r* (rare) if 1:4 but, only if its nearest neighbor from the same class has, in turn, a proportion or either 0:5 or 1:4. Otherwise, **x** is *b* (Sáez et al., 2016);
- *o* (outlier) if 0:5.

Proposed typification of feature line segments

- We propose to categorize each feature line segment according to the types of its endpoints: *s2s*, *s2b*, *s2r*, *s2o*, *b2b*, *b2r*, *b2o*, *r2r*, *r2o*, *o2o*.
- The most preserved category after the rectification process is *s2s*. In addition, most of the class labels are assigned by them.
- Hypothesis: removing of all non-safe examples, prior the building of the feature line segments, allows to avoid many computations without significantly deteriorating the classification performance of the original RNFLS.

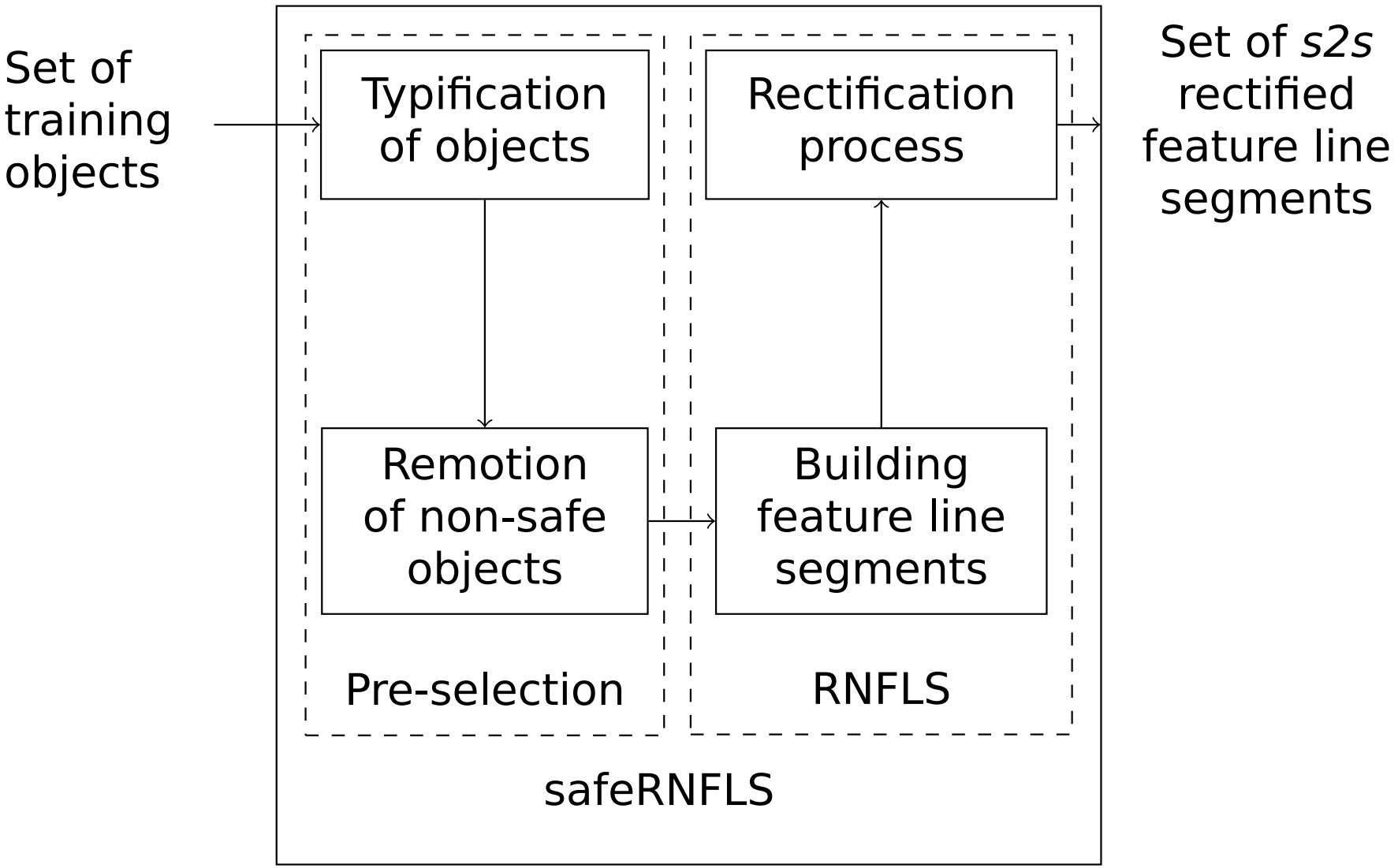
Proposed typification of feature line segments

- We propose to categorize each feature line segment according to the types of its endpoints: *s2s*, *s2b*, *s2r*, *s2o*, *b2b*, *b2r*, *b2o*, *r2r*, *r2o*, *o2o*.
- The most preserved category after the rectification process is *s2s*. In addition, most of the class labels are assigned by them.
- Hypothesis: removing of all non-safe examples, prior the building of the feature line segments, allows to avoid many computations without significantly deteriorating the classification performance of the original RNFLS.

Proposed typification of feature line segments

- We propose to categorize each feature line segment according to the types of its endpoints: *s2s*, *s2b*, *s2r*, *s2o*, *b2b*, *b2r*, *b2o*, *r2r*, *r2o*, *o2o*.
- The most preserved category after the rectification process is *s2s*. In addition, most of the class labels are assigned by them.
- Hypothesis: removing of all non-safe examples, prior the building of the feature line segments, allows to avoid many computations without significantly deteriorating the classification performance of the original RNFLS.

safeRNFLS: the cheaper proposal



Classification accuracies

- Setup: 20 repetitions, 50-50 random training-test.
- Safe variants of 1-NN and NFL were also studied: safeNN and safeNFL, respectively.

	(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

- safeRNFLS is, in general, not significantly different from RNFLS.

Execution times (in seconds) and percentage of savings

- Setup: HP laptop: AMD A9-9420 proc., 3GHz, 8GB RAM, Windows 10, timing with the `time.perf_counter()` Python 3.4.1 function.

	(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%

- safeRNFLS is, in general, much cheaper than RNFLS.

Conclusions

- *s2s* feature line segments are typically the ones providing the class label assignments for the RNFLS classifier (see the paper).
- safeRNFLS is, in general, not significantly different from RNFLS but much cheaper (saved computations and excecution times, in most cases, are outstanding).
- safeRNFLS is not recommended for complicated compositions along with very sparse representations (few examples in very high-dimensional feature spaces); c.f. `Gastro`.

Contact information

Thanks for your attention.

Contact information:

- Mauricio Orozco-Alzate
  Universidad Nacional de Colombia
  Sede Manizales, Colombia
  E-mail: morozcoa@unal.edu.co
- Manuele Bicego
  University of Verona, Italy
  E-mail: manuele.bicego@univr.it

References

Chien, J.-T. and Wu, C.-C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1644–1649.

Du, H. and Chen, Y. Q. (2007). Rectified nearest feature line segment for pattern classification. *Pattern Recognition*, 40(5):1486 – 1497.

García, S., Derrac, J., Cano, J. R., and Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435.

Li, S. Z. and Lu, J. (1999). Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks*, 10(2):439–443.

Napierala, K. and Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers

from imbalanced data. *Journal of Intelligent Information Systems*, 46(3):563–597.

Sáez, J. A., Krawczyk, B., and Woźniak, M. (2016). Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164 – 178.