Rodrigo Capobianco Guido

Paraconsistent Feature Engineering

Today's modern world is filled with uncertainties and contradictions. As artificial intelligence (AI) advances, machines are frequently expected to mimic the human brain and, consequently, face the conflicts associated with this task. To overcome them, feature engineering has emerged as the field of science responsible for turning raw data into relevant input information, setting up classifiers in the fused digital signal processing (DSP) and pattern recognition (PR) domain. Despite the ongoing efforts to improve feature learning, handcrafted extraction still plays a very important role. In this context, a careful choice of features is extremely relevant for creating an accurate classification. This article sheds light on the problem of feature quality by using a nonclassical logical system capable of handling conflictive situations. It is known as *paraconsistent logic* (*PL*).

Relevance

More often than not, AI algorithms specifically dedicated to PR have dominated DSP systems, stimulating further studies on feature engineering [1]. Whenever classic logic fails to address an issue in this field, PL [2] may be a solution. Therefore, this study, which is complemented by a numerical example, is of paramount importance.

Prerequisites

Very basic notions of classic logic and PR are desirable but not imperative. Readers who are unfamiliar with the topics discussed in this article may want to consult the literature referenced in [3] and [4] before proceeding any further. Basic comments about PL, which comprises the focus of this article, are provided in future sections and can also be drawn from [2], [5], and [6].

Problem statement and solution

Problem statement

Consider an N-class classification problem for which the classes $\{C_1, C_2, \ldots, C_N\}$ are represented by a certain number of T-sample long feature vectors, all of which are obtained based on a handcrafted extraction. The system engineer, who intuitively selects the features, needs a quantitative evaluation on their suitability for either supervised or unsupervised learning [4]. In other words, the question is: "Are those features convenient to classify my data?"

Essentially, the answer not only depends on the features themselves but also on the technique adopted to analyze them. Typically, some features yield poor categorization when associated with a certain classifier, while being excellent whenever used in conjunction with another. A modest strategy, such as an ordinary distance metric, requires exceptionally prepared features to treat a real-world problem successfully. On the contrary, a deep neural network is possibly capable of solving difficult tasks if it receives only a set of modest features as input.

Let us consider the case in which a weak classifier is able to generate accurate results for a specific task using a given set of features. A better classifier, therefore, is guaranteed to produce good results using the same set of features. Thus, selecting the best features based on a modest method, which unavoidably works as a simple classifier, allows for generalization, i.e., the features will efficiently address the problem in conjunction with basic or sophisticated classifiers. For this reason, elementary techniques are adopted here.

As the PR community knows, favorable feature vectors exhibit considerable similarity when extracted from inputs of a particular class and a notable distinction when coming from different classes. Furthermore, whenever the features substantially contribute to solve a problem, they avoid the simple forwarding of a nonpolished issue to the classifier, which is the next stage of the classification system. As a result, the problem is that of defining a quantitative strategy for investigating the extracted features, observing that independent intraclass and interclass analyses may cause conflicts.

Digital Object Identifier 10.1109/MSP.2018.2874549
Date of publication: 24 December 2018

1053-5888/19©2019IEEE

Solution

As most of the classifiers, the technique adopted to solve the stated problem also requires all of the feature vectors to first be normalized within the $0 \sim 1$ range, which allows for a proper scale of inspection. There are different ways to do this; for example, to analyze the behavior of a particular physical entity at the time intervals it is observed, we frequently register its percentual distribution, not its values, as in methods A_1, A_2, B_1, and B_2 from [7] and [8]. Contrarily, proper records of magnitudes related to a maximum amount contain the unity as the largest possible value, as in methods A_3 and B_3, also defined in [7] and [8]. Hypothetical examples of the former and latter cases are the feature vectors {0.28, 0.25, 0.24, 0.23}, for which the components add up to 1, and {0.82, 0.86, 0.89, 1}, where 1 is the baseline, respectively. Occasionally, measuring the amplitudes related to a predefined independent value also makes sense, as in the feature vector {0.80, 0.30, 0.98, 0.04}, for which 1 neither appears nor consists of the additive sum of its elements. Therefore, normalization as well as the choice of features are important preprocessing steps that help to conveniently represent the physical entity of interest, depending on the specific problem assessed. Works in [7]–[9] contain various illustrative examples and allow for practical hands-on experience in such a task, which is at the discretion of the system engineer.

Once the normalization is completed, we are ready to study the feature vectors adequately. As I mentioned, this problem is solved by using two independent criteria: one to quantify the intraclass similarities and another to reflect the interclass dissimilarities. Each is represented by the quantities α, which expresses the level of faith in the features, and β, which specifies their level of discredit, respectively, where $(0 \leq \alpha, \beta \leq 1)$. Independence indicates that α and β are not complementary, i.e., $\alpha + \beta$ might be different from the unity, implying that ordinary logic [3] is not the proper tool for treating this problem.

As a basis for performing the intraclass analysis, we note that the amplitude, or range, of a set of K real numbers, i.e., $s[\cdot] = \{s_0, s_1, \ldots, s_{K-1}\}$, is the simplest way to measure its deviation [10]. It is defined as $A = L(s[\cdot]) - S(s[\cdot])$, i.e., the difference between the largest and the smallest values in $s[\cdot]$. Notably, the lesser A is, the closer the scalars in the set are, and vice versa. Since $(0 \leq A \leq 1)$ due to its previous normalizations, $Y = (1 - A)$ can be used as a standardized measure of similarity among the scalars in such a way that $Y \approx 0$ and $Y \approx 1$ indicate low and high similarities, respectively.

In this article, we are interested in finding the similarity among feature vectors of size *T*, not scalars. Consequently, we can perform, separately for each class, an element-wise similarity-vector computation where the *i*th element represents the similarity among the corresponding components of the feature vectors, as shown in Figure 1. Hereafter, the similarity vectors are intuitively named as $\text{svC}_1[\cdot], \text{svC}_2[\cdot], \ldots, \text{svC}_N[\cdot]$. Their corresponding arithmetic means, i.e.,

$$\bar{Y}(C_1) = \frac{1}{T}\sum_{i=0}^{T-1} \text{svC}_1[i],$$

$$\bar{Y}(C_2) = \frac{1}{T}\sum_{i=0}^{T-1} \text{svC}_2[i],$$

$$\ldots$$

$$\bar{Y}(C_N) = \frac{1}{T}\sum_{i=0}^{T-1} \text{svC}_\text{N}[i],$$

which are used to balance their respective individual values, correspond to the intraclass similarities. Ideally, all of them would be close to 1. Thus, to assess the worst case, we define α as the smallest among the intraclass similarities, i.e., $\alpha = \min\{\bar{Y}(C_1), \bar{Y}(C_2), \ldots, \bar{Y}(C_N)\}$. Once the calculations to find α are completed, the next step is to define β as follows.

To perform the interclass analysis, as shown in Figure 1, we initially compute two range vectors of size T for each class: one with element-wise minimum values of the whole set and the other with element-wise maximum values of the whole set. Thus, range vectors store the exact interval containing all the feature vector values from their respective classes. Just to clarify, if a certain class is composed of the feature vectors {0.5, 0.4}, {0.6, 0.1} and {0.3, 0.2}, then, the corresponding range vectors for the smallest and largest values are {0.3, 0.1} and {0.6, 0.4}, respectively.

We then perform an element-wise comparison between the feature vectors from each class and the range vectors from all of the other classes to define *R*, i.e., the number of feature vector elements, if any, that overlap the respective range. Each overlapped element means that a feature from one class invaded the range used by the features from another class, which is disadvantageous for the classification, i.e., a straight line is not capable of separating between those interclass features, and a nonlinear technique is likely necessary. Finally, we define

$$\beta = \frac{R}{F},$$

where F is the maximum possible number of overlaps. Assuming that each one of the N classes contains *X* feature vectors of size *T*, then $F = N \cdot (N-1) \cdot X \cdot T$. Notably, we consider one comparison as a look at a particular position of both range vectors of a class.

The quantities α and β create two distinct, and possibly conflictive, measures. At this point, it is important to note that $\alpha = 1$ strongly suggests that the intraclass feature vectors are similar and represent their respective classes precisely. Complementarily, $\beta = 0$ suggests that the interclass feature vectors do not overlap, thus reassuring our faith in them. Both of these conditions are expected, at least approximately, for an accurate and easy-to-perform classification.

Despite this best case, there are three other extreme possibilities involving those measures, i.e., $(\alpha, \beta) = (0, 1)$, $(\alpha, \beta) = (0, 0)$, and $(\alpha, \beta) = (1, 1)$, in addition to an infinite number of intermediary values where $(0 < \alpha, \beta < 1)$. To shed some light on this scenario, we adopt a simple yet flexible tool used to treat conflictive information as potentially informative: PL. Described in [2], [5], and [6], it uses α and β to calculate the degree of certainty, i.e., $G_1 = \alpha - \beta$, and the degree of contradiction, i.e., $G_2 = \alpha + \beta - 1$, of a statement,

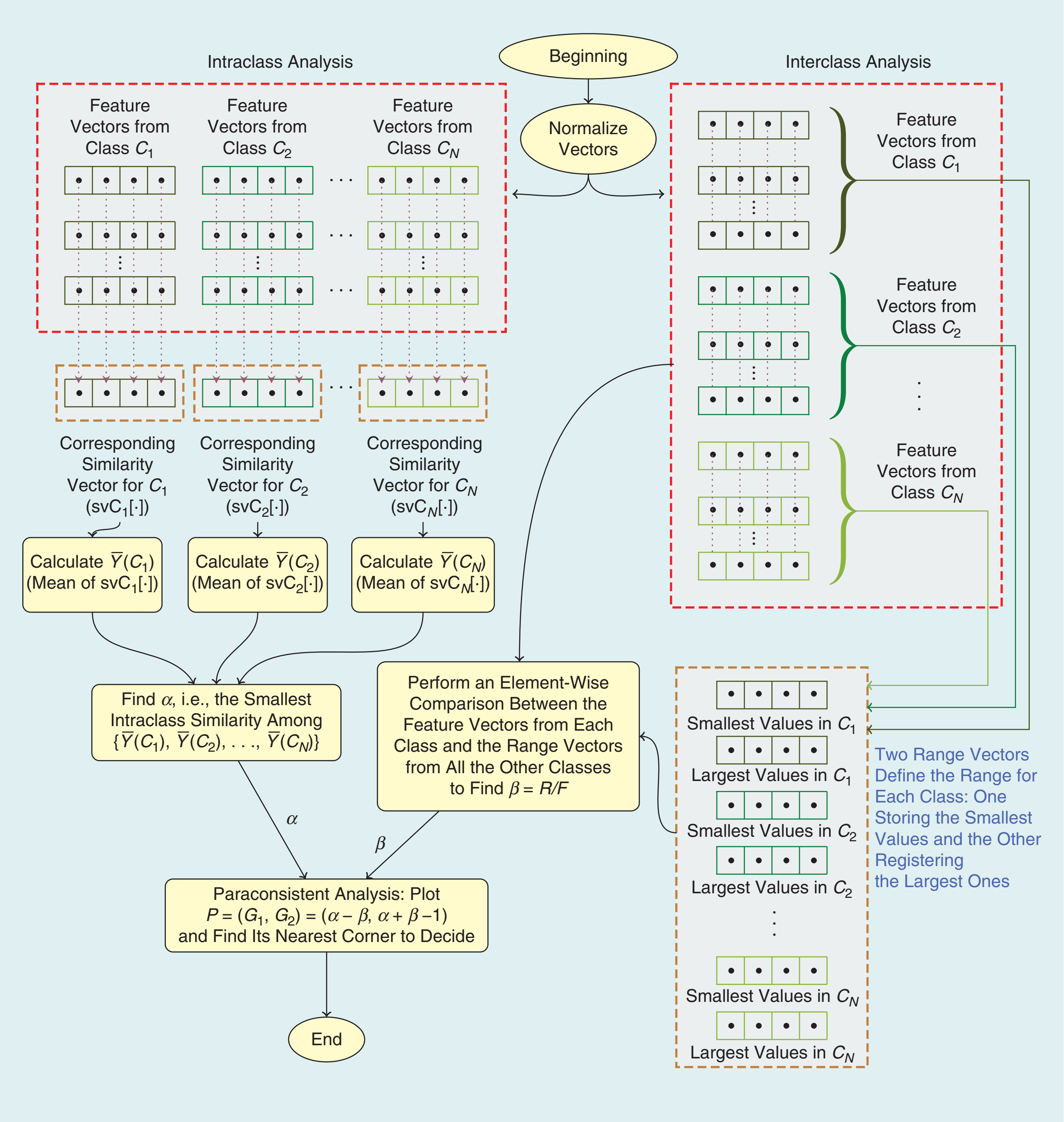
Beginning
Normalize Vectors
Intraclass Analysis
Feature Vectors from Class C_1
Feature Vectors from Class C_2
Feature Vectors from Class C_N
Corresponding Similarity Vector for C_1 (svC$_1$[·])
Corresponding Similarity Vector for C_2 (svC$_2$[·])
Corresponding Similarity Vector for C_N (svC$_N$[·])
Calculate $\overline{Y}(C_1)$ (Mean of svC$_1$[·])
Calculate $\overline{Y}(C_2)$ (Mean of svC$_2$[·])
Calculate $\overline{Y}(C_N)$ (Mean of svC$_N$[·])
Find α, i.e., the Smallest Intraclass Similarity Among $\{\overline{Y}(C_1), \overline{Y}(C_2), \ldots, \overline{Y}(C_N)\}$
Interclass Analysis
Feature Vectors from Class C_1
Feature Vectors from Class C_2
Feature Vectors from Class C_N
Perform an Element-Wise Comparison Between the Feature Vectors from Each Class and the Range Vectors from All the Other Classes to Find $\beta = R/F$
Smallest Values in C_1
Largest Values in C_1
Smallest Values in C_2
Largest Values in C_2
Smallest Values in C_N
Largest Values in C_N
Two Range Vectors Define the Range for Each Class: One Storing the Smallest Values and the Other Registering the Largest Ones
α
β
Paraconsistent Analysis: Plot $P = (G_1, G_2) = (\alpha - \beta, \alpha + \beta - 1)$ and Find Its Nearest Corner to Decide
End

FIGURE 1. The proposed approach based on PL: (upper left) intraclass analysis, (right) interclass analysis, and (lower left) paraconsistent analysis.

where $(-1 \le G_1, G_2 \le 1)$, as shown in Figure 1.

On one hand, certainty varies from falsehood to truth, i.e., $G_1 = -1$ and $G_1 = 1$, respectively. On the other hand, contradiction varies from indefinition to ambiguity, i.e., $G_2 = -1$ and $G_2 = 1$, respectively. Figure 2, which contains additional explanations, shows the paraconsistent plane where the point $P = (G_1, G_2)$ is plotted to allow for the intended analysis. Particularly, the distances from P to the corners $(-1, 0)$, $(1, 0)$, $(0, -1)$ and $(0, 1)$, i.e., $\sqrt{(G_1+1)^2+(G_2)^2}$, $\sqrt{(G_1-1)^2+(G_2)^2}$, $\sqrt{(G_1)^2+(G_2+1)^2}$, and $\sqrt{(G_1)^2+(G_2-1)^2}$, respectively, reveal the following conclusion: favorable feature vectors place P closer to the corner $(1, 0)$ than to the other corners.

Numerical example

Problem statement

To illustrate the proposed approach numerically, we assume a hypothetical three-class classification problem in which the four bidimensional normalized feature vectors in each class are such that:

- class C_1: $\{0.90, 0.12\}$, $\{0.88, 0.14\}$, $\{0.88, 0.13\}$, and $\{0.89, 0.11\}$

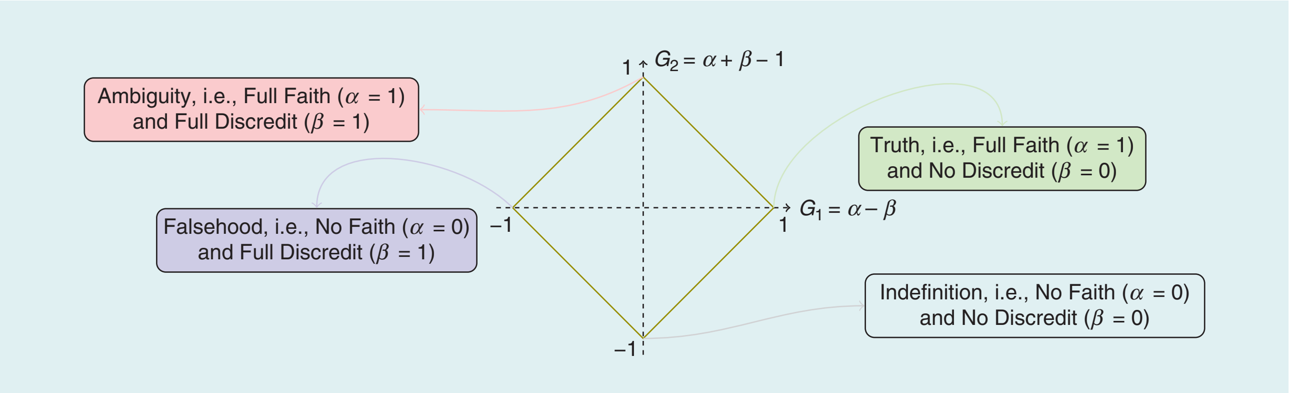
$G_2 = \alpha + \beta - 1$
1
Ambiguity, i.e., Full Faith (α = 1) and Full Discredit (β = 1)
Truth, i.e., Full Faith (α = 1) and No Discredit (β = 0)
Falsehood, i.e., No Faith (α = 0) and Full Discredit (β = 1)
–1
$G_1 = \alpha - \beta$
1
Indefinition, i.e., No Faith (α = 0) and No Discredit (β = 0)
–1

FIGURE 2. The paraconsistent plane. Axes G_1 and G_2 represent the degrees of certainty and contradiction, respectively. As shown, $(G_1, G_2) = (-1, 0)$, $(G_1, G_2) = (1, 0)$, $(G_1, G_2) = (0, -1)$, and $(G_1, G_2) = (0, 1)$ represent falsehood, truth, indefinition, and ambiguity, respectively. In the context of the proposed technique, as $P = (G_1, G_2)$ approaches these corners, we see that "a strong classifier is likely to be required because intraclass feature vectors are notably scattered and interclass feature vectors significantly overlap"; "a weak classifier is likely to solve the problem because intraclass feature vectors are consistently grouped together and interclass feature vectors minimally overlap"; "the features are likely to cause an indefinition, i.e., both intraclass and interclass feature vectors are much different," and "the features are likely to cause an ambiguity, i.e., both intraclass and interclass feature vectors are considerably similar," respectively. Overall, the shorter the path from $P = (G_1, G_2)$ to the corner (1, 0) is, the better the feature vectors are, and the weaker the classifier used in conjunction with them can be. Additionally, if P approaches the corner (1, 0) by the fourth quadrant, i.e., the lower-right triangle that forms the paraconsistent plane, a linear classifier solves the problem because, in that region, $\beta = 0$, i.e., no interclass overlap exists.

- class C_2: $\{0.55, 0.53\}$, $\{0.53, 0.55\}$, $\{0.54, 0.54\}$, and $\{0.56, 0.54\}$
- class C_3: $\{0.10, 0.88\}$, $\{0.11, 0.86\}$, $\{0.12, 0.87\}$, and $\{0.11, 0.88\}$.

The problem is to determine how adequate these feature vectors are to classify the corresponding raw input data.

Solution

To perform the intraclass analysis, we compute the amplitude vectors as follows:

- class C_1: $\{0.90 - 0.88, 0.14 - 0.11\} = \{0.02, 0.03\}$
- class C_2: $\{0.56 - 0.53, 0.55 - 0.53\} = \{0.03, 0.02\}$
- class C_3: $\{0.12 - 0.10, 0.88 - 0.86\} = \{0.02, 0.02\}$.

Thus, the corresponding similarity vectors are

- class C_1: $\{1 - 0.02, 1 - 0.03\} = \{0.98, 0.97\}$
- class C_2: $\{1 - 0.03, 1 - 0.02\} = \{0.97, 0.98\}$
- class C_3: $\{1 - 0.02, 1 - 0.02\} = \{0.98, 0.98\}$.

and the respective means are

$$\bar{Y}(C_1)\text{: } \frac{0.98 + 0.97}{2} = 0.975$$

$$\bar{Y}(C_2)\text{: } \frac{0.97 + 0.98}{2} = 0.975$$

$$\bar{Y}(C_3)\text{: } \frac{0.98 + 0.98}{2} = 0.980.$$

Consequently, $\alpha = \min\{0.975, 0.975, 0.980\} = 0.975$.

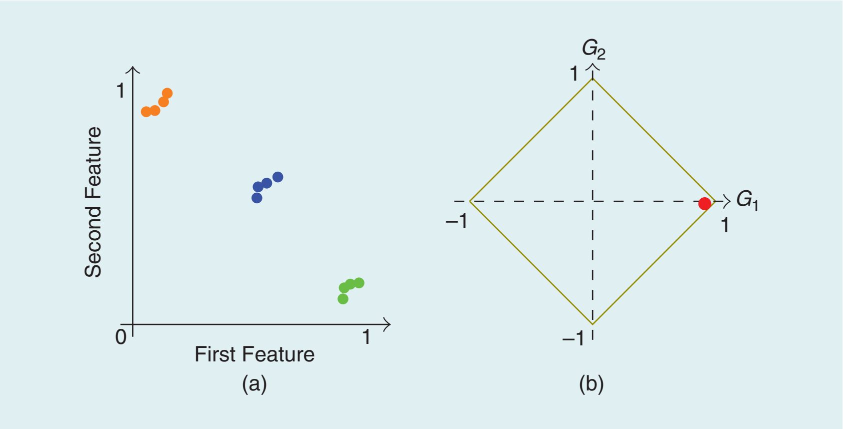

FIGURE 3. (a) The feature vectors from classes C_1, C_2, and C_3 in green, blue, and orange, respectively, localized in the space (clearly, they are linearly separable). (b) The point P is represented as a red dot in the paraconsistent plane for the numerical example.

To perform the interclass analysis, we initially compute the range vectors as follows:

- class C_1: $\{0.88, 0.11\}$, containing the smallest components, and $\{0.90, 0.14\}$ with the largest elements
- class C_2: $\{0.53, 0.53\}$, containing the smallest components, and $\{0.56, 0.55\}$ with the largest elements
- class C_3: $\{0.10, 0.86\}$, containing the smallest components, and $\{0.12, 0.88\}$ with the largest elements.

Then, comparing each feature vector component from one class to the range vector components from all of the other classes, we detect no overlap, i.e., $R = 0$, among $F = 3 \cdot (3 - 1) \cdot 4 \cdot 2 = 48$ possible overlaps, as shown in Figure 3(a). Thus, $\beta = R/F = 0/48 = 0$.

Proceeding with the paraconsistent analysis, we compute $G_1 = \alpha - \beta = 0.975 - 0 = 0.975$ and $G_2 = \alpha + \beta - 1 = 0.975 + 0 - 1 = -0.025$. Plotting $P = (G_1, G_2)$, as shown in Figure 3(b), and calculating its distance (d) to the four corners of the paraconsistent plane, we have

$$\begin{aligned} &d((G1, G2), (-1, 0)) \\ &\quad = \sqrt{(0.975 + 1)^2 + (-0.025)^2} \\ &\quad = 1.975, \end{aligned}$$

$$
\begin{aligned}
d((G1,G2),(1,0)) &= \sqrt{(0.975-1)^2+(-0.025)^2}\\
&= 0.035,
\end{aligned}
$$

$$
\begin{aligned}
d((G1,G2),(0,-1)) &= \sqrt{(0.975)^2+(-0.025+1)^2}\\
&= 1.415
\end{aligned}
$$

and

$$
\begin{aligned}
d((G1,G2),(0,1)) &= \sqrt{(0.975)^2+(-0.025-1)^2}\\
&= 1.416.
\end{aligned}
$$

Therefore, since the smallest among the distances places P closer to (1, 0) than to the other corners and P is in the fourth quadrant, the features are linearly separable, thus providing an accurate classification based on a modest strategy. Although an advanced classifier may also be used to correctly interpret the input data, it is not required in this case.

What we have learned

Based on the information in this article, the reader may effectively use PL to overcome conflictive information and investigate how adequate a set of features is to classify data in an N-class problem. The proposed solution provides highly generalizable results and disregards the specific classifier that will be used in conjunction with these features.

Author

Rodrigo Capobianco Guido (guido@ieee.org) received his B.Sc. degree in 1998, his M.Sc. degree in 2000, his Ph.D. degree in 2003, and his L.D. degree in 2008. He is an associate professor at São Paulo State University (UNESP) in José do Rio Preto, Brazil, and is a Senior Member of the IEEE. Contact him for more information on how to obtain the C++ source code necessary to implement the technique described in this article.

References

[1] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. Birmingham, UK: O'Reilly Media, 2018.

[2] W. Carnielli and M. E. Coniglio, *Paraconsistent Logic: Consistency, Contradiction and Negation*. New York: Springer-Verlag, 2016.

[3] R. M. Smullyan, *A Beginner's Guide to Mathematical Logic*. New York: Dover, 2014.

[4] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer-Verlag, 2018.

[5] F. R. Carvalho and J. M. Abe, *A Paraconsistent Decision-Making Method*. New York: Springer-Verlag, 2018.

[6] J. M. Abe, *Paraconsistent Intelligent-Based Systems: New Trends in the Applications of Paraconsistency*. New York: Springer-Verlag, 2015.

[7] R. C. Guido, "A tutorial on signal energy and its applications," *Neurocomput.*, vol. 179, pp. 264–282, Feb. 2016.

[8] R. C. Guido, "ZCR-aided neurocomputing: A study with applications," *Knowledge-Based Syst.*, vol. 105, pp. 248–269, Aug. 2016.

[9] R. C. Guido, "A tutorial-review on entropy-based handcrafted feature extraction for information fusion," *Inform. Fusion*, vol. 41, pp. 161–175, May 2018.

[10] P. Bruce and A. Bruce, *Practical Statistics for Data Scientists: 50 Essential Concepts*. Sebastopol, CA: O'Reilly Media, 2017.

SP

Digital Object Identifier 10.1109/MSP.2018.2876521

Correction

The "Lecture Notes" article that was published in the January 2019 issue of *IEEE Signal Processing Magazine* [1] contains an error in the last line of the caption of Figure 2, page 157: the word *because* should be *whenever*.

The correct caption of Figure 2 is as follows:

Additionally, if P approaches the corner (1, 0) by the fourth quadrant, i.e., the lower-right triangle that forms the paraconsistent plane, a linear classifier solves the problem whenever, in that region, $\beta = 0$, i.e., no interclass overlap exists.

Subsequently, the text on page 158, left column, fourth row after the equations, is amended as follows:

Therefore, since the smallest among the distances places P closer to (1, 0) than to the other corners and P is in the fourth quadrant with $\beta = 0$, the features are linearly separable, thus providing an accurate classification based on a modest strategy.

Reference

[1] R. C. Guido, "Paraconsistent feature engineering," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 154–158, 2019.

Digital Object Identifier 10.1109/MSP.2018.2890523
Date of publication: xxxxxx