Over-sampling the minority class in the feature space

M. Pérez-Ortiz, P.A. Gutiérrez, *Senior Member, IEEE,* P. Tiňo and César Hervás-Martínez, *Senior Member, IEEE,*

***Abstract*—The imbalanced nature of some real-world data is one of the current challenges for machine learning researchers. One common approach over-samples the minority class through convex combination of its patterns. We explore the general idea of synthetic over-sampling in the feature space induced by a kernel function (as opposed to input space). If the kernel function matches the underlying problem, the classes will be linearly separable and synthetically generated patterns will lie on the minority class region. Since the feature space is not directly accessible, we use the empirical feature space (a Euclidean space isomorphic to the feature space) for over-sampling purposes. The proposed method is framed in the context of support vector machines where imbalanced datasets can pose a serious hindrance. The idea is investigated in three scenarios: 1) over-sampling in the full and reduced-rank empirical feature spaces; 2) a kernel learning technique maximising the data class separation to study the influence of the feature space structure (implicitly defined by the kernel function); 3) a unified framework for preferential over-sampling that spans some of the previous approaches in the literature. We support our investigation with extensive experiments over 50 imbalanced datasets.**

***Index Terms*—Over-sampling, imbalanced classification, kernel methods, empirical feature space, support vector machines**

I. Introduction

Classification methods often conveniently assume that the prior class probability distribution is of high entropy. However, this is not the case in many real-world applications from areas such as medical diagnosis, information retrieval, fraud detection, etc. The classification paradigm when one or several classes have a much lower prior probability in the training set is known as imbalanced classification [1], [2] and it poses a difficult challenge for machine learning researchers. Because of that, imbalanced classification is currently receiving a lot of attention from the pattern recognition and machine learning communities [3]–[9]. Often, the minority class happens to be more important than the majority one, but it may also be much more difficult to model due to the low number of available samples. Since most traditional learning systems have been designed to work on balanced data, they will usually be focused on improving overall performance and be biased towards the majority class, consequently harming the minority one [10]. Although from a formal definition an imbalanced dataset is any set of labelled data exhibiting an unequal distribution between classes, it has been shown that this is not the only factor involved hindering the learning in this context [1], [2]. The complexity of the data (existence of noisy and non-representative samples or class overlapping) or the size of the training set (high-dimensional data or small sample size) can also be part of the nature of the class imbalance problem. The approaches developed over the years for tackling the class imbalance problem can be categorised in two groups:

- Data approach - based on sampling methods, including over-sampling minority groups (groups of interesting rare examples), or under-sampling majority groups (groups with large example sizes), the combination of both being also very popular [1].
- Algorithm approach - forces the classifier to pay more attention to the minority class (e.g. by cost-sensitive learning [11]).

The analysis made in this paper is contextualised on data approaches. Thus, a brief discussion on these techniques is now given (for a detailed review of over-sampling see [1]). Roughly speaking, it can be said that over-sampling and under-sampling are opposite and equivalent, since they are aimed at the same purpose (i.e., balance the class distribution) but using different approaches. Formally, over-sampling concerns to the process of sampling a distribution with a significantly higher frequency than the given one and under-sampling to the process of reducing the frequency of the majority class. In both cases, the methodologies impose a balance in the class distribution in order to avoid aliasing and focus on the classification of minority classes. Although both over-sampling and under-sampling approaches have been shown to improve classifier performance over imbalanced datasets, different studies suggest that over-sampling is more useful than under-sampling [2], specially for highly imbalanced and complex datasets. Recall that under-sampling could entail a loss of potentially meaningful information of the dataset.

Concerning over-sampling, the first idea is to perform a random replication of minority data, but this often leads to over-fitting [10]. Another common approach is to generate new synthetic patterns according to the minority class distribution. One of the most well-known methods to do so is the synthetic minority over-sampling technique (SMOTE) [3] based on generating new instances by convex combination of one point and one of its k-nearest neighbours (both belonging to the minority class). However, the classes in general cannot be assumed to be convex and hence SMOTE does not avoid synthetic patterns to fall inside majority regions, therefore, more careful techniques have been developed to prevent this issue (prevent, but not

The work of M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez has been subsidized by the TIN2014-54583-C2-1-R project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P11-TIC-7508 project of the "Junta de Andalucía" (Spain). The work of P. Tino has been supported by EPSRC grant EP/L000296/1. M. Pérez-Ortiz, P.A. Gutiérrez and C. Hervás-Martínez are with the Department of Computer Science and Numerical Analysis of the University of Córdoba, Spain, email: {i82perom,pagutierrez,chervas}@uco.es. P. Tiňo is with the School of Computer Science of the University of Birmingham, Birmingham, United Kingdom, email: p.tino@cs.bham.ac.uk

solve). Adaptive synthetic [5]–[7] and cluster-based sampling methods [8], [9] are examples of more powerful techniques, based on extracting knowledge from the data to analyse which patterns and regions of the space are more suitable for over-sampling. This will be referred in the paper to as preferential over-sampling. At the same time, kernel methods have been spreading rapidly and gaining acceptance in machine learning due to their good generalisation ability and determinism, being one of the most widely used the Support Vector Machine (SVM) [12], [13]. However, for SVM, imbalanced data pose a serious challenge, due to the formulation of the soft-margin maximisation which focus on improving overall performance. Thus, the combination of kernel methods with techniques for tackling class imbalance is widely spread [4], [14].

It is clear that over-sampling by linear interpolation is not as suitable when dealing with nonlinear classifiers as it could be than when applying linear classifiers. However, linearly separable datasets are not common in real-world applications, thus making advisable the application of classifiers able to capture this nonlinearity. Besides, the development of a suitable nonlinear over-sampling strategy could be tricky. Thus, in contrast to previous approaches, we propose to generate new synthetic data by convex combination of points in a space where the classes are (ideally) linearly separated - making generation of new synthetic points by convex combination of the original points belonging to the same class safe. This is done using the feature space induced by a kernel function for over-sampling the patterns rather than using the input space. However, this is not so straightforward, because when dealing with kernel methods the only information available is the dot products of the images of the patterns [15]. To cope with this issue, this paper makes use of the notion of the empirical feature space (EFS) [16], [17], which is Euclidean and preserves the geometrical structure of the original feature space, given that distances and angles in the feature space are uniquely determined by dot products and that the dot products of the corresponding images are the original kernel values.

The main motivation for performing over-sampling in the EFS (instead of in the input space) is the hypothesis that the feature space provide a more suitable space for over-sampling via convex combination because the class separation will be simpler and larger (ideally, due to the kernel trick linearly separable). At the same time, this technique can be seen as a general nonlinear over-sampling in the input space due to the application of the nonlinear map Φ related to the kernel trick and could be used in combination with any classifier.

To the best of our knowledge, performing over-sampling in the feature space has only been researched in [14] (recall that in our case, it is performed in the EFS). In this previous work, the synthetic instances were generated by using the geometric interpretation of the dot products in the kernel matrix, and the pre-images of the synthetic instances were approximated based on a distance relation between the feature space and the input one, since inverse mapping $\Phi(\cdot)^{-1}$ from the feature space to the input space is not available. Our proposal is free of the assumptions of this inverse mapping approximation.

The study made in this paper intends to provide an extensive analysis of over-sampling in the EFS and can be subdivided in three sections. The first one deals with the issue of extending the SMOTE algorithm to be used in the full and reduced-rank EFS. The objective is to test whether the EFS provides a more suitable framework for over-sampling by convex combination of patterns and to deal with the dimensionality of the EFS. The second part deals with the kernel function choice (since our methodology depends on how the kernel matches the underlying classification problem) and we develop a strategy for optimising the feature space based on analytical knowledge (using the notion of kernel-target alignment [18], [19]). Ideally, a better fitted kernel will increase the class separability, providing a '*safer*' environment for the generation of synthetic patterns. The last part of this paper proposes a unified adaptive framework for preferential over-sampling generalising several over-sampling approaches in the literature [3], [5], [6]. The optimal SVM hyperplane and kernel learning techniques are used for optimising the synthetically generated patterns. The objective is to check if some regions of the space can be more useful for over-sampling than others. To test the different hypotheses exposed in this paper, we perform a thorough set of experiments with 50 binary imbalanced datasets.

The paper is organized as follows: Section II introduces some useful notions; Section III exposes how to perform over-sampling in the EFS; Section IV develops a new methodology for kernel learning; Section V proposes a general preferential over-sampling framework; Section VI exposes the experimental study and analyses the results obtained; and finally, Section VII outlines some conclusions and future work.

II. Background

This section is intended to introduce the notation used throughout all the paper and to provide some previous notions about SVM classifiers and the empirical feature space.

Consider a sample $D = \{\mathbf{x}_i, y_i\}_{i=1}^m \subseteq \mathcal{X} \times \mathcal{Y}$ generated i.i.d. from a (unknown) joint distribution $P(\mathbf{x}, y)$, where $\mathcal{X} \subseteq \mathbb{R}^d, \mathcal{Y} = \{+1, -1\}$. The goal in binary classification is to assign an input vector $\mathbf{x}$ to one of 2 classes $\{+1, -1\}$. Denote by $X^{\mathtt{tr}}$ and $X^{\mathtt{ts}}$ the sets of training and testing inputs, respectively. Furthermore, we will mark by subscript $+$ and $-$ to the sets containing inputs from the positive and negative class, respectively. For a set X, we denote by $\mathbf{X}$ the design matrix storing points of X as rows.

Reproducing kernels (often referred as Mercer kernels) [15] are functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which for all pattern sets $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ give rise to semidefinite positive matrices $\mathbf{K}_{m\times m}$, where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Kernel functions allow us to derive nonlinear classifiers by reducing them to linear ones but in some Hilbert space $\mathcal{H}$ nonlinearly related to the input space and furnished with a dot product $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$. The use of this kernel function instead of the dot product in $\mathbb{R}^m$ corresponds to using a (usually) nonlinear mapping of patterns from $\mathcal{X}$ to a high-dimensional or infinite-dimensional Hilbert Space $\mathcal{H}$ such that $\Phi : \mathcal{X} \to \mathcal{H}$, where the separation would ideally be easier, and take the dot product there. Kernel machines trained on D do not operate on the whole of $\mathcal{H}$ but on its subset $\mathcal{F} = \operatorname{span}\{\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_m)\}$, which we will refer to as the feature space such that $\mathcal{F} \subset \mathcal{H}$. Note that $\mathcal{F}$ is at most an m-dimensional linear space.

A. Support Vector Machines

SVM [12], [13] is perhaps the most common kernel method for statistical pattern recognition due to its good generalisation ability and freedom from local minima. The basic idea behind this technique is the separation of two different classes through a hyperplane which is specified by a normal vector $\mathbf{w}$ and a bias b. The optimal separating hyperplane is the one which maximises the distance between the hyperplane and the nearest points in both classes (called margin). Beyond the application of kernel techniques to allow non-linear decision discriminants (the kernel trick), another generalisation was made to replace hard margins with soft margins [13], using the so-called slack-variables ξ_i in order to deal with overlapping classes. Therefore, this algorithm seeks for a classifier $f : \mathbb{R}^d \to \mathbb{R}$ of the form $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$ (Φ being the mapping function induced by the kernel) that minimises the objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i, \tag{1}$$

for some parameter C, subject to the constraints:

$$y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall_i \in \{1, \ldots, m\}.$$

It is clear that, using SVMs, the soft-margin maximization paradigm poses a serious hindrance for imbalanced datasets [20]. The main reason for this is that soft-margin SVM optimisation is focused on overall error, therefore, they are inherently biased toward the majority class. In the worst case, for a noisy and highly imbalanced dataset, the SVM paradigm is very likely to obtain a trivial classifier (i.e., the one that classifies all the patterns in the majority class), a solution that, as said, if the imbalance is severe, could provide the minimal error [1]. To cope with this issue, several studies in the machine learning literature have explored different solutions to the imbalanced classification problem considering the SVM paradigm. Most of them are based on over-sampling [20], under-sampling [4], cost-sensitive classification [**?**], ensembles [21], [22] and kernel optimisation techniques [23], [24], among others [25], [26]. However, some studies suggest that under-sampling is not as effective as over-sampling in this case because of the potential loss of information on the class boundaries [20], which is crucial for the SVM solution.

B. Synthetic minority over-sampling technique (SMOTE)

As stated, one of the most widely used techniques for over-sampling is the SMOTE algorithm [3]. The process is very simple: the method consists on generating new instances on the line that connects one randomly chosen point with one of its k-nearest neighbours [27], both belonging to the minority class. Therefore, this methodology relies on a convex combination of two patterns. Note that with this approach new patterns could lie inside the majority class region (although choosing a correct value for the k parameter of the k-nearest neighbours method could avoid this to happen in some cases).

C. Empirical feature space (EFS)

We can endow an r-dimensional ($r \leq m$) space $\mathcal{F}$ with an orthonormal basis $\{\mathbf{u}_g\}, g \in B, B = \{1, 2, \ldots, r\}$, satisfying orthogonality, normalisation and completeness. Consider the set: $\mathcal{E} = \{\varphi(\mathbf{v}) | \mathbf{v} \in \mathcal{F}\}$, where $\varphi(\mathbf{v}) = \{\langle \mathbf{v}, \mathbf{u}_g \rangle_{\mathcal{F}}\}_{g \in B}$. The map φ is an isometric isomorphism of $\mathcal{F}$ and $\mathcal{E}$ [28], i.e. a bijective linear mapping such that the dot products are preserved: $\langle \varphi(\mathbf{v}), \varphi(\mathbf{v}') \rangle_{\mathcal{E}} = \langle \mathbf{v}, \mathbf{v}' \rangle_{\mathcal{F}}$. When $\mathcal{F}$ is the feature space, the set $\mathcal{E}$ is referred to as empirical feature space (EFS).

Consider a set of training points $\{\mathbf{x}_i\}_{i=1}^m \subseteq \mathcal{X}$. Then, when working with kernel methods we use a kernel function k to map the patterns to the feature space $\mathcal{F}$ and thus obtain a Gram matrix $\mathbf{K}$ with rank r, $r \leq m$. The nonlinear map from the input space to the r-dimensional Euclidean space $\Phi_r^e : \mathcal{X} \to \mathbb{R}^r$ which preserves the feature space structure is referred to as the empirical kernel map [16]. The EFS $\mathcal{E}$ is chosen so as to preserve the dot product information about $\mathcal{F}$ contained in $\mathbf{K}$, i.e., to be isometric isomorphic to the embedded feature space $\mathcal{F} \subset \mathcal{H}$. In this sense, it can be said that the empirical kernel map corresponds to a bijective linear mapping $\varphi : \mathcal{F} \to \mathcal{E}$.

A graphical representation of the input space, feature space, EFS and mappings between these spaces is shown in Fig. 1.

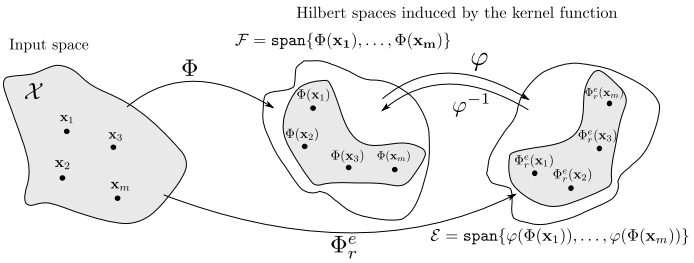

Fig. 1: Representation of the relation and mapping between input space, feature space and empirical feature space.

Any given Gram matrix $\mathbf{K}$ of rank r can be diagonalised as follows:

$$\mathbf{K}_{m\times m} = \mathbf{P}_{m\times r} \cdot \mathbf{\Lambda}_{r\times r} \cdot \mathbf{P}_{r\times m}^{\mathrm{T}},$$

where $(\cdot)^{\mathrm{T}}$ is the transpose operation, $\mathbf{\Lambda}$ is a diagonal matrix containing the r nonzero eigenvalues of $\mathbf{K}$ in decreasing order (i.e., $\lambda_1, \ldots, \lambda_r$), and $\mathbf{P}$ is a unitary matrix that consists of the eigenvectors associated to those r eigenvalues (i.e., $\mathbf{u}_1, \ldots, \mathbf{u}_r$) constituting an orthonormal basis of $\mathbb{R}^r$. Then, the empirical kernel map is defined as:

$$\Phi_r^e : \mathbf{x}_i \to \mathbf{\Lambda}^{-1/2} \cdot \mathbf{P}^{\mathrm{T}} \cdot (k(\mathbf{x}_i, \mathbf{x}_1), \ldots, k(\mathbf{x}_i, \mathbf{x}_m))^{\mathrm{T}}. \tag{2}$$

Consider the set $\{\Phi_r^e(\mathbf{x}_1), \ldots, \Phi_r^e(\mathbf{x}_m)\}$ of the EFS images of the training points. Let $\mathbf{Z}_{m\times r}$ be the design matrix storing $\Phi_r^e(\mathbf{x}_i)$ as rows. It is easy to check that the standard dot product matrix of $\Phi_r^e(\mathbf{x}_i)$, $i = 1, \ldots, m$ evaluated in $\mathcal{E}$ is $\mathbf{K}$ [16], [17]. Writing $\mathbf{Z} = \mathbf{\Lambda}^{-1/2} \cdot \mathbf{P}^{\mathrm{T}} \cdot \mathbf{K}$, we obtain[1]:

$$\mathbf{Z}^{\mathrm{T}}\mathbf{Z} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\mathrm{T}} = \mathbf{K}.$$

Since the distances and the angles of the m vectors $\Phi(\mathbf{x}_i), (i = 1, \ldots, m)$ in the feature space are uniquely determined by the

[1]Note that $\mathbf{P}$ is a unitary matrix and $\mathbf{K}$ a symmetric matrix

dot product (i.e., $||\Phi(\mathbf{x}_i)-\Phi(\mathbf{x}_j)||^2 = k(\mathbf{x}_i,\mathbf{x}_i)+k(\mathbf{x}_j,\mathbf{x}_j)-2k(\mathbf{x}_i,\mathbf{x}_j)$), the training data have the same geometrical structure in both spaces $\mathcal{F}$ and $\mathcal{E}$.

However, recall that the map Φ into the feature space is nonlinear, therefore each point in the span of the mapped input data would not necessarily be the image of some input pattern [16], [29]. This is known as the preimage problem. This problem also appears when using the empirical kernel map, because it also corresponds to a nonlinear transformation. Note that this is not a problem for the over-sampling of minority class, since the linear decision boundary is built in the feature space and if the classes are (almost) linearly separable in the feature space, doing local convex combination is reasonable, whether the pre-images of the synthetic points exist or not.

III. Synthetic over-sampling by convex combination in the EFS

The main hypothesis in this section is that the EFS provide us with a more suitable class distribution for over-sampling. It is clear that when classes are nonlinearly separable (which may occur in the input space), one should be very careful when creating synthetic patterns by convex combination because these could lie on the majority class region. However, if the data are linearly separable (a statement that will be true if the kernel function matches the underlying learning problem), over-sampling by convex combination of patterns is not a problem. To illustrate this, consider Fig. 2 where a toy nonlinearly separable dataset have been represented by the Φ_2^e transformation using a Gaussian kernel retaining only two dominant dimensions[2].

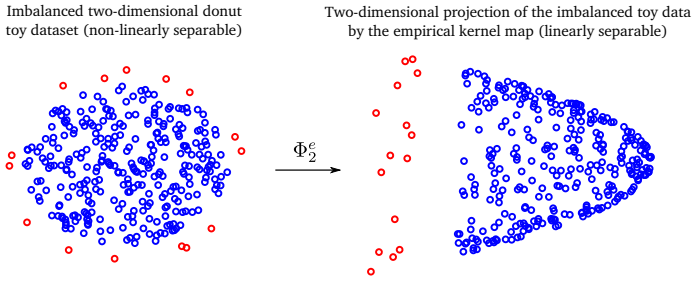

Fig. 2: Synthethic two-dimensional dataset representing a nonlinearly separable classification problem and their transformation to the 2 dominant dimensions of the EFS $\mathcal{E}^{(2)}$ induced by the Gaussian kernel function (linearly separable problem).

A. Reduced empirical feature space

In this subsection, we present a reduced version of the EFS, where we select the q $(q < r)$ dominant dimensions to approximate the kernel matrix.

In relation to classification, it has been argued that most decisive information can be contained in a subspace of the feature space [30] (under the assumption of smooth kernels matching the underlying problem). However, for the case of SVMs, the capacity control (inclusion of slacks variables and parameter cross-validation for preventing over-fitting) is equivalent to some form of regularisation so that "denoising" is not necessary although it could be very useful for unregularised methods [31]. In this section, we test whether over-sampling a minority class in the reduced dimensionality EFS (as opposed to over-sampling in the full EFS) can be beneficial. One motivation for over-sampling in reduced dimensionality EFS is that the over-sampling procedure relies on distances in the EFS to perform the neighbourhood analysis. Roughly speaking, these distances have been proven to be misleading as the data dimensionality increases, making more probable that the neighbours are chosen in a random fashion [32], [33].

It is well-known that for any real symmetric $m \times m$ matrix $\mathbf{K}$ of rank r, we can find its real nonzero eigenvalues $\lambda_1 \geq \ldots \geq \lambda_r$ and the corresponding orthonormal eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_r$, so that $\mathbf{K} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$. In this case, the best rank-q $(q < r)$ approximation to $\mathbf{K}$ is $\mathbf{K}_q = \sum_{i=1}^{q} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathrm{T}}$, in the sense that it minimises $||\mathbf{K}-\mathbf{K}_q||_{\mathrm{F}}^2$ over all rank-q matrices (where $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm). This concept can be said to be the main idea for the reduced EFS.

Instead of working in the full-rank EFS $\mathcal{E}$ we can operate in its lower dimensional subspace $\mathcal{E}^{(q)}$ where the kernel matrix has the form:

$$\mathbf{K}^{(q)}_{m\times m} = \mathbf{P}^{(q)}_{m\times q} \cdot \mathbf{\Lambda}^{(q)}_{q\times q} \cdot (\mathbf{P}^{(q)})^{\mathrm{T}}_{q\times m},\ q < r,$$

where $\mathbf{P}^{(q)}$ and $\mathbf{\Lambda}^{(q)}$ consist of the first q columns of $\mathbf{P}$ and $\mathbf{\Lambda}$, respectively[3].

Consider the preimage $\mathcal{F}^{(q)}$ of $\mathcal{E}^{(q)}$ under the isomorphism φ. Let $\{\mathbf{u}_j\}_{j=1}^{q}$ be an orthonormal basis of $\mathcal{F}^{(q)}$. Given $\mathbf{v} \in \mathcal{F}$, its projection onto $\mathcal{F}^{(q)}$ is obtained as $\{\langle \mathbf{v}, \mathbf{u}_j \rangle_{\mathcal{F}}\}_{j=1}^{q}$. The isomorphism φ from $\mathcal{F}$ to $\mathcal{E}$ carries the structure over: $\varphi(\mathbf{v}) \in \mathcal{E}$ is projected onto $\mathcal{E}^{(q)}$ as $\{\langle \varphi(\mathbf{v}), \varphi(\mathbf{u}_j) \rangle_{\mathcal{E}}\}_{j=1}^{q}$. Moreover, for all $j = 1, \ldots, q$,

$$\langle \mathbf{v}, \mathbf{u}_j \rangle_{\mathcal{F}} = \langle \varphi(\mathbf{v}), \varphi(\mathbf{u}_j) \rangle_{\mathcal{E}}.$$

Therefore, we could define the kernel associated with the reduced EFS by:

$$k^{(q)}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \Phi_q^e(\mathbf{x}_i), \Phi_q^e(\mathbf{x}_j) \right\rangle_{\mathcal{E}},$$

which, for q being the rank of $\mathbf{K}$, will correspond to k.

B. Synthetic minority over-sampling in the reduced or full-rank EFS

Once that the notion of EFS has been introduced, this subsection will show the main steps to extend a well-known over-sampling algorithm to this space.

Concerning the training phase, the first step of the proposed methodology corresponds to the computation of the training kernel matrix $\mathbf{K}$ through a predefined kernel function k. Then, the reduced or full-rank empirical kernel map Φ_q^e, $1 \leq q \leq r$, can be computed via the eigenvector decomposition of this training kernel matrix $\mathbf{K}$ (Eq. (2)). As said, let Z be the set generated by applying the Φ_q^e transformation to the training patterns and $\mathbf{Z}_{m\times q}$ the design matrix storing points of Z as rows. In the second step, the over-sampling process is

[2]Dimensions associated with the highest eigenvalues of the Gram matrix.

[3]We assume that the singular values are sorted.

performed over the minority class images of this $\mathbf{Z}$ matrix, resulting in the generation of n new synthetic images, arranged in the set S (and the design matrix $\mathbf{S}_{n\times q}$). More specifically, as the standard SMOTE algorithm [3] has been chosen for over-sampling, each new synthetic instance will be generated using a linear interpolation between pattern $\mathbf{x}_i$ and one of its k-nearest neighbours (both belonging to the minority class). At every step $j = 1, \ldots, n$, we create a point $\mathbf{s}_j$ in $\mathcal{E}^{(q)}$ by picking at random a minority class point $\mathbf{x}_i$ and calculating:

$$\mathbf{s}_j = \Phi_q^e(\mathbf{x}_i) + (\Phi_q^e(\hat{\mathbf{x}}_i) - \Phi_q^e(\mathbf{x}_i)) \cdot \delta,$$

where $\Phi_q^e(\hat{\mathbf{x}}_i)$ is one of the k-nearest neighbours for $\Phi_q^e(\mathbf{x}_i)$ in the EFS $\mathcal{E}^{(q)}$, and δ is a random number generated from the uniform distribution $U[0, 1]$. For simplicity, we over-sample the minority class so that the two classes become balanced. From the definition of the EFS, we know that $\varphi^{-1}(\mathbf{s}_j) \in \mathcal{F}^{(q)}$ (i.e., the representation of the new pattern in the feature space) will be unique and will lie on the line between $\varphi^{-1}(\mathbf{x}_i)$ and $\varphi^{-1}(\hat{\mathbf{x}}_i)$ (φ is a linear map). Recall that the norms and distances are preserved, e.g.:

$$\|\Phi_q^e(\hat{\mathbf{x}}_i) - \Phi_q^e(\mathbf{x}_i)\|_{\mathcal{E}} = \|\Phi(\hat{\mathbf{x}}_i) - \Phi(\mathbf{x}_i)\|_{\mathcal{F}},$$

and so are the angles, $(\Phi_q^e(\mathbf{x}_i) - \Phi_q^e(\hat{\mathbf{x}}_i))^{\mathrm{T}}(\Phi_q^e(\mathbf{x}_i) - \mathbf{s}_j) = \langle \Phi(\mathbf{x}_i) - \Phi(\hat{\mathbf{x}}_i), \Phi(\mathbf{x}_i) - \varphi^{-1}(\mathbf{s}_j)\rangle$. As a consequence, if $\Phi_q^e(\hat{\mathbf{x}}_i)$ is one of the k-nearest neighbours of $\Phi_q^e(\mathbf{x}_i)$ in the EFS, this will be so in the feature space as well.

The third step is the execution of the learning machine over the set $\varphi^{-1}(Z \cup S) \subset \mathcal{F}^{(q)}$. In this case, there are two different possibilities to consider. First, we could employ the EFS as a new representation for the data and use the classification algorithm in this new space as done in other works [34], [35]. This idea will provide us with a more easily separable and balanced space than the input space which could indeed be used for any learning machine, independently of being kernelized or not. However, when dealing with a kernel function, it could actually be more advisable to recompute the dot products between patterns (i.e., create a new over-sampled kernel matrix), due to the high number of features (the dimensionality of the EFS), which in most of the cases will increase the computational cost of the learning machine considered. To do so, synthetic samples will be used to complete the kernel matrix, by obtaining their dot product in the EFS with respect to the rest of the training patterns. Using this approach, the over- sampled training Gram matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ will be composed as follows:

$$\tilde{\mathbf{K}}^{\mathrm{tr}}_{(m+n)\times(m+n)} = \begin{pmatrix} (\mathbf{Z}\cdot\mathbf{Z}^{\mathrm{T}})_{m\times m} & (\mathbf{Z}\cdot\mathbf{S}^{\mathrm{T}})_{m\times n} \\ (\mathbf{S}\cdot\mathbf{Z}^{\mathrm{T}})_{n\times m} & (\mathbf{S}\cdot\mathbf{S}^{\mathrm{T}})_{n\times n} \end{pmatrix}. \quad (3)$$

Note that for any number of dominant dimensions q for the empirical kernel map Φ_q^e, the over-sampled kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ obtained will be positive semidefinite. Furthermore, since we are generating new patterns by a linear combination of other patterns in the dataset, the empirical kernel maps associated to $\varphi^{-1}(Z)$ and to $\varphi^{-1}(Z \cup S)$ can be said to be equivalent.

For the generalisation phase, the same steps are considered to complete the test kernel matrix, considering that the EFS images of the test patterns are derived using the same Φ_q^e map (considering only the training data). Note that in this case we will compute the dot product between train and test patterns and between test and synthetic patterns. The over- sampled test Gram matrix $\tilde{\mathbf{K}}^{\mathrm{ts}}$ will be composed as follows:

$$\tilde{\mathbf{K}}^{\mathrm{ts}}_{(m+n)\times(t)} = \begin{pmatrix} (\mathbf{Z}\cdot\mathbf{T}^{\mathrm{T}})_{m\times t} & (\mathbf{S}\cdot\mathbf{T}^{\mathrm{T}})_{n\times t} \end{pmatrix}, \quad (4)$$

where $\mathbf{T}$ is the representation in the EFS of the test patterns and t corresponds to the number of test patterns.

Note that these new over-sampled kernel matrices $\tilde{\mathbf{K}}^{\mathrm{tr}}$ and $\tilde{\mathbf{K}}^{\mathrm{ts}}$ can be used for any kernel-based algorithm.

A summary of this kernel-based over-sampling method can be seen in Fig. 3.

Algorithm synthetic over-sampling in the empirical feature space

- **Input**: Training patterns ($\mathbf{X}^{\mathrm{tr}}$), training targets (y^{tr}) and testing patterns ($\mathbf{X}^{\mathrm{ts}}$).
- **Output**: Testing targets (y^{ts})

1) Compute kernel matrix $\mathbf{K}^{\mathrm{tr}}$ for training patterns.
2) Compute the empirical kernel map Φ_q^e via $\mathbf{K}^{\mathrm{tr}}$.
3) Map training patterns to the EFS using Φ_q^e and obtain their new representation $\mathbf{Z}$.
4) Generate synthetic patterns $\mathbf{S}$ using the new representation $\mathbf{Z}$ of the training patterns.
5) Complete the over-sampled train kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ with the dot product between patterns (Eq. 3).
6) Train the learning algorithm with kernel matrix $\tilde{\mathbf{K}}^{\mathrm{tr}}$ and obtain a hyperplane $\mathbf{w}$ and a bias term b.
7) Map testing patterns to the EFS using Φ_q^e and obtain their new representation $\mathbf{T}$.
8) Complete the over-sampled test kernel matrix $\tilde{\mathbf{K}}^{\mathrm{ts}}$ with the dot product between patterns (Eq. 4).
9) Predict y^{ts} using $\tilde{\mathbf{K}}^{\mathrm{ts}}$ and the model $\{\mathbf{w}, b\}$ (Eq. 1).

Fig. 3: Different steps for the kernel over-sampling algorithm.

As mentioned before, our over-sampled points in the feature space may not have preimages in the input space. However, this does not pose a methodological problem since the class separation is formulated in the feature space.

IV. Optimising the feature space by kernel learning for over-sampling

As stated before, our first hypothesis was that over-sampling in the EFS was more advisable if the kernel function matched the underlying problem in the sense that it can asymptotically represent the function to be learned and is sufficiently smooth. In this section, we propose a method for kernel learning that would ideally provide a clearer class separation in the feature space to analyse its effect in the over-sampling method.

Ideally, we would like to find the kernel that minimises the true risk of a classifier for a specific dataset. Unfortunately, the risk is not accessible; therefore, different analytical bounds for the generalisation error have been developed in the machine learning literature with the aim of better suiting a given dataset. In the kernel machine literature, a considerable interest has been devoted to learning the "optimal" kernel given a particular classification task, as opposed to imposing them. One of the prominent approaches in kernel learning is centred kernel-target alignment (KTA) [19]. Centred KTA is data distribution independent, making it particularly suitable for imbalanced classification. Note that KTA is related to the Fisher criterion, which maximises the distance between different classes and minimises the within class distance. This can be a useful property of the feature space in which to perform minority

class over-sampling. Minority patterns would be far from the majority class region and closely clustered together.

KTA optimises the kernel by aligning it to the so-called ideal kernel matrix $\mathbf{K}_{\mathrm{i}}$ [18], which will submit the structure:

$$k_{\mathrm{i}}(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 \ \text{ if } y_i = y_j, \\ -1 \ \text{ otherwise,} \end{cases}$$

where y_i is the target of pattern $\mathbf{x}_i \in X^{\mathtt{tr}}$. In this sense, $\mathbf{K}_{\mathrm{i}}$ will provide information about which patterns should be considered to be similar when performing a learning task.

Thus, the problem of finding an optimal kernel k is changed to the one of finding a good approximation $\mathbf{K}$ for the ideal kernel matrix $\mathbf{K}_{\mathrm{i}}$, given a family of kernel functions. This formulation allows to separate the optimisation from kernel machine learning and to reduce the increase in the computational cost of learning more complex kernels, given that the kernel machine will be unaffected by this higher complexity.

As said before, concerning imbalanced classification, previous studies have noted several issues in KTA for different pattern distributions [18], [36] but a recent study [19] has shown that this can be solved by the use of centred kernel matrices. The notion of centred alignment $\mathcal{A}_{\mathrm{c}}$ between $\mathbf{K}$ and $\mathbf{K}_{\mathrm{i}}$ [18], [19] is defined as:

$$\mathcal{A}_{\mathrm{c}}(\mathbf{K}, \mathbf{K}_{\mathrm{i}}) = \frac{\langle \mathbf{K}_{\mathrm{c}}, \mathbf{K}_{\mathrm{i}_{\mathrm{c}}} \rangle_{\mathrm{F}}}{\sqrt{\langle \mathbf{K}_{\mathrm{c}}, \mathbf{K}_{\mathrm{c}} \rangle_{\mathrm{F}} \langle \mathbf{K}_{\mathrm{i}_{\mathrm{c}}}, \mathbf{K}_{\mathrm{i}_{\mathrm{c}}} \rangle_{\mathrm{F}}}},$$

where $\mathbf{K}_{\mathrm{c}}$ denotes the centred version of kernel matrix $\mathbf{K}$ and is computed as:

$$\mathbf{K}_{\mathrm{c}} = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{m}} - \mathbf{1}_{\frac{1}{m}}\mathbf{K} + \mathbf{1}_{\frac{1}{m}}\mathbf{K}\mathbf{1}_{\frac{1}{m}},$$

being $\mathbf{1}_{\frac{1}{m}}$ a matrix with all elements equal to $\frac{1}{m}$.

Centred KTA is maximised when a kernel reflect the discriminant properties of the data used to define the ideal kernel.

Consider a kernel function depending on a vector of parameters $\boldsymbol{\alpha}$. Because of the differentiability of $\mathcal{A}_{\mathrm{c}}$ with respect to these kernel parameters $\boldsymbol{\alpha}$, a gradient ascent algorithm can be used to maximise the alignment between the kernel matrix constructed $\mathbf{K}_{\boldsymbol{\alpha}}$ and the ideal one $\mathbf{K}_{\mathrm{i}}$, as follows: $\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{i}})$. The alignment derivative with respect to these kernel parameters $\boldsymbol{\alpha}$ is:

$$\frac{\partial \mathcal{A}_{\mathrm{c}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{i}})}{\partial \boldsymbol{\alpha}} = \tag{5}$$
$$= \frac{1}{||\mathbf{K}_{\mathrm{i}_{\mathrm{c}}}||_{\mathrm{F}}} \left[\frac{\left\langle \left(\frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right), \mathbf{K}_{\mathrm{i}_{\mathrm{c}}} \right\rangle_{\mathrm{F}}}{||\mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}||_{\mathrm{F}}} - \frac{\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_{\mathrm{i}_{\mathrm{c}}} \rangle_{\mathrm{F}} \cdot \left\langle \mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}, \left(\frac{\partial \mathbf{K}_{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}}\right) \right\rangle_{\mathrm{F}}}{||\mathbf{K}_{\boldsymbol{\alpha}_{\mathrm{c}}}||_{\mathrm{F}}^3} \right],$$

where $||\mathbf{A}||_{\mathrm{F}} = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_{\mathrm{F}}}$ and, for arbitrary matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, it holds that $\langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_1, \mathbf{K}_{2_{\mathrm{c}}} \rangle_{\mathrm{F}} = \langle \mathbf{K}_{1_{\mathrm{c}}}, \mathbf{K}_2 \rangle_{\mathrm{F}}$ [19], which simplifies the computation.

In this paper, we will consider a generalised Gaussian kernel with covariance structure defined by a positive semidefinite matrix $\mathbf{Q}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left((\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \mathbf{Q} (\mathbf{x}_i - \mathbf{x}_j)\right).$$

As usual, the matrix $\mathbf{Q}$ will be parametrised by $\mathbf{U}^{\mathrm{T}}\mathbf{U}$, where $\mathbf{U}$ is a $d \times d$ matrix (d being the dimensionality of the input space). Therefore, we can equivalently restate our problem as learning the best matrix $\mathbf{U}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left((\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \mathbf{U}^{\mathrm{T}} \mathbf{U} (\mathbf{x}_i - \mathbf{x}_j)\right).$$

Now, we can compute the derivative of the kernel with respect to the entries of the $\mathbf{U}$ matrix:

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{U}}\right) = \left(\mathbf{U}(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} (\mathbf{x}_i - \mathbf{x}_j)\right) \cdot k(\mathbf{x}_i, \mathbf{x}_j).$$

Therefore, we will optimise a vector of parameters $\boldsymbol{\alpha}$ composed of the entries of the $\mathbf{U}$ matrix.

It is important to note that some attempts have been made to establish learning bounds for the Gaussian kernel with several parameters when considering large margin classifiers [37]. These studies suggest that the interaction between the margin and the complexity measure of the kernel class is multiplicative, thus discouraging the development of techniques for the optimisation of more complex kernels. However, recent developments have shown that this interaction is additive [38] (up to log factors), rather than multiplicative, yielding then stronger bounds. Therefore, the number of patterns needed to obtain the same estimation error with the same probability for a multi-scale kernel compared to a spherical one grows slowly (and directly depends on the number of parameters).

To demonstrate the usefulness of learning the kernels, we present in Fig. 4 a graphical representation of three two-dimensional toy datasets and their mapping Φ_2^e using a spherical Gaussian kernel with $\mathbf{Q} = 0.001 \cdot \mathbf{I}_d$, an optimised spherical Gaussian kernel obtained through centred KTA and an optimised generalised Gaussian kernel.

Summarising, kernel learning will be applied before the over-sampling procedure to learn a suitable kernel $\mathbf{K}_{\boldsymbol{\alpha}^*}$ for the data representation. After this, the EFS Φ_q^e associated to this kernel $\mathbf{K}_{\boldsymbol{\alpha}^*}$ will be computed, and then, the images of the training patterns for the minority class (contained in the $\mathbf{Z}$ matrix) will be over-sampled. For comparison purposes, we will also test the optimization of a spherical Gaussian kernel with one kernel parameter via kernel-target alignment.

V. Unified framework for preferential over-sampling

As stated before, several approaches have been developed in the literature for handling imbalanced data, and a large number of these contributions are based on analysing the patterns which could be more suitable for over-sampling, giving rise to approaches based on over-sampling on the class boundary [5], [7] or in the within class "safe region" [6] (these techniques are commonly referred to as weighted over-sampling). However, to our best knowledge, there is no principled method for choosing the region of the minority class to be used for over-sampling. In this section we propose a new adaptive weighted over-sampling technique that naturally spans unweighted and weighted over-sampling methods (both on the boundary and within class). To do so, our approach will take advantage of the spatial distribution of the patterns according to the optimal hyperplane obtained from the SVM solution.

A. Knowledge extraction: Spatial distribution of the patterns

Weighted over-sampling techniques are based on the idea that not all the patterns of the dataset are equally important and suitable for over-sampling and therefore, they should not

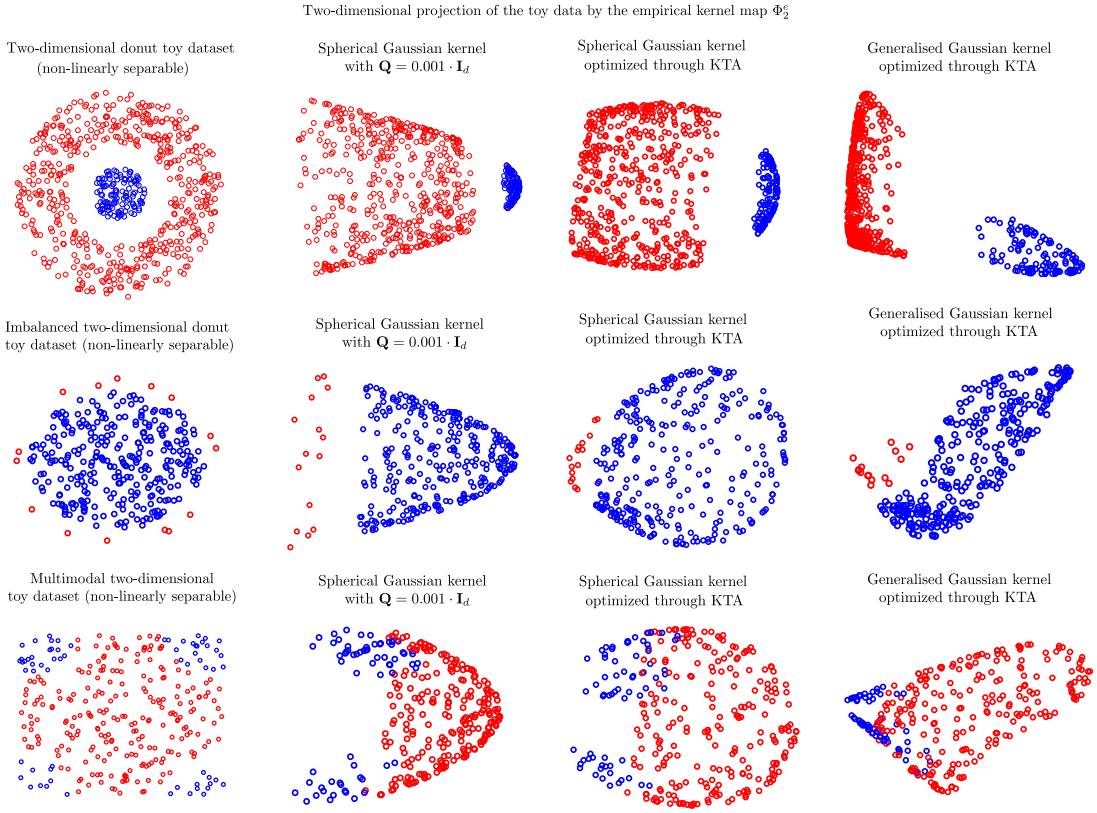

Fig. 4: Synthethic two-dimensional datasets representing non-linearly separable classification problems and their transformation to the 2 dominant dimensions of the EFS induced by the Gaussian kernel function (linearly separable problem).

contribute equally to the new synthetic data. One of the first steps of these methodologies corresponds to the identification of the '*useful*' patterns to be used for over-sampling. Most of the approaches in the literature do so by analysing local neighbourhood of points in the minority class. In this paper, however, we will derive a weighted over-sampling technique considering the spatial distribution of the patterns with respect to the optimal SVM hyperplane. In particular, the patterns to be used for over-sampling will be selected based on their position and distance to the optimal hyperplane.

However, as stated before, the soft-margin optimisation of the SVM paradigm poses a serious problem for imbalanced datasets. Therefore, for the purpose of weighted over-sampling, we use the cost-sensitive approach giving more importance to errors committed by patterns belonging to the minority class [11]. The cost-sensitive SVM approach consists of introducing different penalty factors C_{+1} and C_{-1} for the positive and negative SVM slack variables during training. The primal SVM problem is transformed into:

$$\frac{1}{2}||\mathbf{w}||^2 + C_{+1} \sum_{\{i|y_i=+1\}} \xi_i + C_{-1} \sum_{\{i|y_i=-1\}} \xi_i,$$

subject to the constraints:

$$y_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall_i \in \{1, \ldots, m\}.$$

For simplicity, we will set $C_{+1} = \frac{m_{-1}}{m_{+1}} \cdot C_{-1}$, where $+1$ is assumed to be the minority class, m_{+1} is the number of patterns belonging to class $+1$ and m_{-1} the number of patterns belonging to class -1. The ratio $\frac{m_{-1}}{m_{+1}}$ is usually known as the imbalanced ratio.

As stated before, each synthetically generated point $\mathbf{s}_z \in \mathcal{E}^{(q)}$, $z = 1, \ldots, n$, in the minority class represented by training samples X_+^{tr} is generated by first picking a pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ from X_+^{tr} and then constructing their convex combination in the EFS $\mathcal{E}^{(q)}$:

$$\mathbf{s}_z = \Phi_q^e(\mathbf{x}_i) + (\Phi_q^e(\mathbf{x}_j) - \Phi_q^e(\mathbf{x}_i)) \cdot \delta,$$

where δ is a random number generated from the uniform distribution $U[0, 1]$.

B. Optimisation of the over-sampling procedure

The points $\mathbf{x}_i$ and $\mathbf{x}_j$ will be randomly selected based on their relative position in the feature space with respect to the separating hyperplane. Because the norm of $\mathbf{w}$ is 1, the signed distance of $\Phi(\mathbf{x}_i) \in \mathcal{F}^{(q)}$ from the hyperplane is given by $f(\mathbf{x}_i) = \mathbf{w} \cdot \Phi(\mathbf{x}_i) + b$. Note that if $\Phi(\mathbf{x}_i)$ is on the '*right*' side of the hyperplane $f(\mathbf{x}_i)$ is positive, otherwise it is negative[4]. We will represent the selection process as draws from a multinomial distribution over X_+^{tr} (i.e., patterns belonging to the minority class) with natural parameters $\mu_i = -\beta \cdot f(\mathbf{x}_i)$, where $\beta \in \mathbb{R}$ is a scale parameter. Using the soft-max link function, the probability of picking $\mathbf{x}_i \in X_+^{\text{tr}}$ is:

$$P(\mathbf{x}_i) = \frac{\exp(-\beta f(\mathbf{x}_i))}{\sum_{\mathbf{x} \in \mathbf{X}_+^{\text{tr}}} \exp(-\beta f(\mathbf{x}))}. \tag{6}$$

[4]If $\Phi(\mathbf{x}_i)$ lies on the separating hyperplane, then $f(\mathbf{x}_i) = 0$.

Note that when $\beta < 0$, points deep within the minority class (in the feature space) are more likely to be picked; when $\beta > 0$, points closer to the class boundary or lying inside the opposite class are preferred and, when $\beta = 0$, all the points are equally likely to be chosen, as this will correspond to the uniform distribution over X_+^{tr}. This approach naturally spans different approaches to weighted [5]–[7] and unweighted [3] over-sampling previously introduced in the literature.

For selecting the pairs $(\mathbf{x}_i, \mathbf{x}_j) \in X_+^{\text{tr}}$ we could use two different ideas:

- Pick $\mathbf{x}_i$ and $\mathbf{x}_j$ independently with respect to the distribution of Eq. (6).
- Pick $\mathbf{x}_i$ according to the distribution of Eq. (6) and select $\mathbf{x}_j$ using k-nearest neighbours method [27].

In most of the weighted approaches in the literature they make use of the k-nearest neighbours method because they obtain the spatial distribution information of the patterns according to their neighbourhood. However, for this approach, note that it is actually more advisable to select $\mathbf{x}_i$ and $\mathbf{x}_j$ independently according to the probability distribution obtained, because otherwise the effect of the preferential learning in the over-sampling process could be smoothed (i.e., picking points by the k-nearest neighbours approach may differ to a large extent to the selection made with the probability function).

Based on the arguments in Section III, over-sampling of the minority class in the feature space is done through over-sampling in the EFS. Note that the patterns preferred for over-sampling in the input space could not be the ones preferred in the feature space, therefore the use of the EFS is needed for this methodology as well.

To optimise the β values (as different β values will induce different synthetic patterns), we will test two approaches:

- The first idea is to use a single value of β found by, e.g., cross-validation over a set of p predefined β values.
- The second idea is to use multiple β values within the framework of multiple kernel learning (MKL), i.e., a combination of different over-sampled kernel matrices. For a particular value of β, we denote by $\tilde{\mathbf{K}}_\beta$ the kernel matrix obtained on the extended data sample (i.e., including over-sampled points obtained using β). We fix a set of β values $\{\beta_1, \ldots, \beta_p\}$ and compute the over-sampled kernel matrices $\{\tilde{\mathbf{K}}_{\beta_1}, \ldots, \tilde{\mathbf{K}}_{\beta_p}\}$. Then, using KTA, we could derive a kernel matrix $\tilde{\mathbf{K}}_{\boldsymbol{\omega}} = \sum_{k=1}^{p} \omega_k \tilde{\mathbf{K}}_{\beta_k}$ with $\omega_k \geq 0$ and $\sum_{k=1}^{p} \omega_k = 1$ (convex combination of kernel matrices $\tilde{\mathbf{K}}_{\beta_k}$) by multiple kernel learning techniques. Thus, this strategy will be more flexible than the cross-validation one, because we can optimise a combination of over-sampled kernel matrices, instead of restricting the solution to only choosing the best performing one. For the optimisation we will need to define an extended ideal kernel matrix $\tilde{\mathbf{K}}_{\text{i}}$, by introducing the information of the new synthetic patterns (recall that all these patterns will belong to the minority class). The optimisation problem to solve in this case will be the following:

$$\max_{\boldsymbol{\omega} \in \mathcal{M}} \frac{\left\langle \tilde{\mathbf{K}}_{\boldsymbol{\omega}_c}, \tilde{\mathbf{K}}_{\text{i}} \right\rangle_{\text{F}}}{||\tilde{\mathbf{K}}_{\boldsymbol{\omega}_c}||_{\text{F}}},$$

where $\mathcal{M} = \{\boldsymbol{\omega} : ||\boldsymbol{\omega}||_2 = 1\}$. Note that since we are trying to align the real kernel matrix $\tilde{\mathbf{K}}$ with the ideal one $\tilde{\mathbf{K}}_{\text{i}}$ the value of $\left\langle \tilde{\mathbf{K}}_{\text{i}}, \tilde{\mathbf{K}}_{\text{i}} \right\rangle_{\text{F}}$ does not change and it can be obviated in the optimisation process. The Quadratic Programming (QP) optimization problem associated can be seen in [19].

Fig. 5 shows the representation of the training data for the cleveland0vs4 dataset in different EFS using the transformation Φ_2^e (original EFS, over-sampled EFS for $\beta = -5$ and $\beta = 5$, and optimised over-sampling through MKL). In this case, the difference between over-sampling for different β values could be difficult to appreciate. However, for the case of the optimised over-sampled EFS one can note that the class separation increases and the within class decreases (recall that KTA was related to the Fisher criterion).

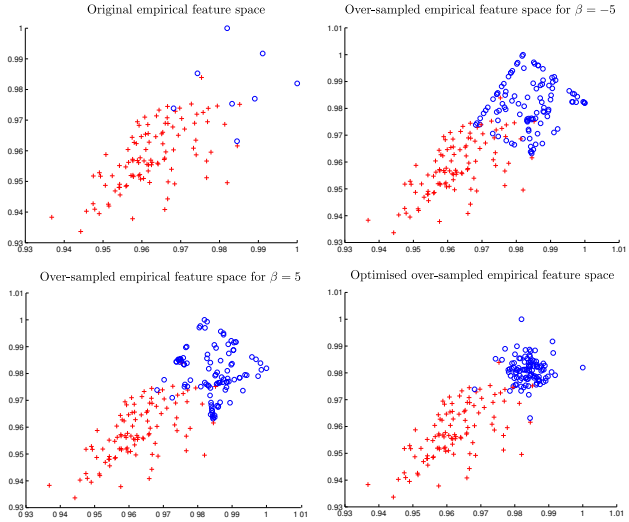

Fig. 5: Empirical feature spaces for the cleveland0vs4 dataset associated to the original data, over-sampling for different β values and optimised over-sampling.

In the same vein, Fig. 6 shows the case of the training data for the led7digit02456789vs1 dataset and the transformation Φ_2^e. In this case, the difference for the over-sampling procedure when using different β values can be easily appreciated.

VI. Experimental results

The proposed methodologies have been tested considering Support Vector Machines (SVM) [13] and the well-known SMOTE algorithm [3]. 50 binary datasets from the UCI repository [39] with different imbalance ratios (proportion of majority patterns with respect to minority ones) have been used to test the performance of the methods in different situations. The characteristics of these datasets can be seen in TABLE I. As done in other state-of-the-art works [10], some multiclass datasets have also been considered by grouping some classes, e.g. ecoli1 represents the ecoli dataset when considering class 1 versus the rest, and yeast0359vs78 is the yeast dataset when grouping classes 0, 3, 5, and 9 versus classes 7 and 8 in order to obtain higher imbalance ratio (IR) values.

A stratified 5×2-fold Dietterich technique was performed to divide the data and the results are taken as mean and

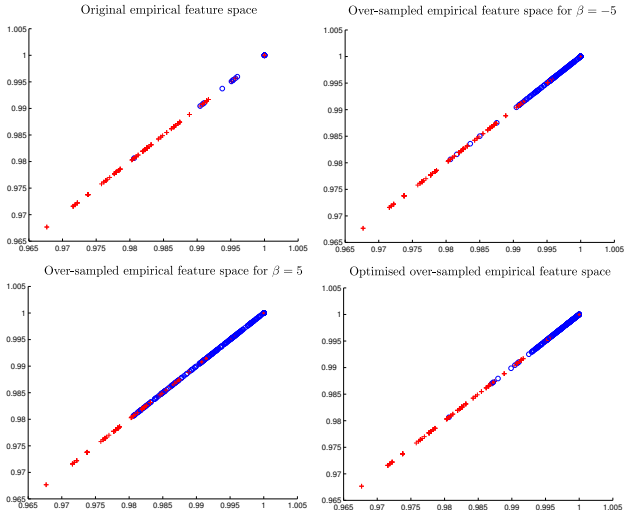

Fig. 6: Empirical feature spaces for the led7digit02456789vs1 dataset associated to the original data, over-sampling for different β values and optimised over-sampling.

standard deviation of the selected measures as done elsewhere (e.g. [10]). Each experiment over each data partition has been repeated 6 times using a different seed to obtain more robust results[5] (i.e., at the end of the execution we will have 30 results for each dataset). The Gaussian kernel was used. The kernel width and the cost parameter of SVM were selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ by means of a nested 5-fold method applied to the training set. As done in other works [8], [9], the number of synthetic patterns generated was that needed to balance the distributions, i.e. after applying the over-sampling process, the number of majority and minority patterns were the same. $k = 3$ nearest neighbours were evaluated to generate synthetic samples, in order to minimise the chance that synthetic patterns are generated in the majority class region when using the standard SMOTE technique.

The results have been reported in terms of two metrics, one of them specially designed to deal with imbalanced data:

1) The well-known Accuracy metric (Acc), which corresponds to the ratio of correctly classified patterns and measures overall performance. For imbalanced datasets, this metric may not be the best option, since the classification of the minority class may be compromised for the sake of the majority one (it does not distinguish between the numbers of correctly classified examples of each class), and we could therefore obtain a trivial classifier always outputting the majority class.
2) The Geometric Mean of the sensitivities ($GM = \sqrt{S_{\mathrm{p}} \cdot S_{\mathrm{n}}}$) [40], where S_{p} is the sensitivity for the positive class (ratio of correctly classified patterns considering only this class) and S_{n} is the sensitivity for the negative one.

The measure for the parameter selection was GM, given its robustness and extended use for imbalanced data [40]–[42]. Note that this metric gives much importance to worst-classified classes being therefore sensitive to trivial classifiers (e.g. if $S_{\mathrm{p}} = 0$ then $GM = 0$, independently of the value of S_{n}).

[5]Recall that synthetic patterns are randomly generated.

TABLE I: Datasets used for the experiments (N corresponds to the total number of patterns, d to the dimensionality of the input space and IR to the imbalance ratio).

Dataset	N	d	IR	Dataset	N	d	IR
ecoli0vs1	352	7	1.84	ecoli067vs35	354	7	9.41
glass1	342	9	1.85	glass04vs5	146	9	9.43
wisconsin	1092	9	1.86	ecoli0267vs35	358	7	9.53
pima	1228	8	1.87	yeast05679vs4	844	8	9.55
yeast1	2374	8	2.46	ecoli067vs5	352	6	10.00
haberman	488	3	2.81	glass016vs2	306	9	10.77
vehicle2	1352	18	2.89	ecoli01vs5	384	6	11.00
vehicle1	1352	18	2.91	led7digit02456789vs1	708	7	11.21
vehicle3	1352	18	3.00	glass06vs5	172	9	11.29
vehicle0	1352	18	3.25	glass0146vs2	328	9	11.62
glass0123vs456	342	9	3.28	glass2	342	9	12.15
ecoli1	536	7	3.39	ecoli0147vs56	530	6	12.25
newthyroid1	344	5	5.14	cleveland0vs4	276	13	12.80
newthyroid2	344	5	5.14	ecoli0146vs5	448	6	13.00
ecoli2	536	7	5.54	shuttle0vs4	2926	9	13.78
yeast3	2374	8	8.13	yeast1vs7	734	7	14.29
ecoli3	536	7	8.57	ecoli4	536	7	15.75
ecoli034vs5	320	7	9.00	pageblocks13vs4	754	10	16.14
yeast0359vs78	808	8	9.10	abalone9-18	1168	10	16.70
ecoli046vs5	324	6	9.13	glass016vs5	294	9	20.00
yeast0256vs3789	1606	8	9.16	yeast2vs8	770	8	23.06
yeast02579vs368	1606	8	9.16	shuttle2vs4	206	9	24.75
ecoli0347vs56	410	7	9.25	yeast4	2374	8	28.68
ecoli01vs235	390	7	9.26	yeast5	2374	8	32.91
yeast2vs4	822	8	9.28	yeast6	2374	8	41.39

The source codes in Matlab for the methods developed in this paper are available, together with the datasets, partitions and the results on the website associated with this paper[6].

The purpose of this section is three-fold. The first experiment is intended to test whether the empirical kernel map provides a more suitable space for over-sampling than the input space when dealing with kernel methods and analyses the effect of the number of dimensions chosen for over-sampling (i.e., the influence of the concentration of spectral properties). The second experimental subsection will complement the approach proposing a new kernel learning algorithm, to optimise a more flexible kernel function, which would ideally better fit the data. The purpose of this experiment is to test whether the kernel function chosen influences the results and how, optimising this kernel function the synthetic generated data will be better adapted to the classification problem. Finally, the third experiment focuses on the case of weighted or preferential over-sampling to analyse which patterns should be more prone to be over-sampled and test a new multiple kernel learning algorithm for optimising the generated patterns.

TABLE II contains information about all of the methods used for this three-fold experimentation and a brief summary (mean and standard deviation) of the mean results obtained along the 50 datasets used. Apart from the fact that the SVM without over-sampling performs poorly for minority classes, it can be seen that the standard deviation is very high in GM, indicating large fluctuations in the results. One can also see that the optimisation of a spherical Gaussian kernel by KTA (i.e., OSK) does not lead to very good results, and a better option is to use cross-validation instead or a more flexible

[6]http://www.uco.es/grupos/ayrna/efso

kernel, as the one used in OGK. Further information about the results will be extracted using statistical tests.

The complete results for all of the methods can be seen in the webpage associated to this paper[6], including the individual results for all the datasets. For the sake of comparison, we included the results obtained by a majority class rule (MCR) classifier as a baseline result (i.e., a naïve rule that classify all the patterns as belonging to the majority class). From the results of MCR it can be seen that Acc is not a suitable metric to take into account, since this trivial methodology achieves the best results in some cases (haberman, yeast05679vs4, glass016vs2, glass0146vs2, glass2, yeast4 and yeast6). In the following subsections, we will perform three differentiated statistical tests to validate the previously stated hypotheses.

TABLE II: Abbreviation for all the methods considered for the experimentation and mean and standard deviation results (Mean_{SD}) for all of the datasets.

Algorithm	$Acc(\%)$	$GM(\%)$
Majority class rule classifier (MCR)	$86.70_{0.53}$	$0.00_{0.00}$
SVM without over-sampling (SVM)	$\mathbf{93.35_{2.02}}$	$77.28_{12.07}$
SVM applying over-sampling in the input space (OIS)	$90.33_{3.11}$	$85.72_{8.50}$
SVM with over-sampling in the empirical feature space ($OEFS$)	$90.24_{3.50}$	$86.30_{8.14}$
SVM with over-sampling in the reduced empirical feature space ($OREFS$)	$90.41_{3.27}$	$86.83_{7.20}$
SVM with an optimised spherical kernel for over-sampling (OSK)	$\mathit{90.95_{3.16}}$	$80.45_{11.34}$
SVM with an optimised generalised kernel for over-sampling (OGK)	$89.45_{4.09}$	$\mathit{87.17_{6.86}}$
SVM with over-sampling via cross-validated preferential learning ($OCPL$)	$90.15_{3.35}$	$\mathbf{87.18_{6.74}}$
SVM with over-sampling via preferential multiple kernel learning ($OPMKL$)	$90.59_{3.45}$	$86.89_{7.20}$

The best method is in **bold** face and the second one in *italics*

A. First experiment: Over-sampling in the EFS

In this subsection, we will validate the hypothesis that the EFS is a more suitable space for over-sampling than the input space. Furthermore, we will test whether by optimising the dimensionality of this space the generated patterns are more adequate for the classification problem. To do so, we will test four different approaches: SVM, OIS, $OEFS$ and $OREFS$ (see TABLE II for the meaning of the acronyms).

As said before, we discarded all dimensions that correspond to zero eigenvalues for the computation of the EFS for $OEFS$. Furthermore, we performed a nested 5-fold cross-validation over the training sets of the number of dominant dimensions when considering over-sampling in the reduced EFS ($OREFS$). To do so, we considered the following values for the q value of the empirical kernel map Φ_q^e: $q \in \{\lfloor 0.1r \rfloor, \lfloor 0.25r \rfloor, \lfloor 0.5r \rfloor, \lfloor 0.75r \rfloor, r\}$, where r is the original rank of the training kernel matrix $\mathbf{K}$ and $\lfloor \cdot \rfloor$ is the floor function.

It can be seen that the results in GM for SVM are in general very poor (analyse for example the case of the haberman and glass2 datasets). Concerning the OIS method, it can be seen that in some cases the results of $OEFS$ are much better (analyse the result of the glass04vs5 dataset where SVM even obtained better results or the case of the glass016vs5 dataset). In relation to the effect of controlling the dimensionality, it can be seen that $OREFS$ generally yielded similar or better performance than $OEFS$ (see the result of the yeast2vs8 and led7digit02456789vs1 datasets, two examples which will be afterwards analysed). When taking Acc into account, it can be seen that the three over-sampling methods obtain very similar values (although $OEFS$ and $OREFS$ obtain better results in some cases, e.g. ecoli0267vs35).

TABLE III shows the test mean rankings (1 for the best method and 4 for the worst) for the methods considered in this experiment along all of the 50 datasets in terms of Acc and GM. The results show that SVM is the best performing method for Acc but the worst performing when considering a metric that takes into account the imbalanced nature of the data (GM). Furthermore, it is shown that both approaches for over-sampling in the EFS ($OEFS$ and $OREFS$) outperfomed the results obtained when over-sampling in the input space (OIS). Finally, it can be seen that controlling the EFS dimensionality we improve the results in most cases, as the $OREFS$ method obtained better mean results than $OEFS$.

To quantify whether a statistical difference exists among the algorithms, a procedure is employed to compare multiple classifiers in multiple datasets [43]. TABLE III also shows the result of applying the non-parametric statistical Friedman's test (for a significance level of $\alpha = 0.05$) to the mean Acc and GM rankings. The test rejects the null-hypothesis that all algorithms perform similarly in mean ranking for both metrics (note that for GM the differences are larger).

TABLE III: Mean ranking results for SVM, OIS, $OEFS$ and $OREFS$.

Ranking	SVM	OIS	$OEFS$	$OREFS$
Acc	**1.53**	3.21	2.74	*2.52*
GM	3.64	2.61	*1.96*	**1.79**
Friedman's test				
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 2.66)$				
F-value$_{Acc}$: $21.06 \notin C_0$, F-value$_{GM}$: $35.70 \notin C_0$				

On the basis of this rejection and following the guidelines of [43], we consider the best performing methods in GM (the two proposals, $OEFS$ and $OREFS$) as control methods for the post-hoc test and we compare them to the rest according to their rankings. It has been noted that the approach of comparing all classifiers to each other in a post-hoc test is not as sensitive as the approach of comparing all classifiers to a given classifier (control method). One approach to this latter type of comparison is the Holm's test. The test statistics for comparing the i-th and j-th method using this procedure is: $z = \frac{R_i - R_j}{\sqrt{\frac{k(k+1)}{6N}}}$, where k is the number of algorithms, N is the number of datasets and R_i is the mean ranking of the i-th method. The z value is used to find the corresponding probability from the table of normal distribution, which is compared with an appropriate level of significance α. Holm's test adjusts the α value in order to compensate for multiple comparisons. This is done in a step-up procedure that sequentially tests the hypotheses ordered by their significance. We will denote the ordered p-values by $p_1, p_2, \ldots, p_k$ so that $p_1 \leq p_2 \leq \ldots \leq p_k$. Holm's test compares each p_i with $\alpha^*_{\text{Holm}} = \alpha/(k-i)$, starting

from the most significant p value. If p_1 is below $\alpha/(k-1)$, the hypothesis is rejected and we compare p_2 with $\alpha/(k-2)$. If the second hypothesis is rejected, the test proceeds with the third, and so on. As soon as a certain null hypothesis cannot be rejected, all the remaining hypotheses are retained as well.

To analyse the results obtained from the Holm's test see TABLE IV. For the $OEFS$ method, the test concluded that there were statistically significant differences with SVM for Acc (note that in this case SVM obtained better results), SVM for GM and OIS for GM as well. This indicates that, although $OEFS$ obtained worst results for Acc in comparison with SVM, the results for GM are significantly better with comparison to SVM and OIS (therefore giving evidence that the EFS provides a more suitable space for over-sampling by convex combination of patterns). Concerning the $OREFS$ method, the same results are obtained, but there are also significant differences when considering the OIS method for Acc, which could indicate that over-sampling in the empirical feature space can be beneficial with other purposes, for example, for ensuring the class boundaries.

TABLE IV: Results of the Holm procedure using $OEFS$ and $OREFS$ as control methods (CMs) when compared to SVM and OIS: corrected α values, compared method and p-values, all of them ordered by the number of comparison (i).

CM: $OEFS$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.016	SVM	0.0000_{--}	SVM	0.0000_{++}
2	0.025	OIS	0.0687	OIS	0.0118_{++}
3	0.050	$OREFS$	0.3941	$OREFS$	0.5102
CM: $OREFS$		Acc		GM	
i	$\alpha^*_{0.05}$	Method	p_i	Method	p_i
1	0.016	SVM	0.0000_{--}	SVM	0.0000_{++}
2	0.025	OIS	0.0007_{++}	OIS	0.0014_{++}
3	0.050	$OEFS$	0.3941	$OEFS$	0.5102

Win ($++$) or lose ($--$) with statistical significant difference for $\alpha = 0.05$

In relation to the optimal dimensionality of the EFS, it can be said that the decay rate of the eigenvalues is related to the smoothness of the kernel and the number of necessary dimensions depends on the interplay between the kernel and the dataset. In this case, the mean value obtained from the cross-validation step for the number of dimensions was $(0.42 \pm 0.29)r$. More specifically, Fig. 7 shows the histogram of the optimal dimensionality of the EFS for all the datasets tested, where it can be seen that in most of the cases $\lfloor 0.5r \rfloor$ is enough to contain all the relevant information about the dataset.

As said before, one of the hypothesis for controlling the dimensionality of the EFS was that our over-sampling algorithm relies on distances computed in the EFS (for computing nearest neighbours and choosing which patterns to over-sample), distances which may bear less neighbourhood information as the EFS dimensionality increases. Fig. 8 shows the histogram of distances between pairs of patterns for different values of the dimensionality of the EFS ($\lfloor 0.1r \rfloor$ and $1r$) for two datasets where the $OREFS$ method obtained much better results than $OEFS$ and where this so-called spectral properties phenomenon [32] can be appreciated. Note that for the case of the yeast2vs8 dataset, using all of the dimensions ($1r$) corresponds to over-sampling in an almost randomly fashion

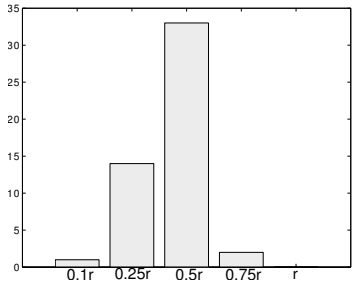

Fig. 7: Histogram of the mean optimal dimensionality of the EFS for all datasets. The abscissa axis represents the mean value, over the 30 results, for the rate of the rank of the kernel matrix. The ordinate axis shows the number of datasets where this value was selected from the cross-validation step.

as the k-nearest neighbours rule will not be very precise since most of the distances between pair of patterns are similar.

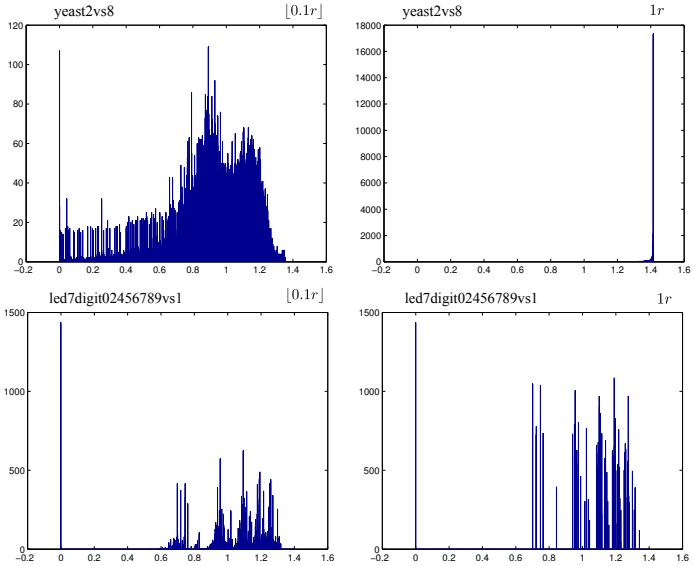

Fig. 8: Histogram of distances between pair of patterns for different dimensionality values of the EFS. The abscissa axis represents the distance between two patterns and the ordinate axis the occurrence of each distance.

From the results, several conclusions can be drawn. Firstly, over-sampling by convex combination is more suitable in an (ideally) linearly separable space such as the EFS. The method obtains better results in metrics that consider the imbalanced nature of the data without compromising the overall accuracy. However, over-sampling in the input space does not achieve this balance, indicating that a convex combination of patterns in a possibly nonlinearly separable space could generate patterns in unwanted areas. Concerning the optimisation of dominant dimensions for the feature space this methodology improves the results in some cases, thus encouraging further development of an analytical method to do so.

B. Second experiment: Influence of the kernel function

For this experiment we compare three different proposals: firstly, $OEFS$, which will be used as a baseline method to test

if the optimisation of the kernel function leads to better results, secondly, SVM with an optimised spherical Gaussian kernel (the same kernel than for $OEFS$ but optimised through KTA) for performing the over-sampling in the empirical feature space (OSK) and finally SVM with an optimised generalised Gaussian kernel in the empirical feature space (OGK).

In this work, the iRprop$^+$ algorithm is used to optimise the aforementioned centred KTA, because of its robustness [44]. The gradient norm stopping criterion was set to 10^{-5} and the maximum number of conjugate gradient steps to 10^2 [44]. For the optimisation of OGK, we also included a γ parameter as an additional parameter in the generalised Gaussian kernel, which will indeed make the parameters initialisation easier. The initial point for γ for all of the methods tested was chosen from the set $\{10^{-1}, 10^0, 10^1\}$, analysing the best result in alignment for the three values. The $\mathbf{Q}$ matrix for the generalised Gaussian kernel is initialised as the Moore-Penrose pseudoinverse of the covariance of the training points: $\mathbf{Q} = (cov(\mathbf{X}^{\text{tr}}))^+$, to address the problem of ill-conditioned covariance matrices. Once the kernel has been optimised via KTA, we optimise the C parameter using cross-validation within the values $\{10^{-3}, 10^{-2}, \ldots, 10^3\}$ (this two stage optimisation method is also referred in the literature as second-order method [45]).

From the results (that can be found in the website[6]) one can see that OSK and OGK obtained in some cases better results in Acc than SVM, this could be due to the application of the kernel optimisation through KTA, which selected a more optimal kernel than the cross-validation method. Analysing GM it can be seen that the performance of the spherical Gaussian kernel is not satisfactory. In optimising the spherical kernel, a cross-validation methodology should be preferred to KTA. To see this, analyse the case of the yeast0359vs78 and glass016vs2 datasets, where although OSK incorporates a over-sampling stage, SVM obtained better GM results. Finally, it can be seen that OGK yielded a much better performance in most of the cases (analyse the shuttle0vs4 dataset), demonstrating therefore that a more flexible kernel combined with kernel learning techniques could optimise the separation of the classes in the feature space, a necessary condition for over-sampling by convex combination of patterns.

As done before, TABLE V shows the mean ranking results for the three methods considered in this subsection and the result of applying the non-parametric Friedman's test (the test accepted the null-hypothesis that all of the algorithms perform similarly for Acc and rejected it for GM). From the results obtained it can be seen that when using a spherical Gaussian kernel, as in $OEFS$ (optimised through cross-validation) and OSK (optimised by KTA), the results are comparable and the methods obtain very similar mean ranking results. In this case, it is clear that the cross-validation method obtains better GM results as this is the metric used for the parameters selection stage. However, when using a more flexible kernel, such as the one considered in the OGK method, the results can be significantly improved. Note that applying cross-validation to the generalised kernel could possibly improve GM results, but the computational task required would be infeasible.

On the basis of Friedman's test rejection, the Holm test for multiple comparisons has been applied (see TABLE VI), and the test concluded that there were statistically significant differences for GM when considering OSK and $OEFS$. As said, there were no statistically significant differences for Acc.

TABLE V: Mean ranking results for $OEFS$, OSK and OGK.

Ranking	$OEFS$	OSK	OGK
Acc	*1.94*	**1.86**	2.20
GM	*2.15*	2.37	**1.48**
Friedman's test			
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 3.09)$			
F-value$_{Acc}$: $1.60 \in C_0$, F-value$_{GM}$: $13.41 \notin C_0$			

TABLE VI: Results of the Holm procedure using OGK as the control method when compared to OSK and $OEFS$: corrected α values, compared method and p-values, all of them ordered by the number of comparison (i).

CM: OGK		GM	
i	$\alpha^*_{0.05}$	Method	p_i
1	0.025	OSK	0.0000_{++}
2	0.050	$OEFS$	0.0008_{++}

Win ($++$) or lose ($--$) with statistical significant differences for $\alpha = 0.05$.

The results in this subsection show that over-sampling in the EFS is affected by the kernel function (although spherical Gaussian kernel has been proven to show promising results in the previous subsection), kernel selection/learning which is indeed a complex issue, shows (much) better results when employing a more flexible kernel such as the one used. Therefore, different kernel learning techniques could be explored in the future for the purpose of over-sampling in the EFS.

C. Third experiment: Preferential over-sampling

This experimental subsection is intended to test if there are patterns which are more suitable for over-sampling and if a general adaptive approach, yielding solutions based on unweighted over-sampling, borderline weighted over-sampling or '*safe*' level weighted over-sampling, could achieve better results than standard unweighted over-sampling. To do so, we compare $OEFS$ to two different approaches: the first one based on a cross-validation strategy ($OCPL$) and the second one based on kernel learning techniques ($OPMKL$).

As said before, to test this idea, we first obtain the spatial distribution of the patterns based on a cost-sensitive SVM hyperplane and we use a parametrised soft-max link function (Eq. (6)) to assign different probabilities of being over-sampled to the patterns according to this spatial distribution. This parametrisation is made using a β scale parameter, which will be optimised through cross-validation ($OCPL$) within a set of values and through kernel learning techniques ($OPMKL$). For the experiments, we select the set $\beta \in \{-5, -1, 0, 1, 5\}$.

Analysing the results obtained it can be seen that both $OCPL$ and $OPMKL$ obtain very competitive results both for Acc and GM. For some cases, the results obtained are equal since $OPMKL$ also includes the solutions of $OCPL$.

Once again, TABLE VII shows the mean ranking results when comparing these two approaches to the standard proposed technique $OEFS$. In this case, the Friedman's test accepted the null-hypothesis that the algorithms perform similarly for Acc and rejected it for GM. From these results, it can be seen that both methods outperform the standard proposal or at least yield similar performance (when considering Acc).

TABLE VII: Mean ranking results obtained by $OEFS$, $OCPL$ and $OPMKL$.

Ranking	$OEFS$	$OCPL$	$OPMKL$
Acc	*1.94*	2.13	**1.93**
GM	2.48	*1.93*	**1.59**
Friedman's test			
Confidence interval $C_0 = (0, F_{(\alpha=0.05)} = 3.09)$			
F-value$_{Acc}$: $0.63 \in C_0$, F-value$_{GM}$: $12.38 \notin C_0$			

The Holm's test for multiple comparisons has been also applied (see TABLE VIII). For both $OCPL$ and $OPMKL$, the test concluded that there are statistically significant differences for GM when compared to $OEFS$, indicating that preferential over-sampling is preferable over the uniform one [6]. Although the cross-validation strategy obtains very good results, the multiple kernel strategy yields slightly better performance (there are statistically significant differences for $\alpha = 0.10$).

TABLE VIII: Results of the Holm procedure using $OCPL$ and $OPMKL$ as control methods when compared to other state-of-the-art methods: corrected α values, compared method and p-values, ordered by the number of comparison (i).

CM: $OPMKL$		GM	
i	$\alpha^*_{0.05}$	Method	p_i
1	0.025	$OEFS$	0.0000_{++}
2	0.050	$OCPL$	0.0891_{+}
CM: $OCPL$		GM	
i	$\alpha^*_{0.05}$	Method	p_i
1	0.025	$OEFS$	0.0059_{++}
2	0.050	$OPMKL$	0.0891_{-}

Win $(++)$ or lose $(--)$ with statistical significant difference for $\alpha = 0.05$
Win $(+)$ or lose $(-)$ with statistical significant difference for $\alpha = 0.10$

To analyse the most appropriate region for over-sampling we analyse the optimal β values obtained from cross-validation (see Fig. 9 for the histogram). Recall that when $\beta < 0$, points within the minority class (in the feature space) are more likely to be picked; when $\beta > 0$ points on the class boundary or even on the other side of the hyperplane are preferred and when $\beta = 0$ all the points are equally likely to be chosen. It can be seen that for most datasets, over-sampling within "interior" of the minority class is preferable. Moreover, note that for a relatively large number of datasets the choice is uniform over-sampling. However, this could mean that uniform over-sampling could be feasible in some cases for over-sampling in the feature space (because of the improved data separation).

Finally, to study the computational cost of preferential over-sampling we included a small comparison of OEFS and OPMKL using only a data partition. The dataset chosen is `haberman`, and the time is reported in terms of seconds needed to over-sample the data. Cross-validation of parameters is not considered. According to this, the results are the following: 0.04 for OEFS and 0.33 for OPMKL. From these results, it can be seen that the computational time is affordable.

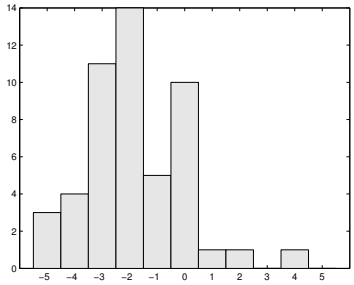

Fig. 9: Histogram of the mean values for the beta parameter used in the over-sampling process. The x coordinate represents the different mean β values chosen for each dataset (related to preferential over-sampling) and the y the number of datasets where the value was selected from the cross-validation process.

VII. Conclusions

This paper explores the notion of over-sampling in the feature space induced by a kernel function to deal with imbalanced classification problems. Since the feature space is not directly accessible, the empirical feature space is used (a Euclidean space that preserves the structure of the original feature space). Over-sampling is tackled by convex combination of patterns (as usually done in the state-of-the-art) and we focus on the paradigm of kernel methods. We explore the ideas of over-sampling in the full and reduced-rank empirical feature space, the optimisation of the feature space by kernel learning and the notion of preferential over-sampling which analyses which patterns should be more prone to be over-sampled. From the results of a thorough set of experiments over 50 imbalanced datasets, several conclusions can be drawn: firstly, over-sampling in the empirical feature space is seen to yield better performance than over-sampling in the input space; secondly, the control of the dimensionality of the empirical feature space could lead to better results; thirdly, the kernel used influence the solution to a great extent, making advisable the optimisation of the feature space structure (although the spherical Gaussian kernel has been shown to perform well for several cases); and finally, that there exist some regions of the dataset which should be preferred for over-sampling and that multiple kernel learning techniques should be explored in the future with the purpose of over-sampling.

The authors would also like to stress several lines of future work: Firstly, an analytical methodology for optimising the number of dominant dimensions of the empirical feature space could be developed with the purpose of over-sampling. Secondly, considering a unique methodology combining the techniques proposed in this paper could be accomplished, to analyse how these methods could complement each other. Furthermore, in the context of kernel learning, the over-sampling process could be incorporated in the kernel learning stage to search the more suitable representation for performing the over-sampling, not only the better class separation. Finally,

other intelligent optimisation techniques could be developed for the generation of the synthetic patterns.

REFERENCES

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, Sept. 2009.

[2] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002.

[4] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs modeling for highly imbalanced classification," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 39, pp. 281–288, Feb. 2009.

[5] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp. 4–21, 2011.

[6] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-Level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Bangkok, Thailand, April 2009, pp. 475–482.

[7] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proceedings of the International Joint Conference on Neural Networks*, Hong Kong, China, Dec. 2008, pp. 1322–1328.

[8] S. Barua, M. M. Islam, and K. Murase, "A novel synthetic minority oversampling technique for imbalanced data set learning," in *Proceedings of the International Conference on Neural Information Processing*, Shanghai, China, Nov. 2011, pp. 735–744.

[9] S. Barua, M. M. Islam, X. Yao, and K. Murase, "MWMOTE - majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 405–425, Feb. 2014.

[10] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, pp. 463–484, Jul. 2012.

[11] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *Proceedings of the International Joint Conference on Neural Networks*, Barcelona, Spain, July 2010, pp. 1–8.

[12] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, July 1992, pp. 144–152.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] Z.-Q. Zeng and J. Gao, "Improving SVM classification with imbalance data set," in *Proceedings of the International Conference on Neural Information Processing*, Bangkok, Thailand, Dec. 2009, pp. 389–398.

[15] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[16] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, Sept. 1999.

[17] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, pp. 460–474, March 2005.

[18] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel-target alignment," in *Proceedings of the International Conference on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2001, pp. 367–373.

[19] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 795–828, 2012.

[20] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proceedings of the European Conference on Machine Learning*, Pisa, Italy, Sept. 2004, pp. 39–50.

[21] Y. Liu, A. An, and X. Huang, "Boosting prediction accuracy on imbalanced datasets with SVM ensembles," in *Proceedings of the Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, Singapore, April 2006, pp. 107–118.

[22] P. Kang and S. Cho, "EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems," in *Proceedings of the International Conference on Neural Information Processing*, Hong Kong, China, Oct. 2006, pp. 837–846.

[23] X. Hong, S. Chen, and C. J. Harris, "A kernel-based two-class classifier for imbalanced data sets," *IEEE Transactions on Neural Networks*, vol. 18, pp. 28–41, Jan. 2007.

[24] G. Wu and E. Y. Chang, "KBA: Kernel Boundary Alignment Considering Imbalanced Data Distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 786–795, June 2005.

[25] ——, "Adaptive Feature-Space Conformal Transformation for Imbalanced Data Learning," in *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, USA, Aug. 2003, pp. 816–823.

[26] J. Yuan, J. Li, and B. Zhang, "Learning concepts from large scale imbalanced data sets using support cluster machines," in *Proceedings of the annual ACM international conference on Multimedia*, New York, USA, Oct. 2006, pp. 441–450.

[27] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, pp. 21– 27, Jan. 1967.

[28] L. W. Johnson and R. D. R. D. Riess, *Numerical analysis*. Reading, Mass. Addison-Wesley Pub. Co. c1982, 1982.

[29] J. T. Y. Kwok and I. W. H. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, Nov. 2004.

[30] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *Journal of Machine Learning Research*, vol. 9, no. 1, pp. 1875–1908, 2008.

[31] A. J. Smola, B. Schölkopf, and K.-R. Müller, "The connection between regularization operators and support vector kernels," *Neural Networks*, vol. 11, no. 4, pp. 637–649, Jun. 1998.

[32] M. Ledoux, *The Concentration of Measure Phenomenon*, ser. Mathematical surveys and monographs. American Mathematical Society, 2005.

[33] A. Giannopoulos and V. Milman, "Concentration property on probability spaces," *Advances in Mathematics*, vol. 156, no. 1, pp. 77 – 106, 2000.

[34] S. Abe and K. Onishi, "Sparse least squares support vector regressors trained in the reduced empirical feature space," in *Proceedings of the International Conference on Artificial neural networks*, San Sebastián, Spain, June 2007, pp. 527–536.

[35] H. Xiong, "A unified framework for kernelization: The empirical kernel feature space," in *Proceedings of the Chinese Conference on Pattern Recognition*, Nanjing, China, Nov. 2009, pp. 1 –5.

[36] M. Ramona, G. Richard, and B. David, "Multiclass feature selection with kernel gram-matrix-based criteria," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, pp. 1611–1623, Aug. 2012.

[37] G. R. G. Lanckriet, N. Cristianini, P. L. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[38] N. Srebro and S. Ben-David, "Learning bounds for support vector machines with learned kernels," in *Proceedings of the Annual Conference On Learning Theory*, Pittsburgh, USA, June 2006, pp. 169–183.

[39] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[40] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proceedings of the International Conference on Machine Learning*, Nashville, USA, July 1997, pp. 179–186.

[41] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems." *Pattern Recognition*, vol. 36, no. 3, pp. 849–851, 2003.

[42] M. Galar, A. Fernández, E. Barrenechea, and F. Herrera, "Eusboost: Enhancing Ensembles for Highly Imbalanced Data-sets by Evolutionary Undersampling," *Pattern Recognition*, vol. 46, no. 12, pp. 3460 – 3471, 2013.

[43] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[44] C. Igel and M. Hüsken, "Empirical evaluation of the improved rprop learning algorithms." *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[45] O. Chapelle and A. Rakotomamonjy, "Second order optimization of kernel parameters," in *In Proceedings of the International Conference on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2008.