P.Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler: Kernel Null Space Methods for Novelty Detection. CVPR 2013, pp. 3374–3381.

Kernel Null Space Methods for Novelty Detection

Paul Bodesheim[1], Alexander Freytag[1], Erik Rodner[1,2], Michael Kemmler[1], Joachim Denzler[1]
[1] Computer Vision Group, Friedrich Schiller University Jena
Ernst-Abbe-Platz 2, 07743 Jena, Germany
firstname.lastname@uni-jena.de
[2] UC Berkeley EECS, International Computer Science Institute, United States

Abstract

Detecting samples from previously unknown classes is a crucial task in object recognition, especially when dealing with real-world applications where the closed-world assumption does not hold. We present how to apply a null space method for novelty detection, which maps all training samples of one class to a single point. Beside the possibility of modeling a single class, we are able to treat multiple known classes jointly and to detect novelties for a set of classes with a single model. In contrast to modeling the support of each known class individually, our approach makes use of a projection in a joint subspace where training samples of all known classes have zero intra-class variance. This subspace is called the null space *of the training data. To decide about novelty of a test sample, our null space approach allows for solely relying on a distance measure instead of performing density estimation directly. Therefore, we derive a simple yet powerful method for multi-class novelty detection, an important problem not studied sufficiently so far. Our novelty detection approach is assessed in comprehensive multi-class experiments using the publicly available datasets Caltech-256 and ImageNet. The analysis reveals that our null space approach is perfectly suited for multi-class novelty detection since it outperforms all other methods.*

1. Introduction

Many of today's real-world applications deviate from the traditional assumption of pattern recognition that all relevant classes are known. Identifying samples from currently unknown classes is hence an essential step in visual object recognition. Instead of assuming a closed-world environment comprising a fixed number of classes, modern pattern recognition systems need to recognize outliers, identify anomalies, or discover entirely new classes that are currently not part of the assumed model. Especially the latter is essential in lifelong learning, where the system evolves over time and the number of object categories can grow. In this scenario, new objects should by detected automatically in order to learn object categories incrementally. Despite its importance, however, novelty detection is an often neglected part in visual recognition systems.

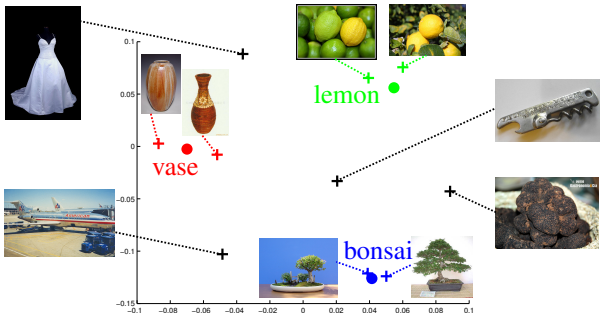


Figure 1. Novelty detection in the null space of a three-class-example: training samples of the known classes bonsai, lemon, and vase are projected to a single point, respectively (*colored dots*). Colored crosses indicate projections of test samples that belong to one of the known classes, black crosses are projections of samples from unknown classes which are considered as novelties.

The definition of novelty detection can be summarized as follows. Based on a fixed set of training samples from a fixed number of categories, novelty detection is a binary decision task to determine for each test sample whether it belongs to one of the known categories or not. A common assumption for novelty detection is that in feature space, samples occurring far away from the training data most likely belong to a new category.

However, we assume that objects of new categories occur far away from the training data *in the null space*. In this specific subspace, the training samples of each known category have zero intra-class variance, since they are projected to a single point, respectively. As a consequence, a whole class is represented as a single point and we can directly use distances between the projection of a test sample and the class representations to obtain a novelty measure. An example of our null space approach using three categories of the ImageNet dataset [3] is shown in Figure 1. In

the null space, test samples of known categories have small distances to the corresponding class representations. In contrast, test samples of unknown object categories are mapped far away. The difference between samples of the known categories and samples of novel ones is clearly observable in terms of distances to class representations.

Related work on novelty detection mainly focuses on modeling the distribution of a single class with arbitrary complex models (see Sect. 4). With our proposed approach, we circumvent the estimation of complex class distributions by totally removing the intra-class variances using null space projections. Furthermore, current multi-class approaches have to pool the scores of individual class models. This is nothing else but combining projections in different one-dimensional subspaces, which is a crucial step if the scores are not scaled properly. In contrast, our novelty detection approach yields a score obtained from a single subspace computed jointly for all known categories.

Therefore, the contribution of this paper is the following. We provide a method for novelty detection where we build on the Null Foley-Sammon transform (NFST) [7] due to its inherent properties explained later in this paper. With this transform, we are able to model *all* known training classes *jointly* and obtain a single novelty score for multiple classes allowing for joint multi-class novelty detection. We are not aware of any existing method that is able to perform *multi-class novelty detection with a single model*. Additionally, we show how to apply our method for one-class classification, where the training set only consists of samples from a single class.

The remainder of this paper is organized as follows. Since our approach is based on the theory of null spaces which is not widely-used in our community, we give a detailed review of null space methods and a kernelization strategy in Sect. 2 in order to make this paper self-contained. Our multi-class novelty detection approach as well as the derived one-class classification method using null space methods is explained in Sect. 3. An overview of related work on novelty detection is given in Sect. 4. Experimental results are presented in Sect. 5 showing the suitability of null space methods for multi-class novelty detection. A summary of our findings and suggestions for future research directions conclude the paper.

2. Reviewing null space methods

In the following, we review NFST in detail, since it lies at the core of our approach and is not widely-used so far. Our resulting novelty detection method based on null spaces is carried out in Sect. 3.

Generally, NFST allows for mapping input features $\boldsymbol{X} = \left[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right] \in \mathbb{R}^{D \times N}$ of C different classes to a representation with zero within-class scatter. This idea is visualized in Figure 2. NFST is limited to problems with small sample size (see Sect. 2.1). Thus, we also review the corresponding kernelized method, which is beneficial especially when using kernel functions with an infinite-dimensional reproducing kernel Hilbert space, *e.g.*, the Gaussian kernel.

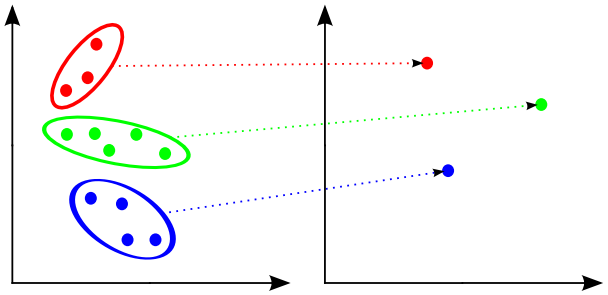


Figure 2. Visualization of NFST using three classes mapped from the input space (*left*) to the null space (*right*), adapted from [7].

2.1. Null Foley-Sammon transform

In the field of subspace methods, the linear discriminant analysis, also known as Fisher transform, is sometimes referred to as Foley-Sammon transform (FST) [4]. Our approach is based on a special case of this technique: the Null Foley-Sammon transform [7]. Generally, FST aims at computing discriminative features for multi-class data by maximizing the Fisher discriminant criterion:

$$\mathrm{J}\left(\boldsymbol{\varphi}\right) = \frac{\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_b \boldsymbol{\varphi}}{\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{\varphi}} \quad , \tag{1}$$

where $\boldsymbol{\varphi} \in \mathbb{R}^D$ is one direction in a discriminative subspace. Maximization of the Fisher criterion leads to simultaneously maximizing the between-class scatter using the between-class scatter matrix $\boldsymbol{S}_b$ and minimizing the within-class scatter using the within-class scatter matrix $\boldsymbol{S}_w$ [1]. Optimization of (1) can be done by solving the generalized eigenproblem:

$$\boldsymbol{S}_b \boldsymbol{\varphi} = \lambda \boldsymbol{S}_w \boldsymbol{\varphi} \quad . \tag{2}$$

The eigenvectors $\boldsymbol{\varphi}^{(1)}, \ldots, \boldsymbol{\varphi}^{(k)}$ according to the k largest eigenvalues $\lambda_1, \ldots, \lambda_k$ are collected as columns of a matrix $\boldsymbol{Q}$ and discriminative features of FST are computed by:

$$\tilde{\boldsymbol{x}}^{(i)} = \boldsymbol{Q}^{\mathsf{T}} \boldsymbol{x}^{(i)} \quad \forall\, i = 1, \ldots, N \quad . \tag{3}$$

For NFST in contrast to FST, each solution $\boldsymbol{\varphi}$ should have zero within-class scatter and positive between-class scatter:

$$\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{\varphi} = 0 \quad , \tag{4}$$

$$\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_b \, \boldsymbol{\varphi} > 0 \quad . \tag{5}$$

In [7], such a $\boldsymbol{\varphi}$ is called null projection direction. The constraints (4) and (5) lead to $\mathrm{J}\left(\boldsymbol{\varphi}\right) = \infty$ and thus to the best separability with respect to the Fisher discriminant criterion from Eq. (1). Only in the case of small sample size, *i.e.*,

$N \leq D$, and linear independent training data, it can be shown that one can compute $C-1$ null projection directions $\boldsymbol{\varphi}^{(1)}, \ldots, \boldsymbol{\varphi}^{(C-1)}$ with C being the number of classes [7]. This is due to the singularity of the within-class scatter matrix $\boldsymbol{S}_w$. If Eq. (4) holds, the inequality (5) boils down to:

$$\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_t \boldsymbol{\varphi} > 0 \ , \tag{6}$$

with $\boldsymbol{S}_t = \boldsymbol{S}_b + \boldsymbol{S}_w$ being the total scatter matrix. To additionally guarantee (6), we first define the null spaces of the matrices $\boldsymbol{S}_t$ and $\boldsymbol{S}_w$:

$$\boldsymbol{Z}_t = \{\boldsymbol{z} \in \mathbb{R}^D \mid \boldsymbol{S}_t \, \boldsymbol{z} = \boldsymbol{0}\} \quad , \tag{7}$$

$$\boldsymbol{Z}_w = \{\boldsymbol{z} \in \mathbb{R}^D \mid \boldsymbol{S}_w \boldsymbol{z} = \boldsymbol{0}\} \quad , \tag{8}$$

and denote their orthogonal complements as $\boldsymbol{Z}_t^{\perp}$ and $\boldsymbol{Z}_w^{\perp}$. With these definitions, it is easy to verify the correctness of the following statement:

$$\boldsymbol{\varphi} \in \left(\boldsymbol{Z}_t^{\perp} \cap \boldsymbol{Z}_w\right) \Rightarrow \left(\boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{\varphi} = 0 \wedge \boldsymbol{\varphi}^{\mathsf{T}} \boldsymbol{S}_b \boldsymbol{\varphi} > 0\right). \tag{9}$$

Therefore, we need to compute directions $\boldsymbol{\varphi}^{(1)}, \ldots, \boldsymbol{\varphi}^{(C-1)} \in \left(\boldsymbol{Z}_t^{\perp} \cap \boldsymbol{Z}_w\right)$. It can be shown [7] that $\boldsymbol{Z}_t^{\perp}$ is exactly the subspace spanned by zero-mean data $\boldsymbol{x}^{(1)} - \boldsymbol{\mu}, \ldots, \boldsymbol{x}^{(N)} - \boldsymbol{\mu}$ with $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}^{(i)}$. To ensure $\boldsymbol{\varphi} \in \boldsymbol{Z}_t^{\perp}$, we can represent each $\boldsymbol{\varphi}$ as:

$$\boldsymbol{\varphi} = \beta_1 \boldsymbol{b}^{(1)} + \ldots + \beta_n \boldsymbol{b}^{(n)} = \boldsymbol{B}\boldsymbol{\beta} \tag{10}$$

using an orthonormal basis $\boldsymbol{B} = \left[\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(n)}\right]$ for the zero-mean data with $n \leq N$. Such a basis can be obtained by Gram-Schmidt orthonormalization or standard PCA.

Replacing $\boldsymbol{\varphi}$ in (4) with its basis expansion (10), we need to compute $\boldsymbol{\beta}$ according to $\boldsymbol{\beta}^{\mathsf{T}} \left(\boldsymbol{B}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{B}\right) \boldsymbol{\beta} = 0$. This is equivalent to solving the eigenproblem:

$$\left(\boldsymbol{B}^{\mathsf{T}} \boldsymbol{S}_w \boldsymbol{B}\right) \boldsymbol{\beta} = \boldsymbol{0} \tag{11}$$

of size n, which is much smaller than the size of $\boldsymbol{S}_w$ for small sample size cases. Having solutions $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(C-1)}$ of problem (11), we can compute null projection directions $\boldsymbol{\varphi}^{(1)}, \ldots, \boldsymbol{\varphi}^{(C-1)}$ by (10), again collect them as columns of a matrix $\boldsymbol{Q}$ and calculate discriminative features of NFST using (3). Let $\boldsymbol{X}_w$ be the matrix consisting of column vectors $\bar{\boldsymbol{x}}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(c_i)}$ with $\boldsymbol{\mu}^{(c_i)}$ being the mean vector of all data points belonging to the class c_i of sample i. Now, we are able to write $\boldsymbol{S}_w = \frac{1}{N} \boldsymbol{X}_w \boldsymbol{X}_w^{\mathsf{T}}$. Thereby, Eq. (11) can be rewritten as:

$$\boldsymbol{H}\boldsymbol{H}^{\mathsf{T}}\boldsymbol{\beta} = \boldsymbol{0} \tag{12}$$

with $\boldsymbol{H} = \boldsymbol{B}^{\mathsf{T}} \boldsymbol{X}_w$ only consisting of inner products between basis vectors and data points corrected by their class mean. The use of inner products in $\boldsymbol{H}$ suggests a kernelized algorithm, which will be reviewed in the next section.

2.2. Kernel Null Foley-Sammon transform

A fundamental assumption for NFST is the small sample size, *i.e.*, the number of training samples N is smaller than their dimension D. To overcome this problem and to allow for more flexibility in the model, we can use the kernel trick and perform NFST in high-dimensional spaces. This leads to Kernel Null Foley-Sammon transform (KNFST) [12, 26] and we map features implicitly to a kernel feature space with a kernel function κ given by $\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \langle \Phi(\boldsymbol{x}^{(i)}), \Phi(\boldsymbol{x}^{(j)}) \rangle$. These pairwise inner products of mapped training data are collected in a kernel matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$. To incorporate kernels, we first note that an orthonormal basis of the subspace $\boldsymbol{Z}_t^{\perp}$ is needed, which can be calculated by Kernel PCA as done in [26].

The Kernel PCA algorithm requires the centered kernel matrix $\bar{\boldsymbol{K}} = (\boldsymbol{I} - \boldsymbol{1}_N) \, \boldsymbol{K} \, (\boldsymbol{I} - \boldsymbol{1}_N)$, where $\boldsymbol{I}$ is the $N \times N$ identity matrix and $\boldsymbol{1}_N$ is a $N \times N$ matrix with all entries equal to $\frac{1}{N}$. The elements of $\bar{\boldsymbol{K}}$ are considered to be pairwise inner products of zero-mean mapped data points $\bar{\Phi}(\boldsymbol{x}^{(i)}) = \Phi(\boldsymbol{x}^{(i)}) - \frac{1}{N} \sum_{j=1}^{N} \Phi(\boldsymbol{x}^{(j)})$. The eigendecomposition of $\bar{\boldsymbol{K}}$ is given by $\bar{\boldsymbol{K}} = \boldsymbol{V}\boldsymbol{E}\boldsymbol{V}^{\mathsf{T}}$ with $\boldsymbol{E}$ being the diagonal matrix containing $n \leq N$ non-zero eigenvalues and $\boldsymbol{V}$ containing the corresponding eigenvectors in its columns. The scaled eigenvectors $\tilde{\boldsymbol{V}} = \boldsymbol{V}\boldsymbol{E}^{-\frac{1}{2}}$ contain coefficients for the eigenbasis $\boldsymbol{B}$:

$$\boldsymbol{b}^{(j)} = \sum_{i=1}^{N} \tilde{v}_{ij} \bar{\Phi}\left(\boldsymbol{x}^{(i)}\right) \quad \forall \, j = 1, \ldots, n \quad . \tag{13}$$

However, the eigenbasis does not have to be calculated directly. Instead of (11), we can equivalently solve (12). Therefore, we just need to compute the matrix $\boldsymbol{H}$ using inner products with the eigenbasis $\boldsymbol{B}$, which leads to:

$$\boldsymbol{H} = \left((\boldsymbol{I} - \boldsymbol{1}_N) \, \tilde{\boldsymbol{V}}\right)^{\mathsf{T}} \boldsymbol{K} \, (\boldsymbol{I} - \boldsymbol{L}) \quad . \tag{14}$$

The normalization of the basis vector coefficients $\tilde{\boldsymbol{V}}$ using $(\boldsymbol{I} - \boldsymbol{1}_N)$ is necessary, since basis vectors in $\boldsymbol{B}$ are linear combinations of zero-mean mapped data points $\bar{\Phi}(\boldsymbol{x}^{(i)})$. Normalizing the kernel matrix $\boldsymbol{K}$ using $(\boldsymbol{I} - \boldsymbol{L})$ is due to the fact that $\boldsymbol{X}_w$ contains mapped data points corrected by their specific class mean. Without loss of generality we can assume that data points $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}$ are sorted according to their class labels, such that the rows and columns of $\boldsymbol{K}$ are ordered as well. In this case, $\boldsymbol{L}$ is a block diagonal matrix with block sizes equal to the number of data points N_c in each class $c \in \{1, \ldots, C\}$ and the value $\frac{1}{N_c}$ at each position. Using matrix $\boldsymbol{H}$ computed by (14), we are able to solve (12) and obtain solutions $\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(C-1)}$. Similar to (10), we calculate $C-1$ null projection directions $\boldsymbol{\varphi}^{(1)}, \ldots, \boldsymbol{\varphi}^{(C-1)}$ but using coefficients in $\tilde{\boldsymbol{V}}$. Therefore, the coefficients for null projection directions are:

$$\tilde{\boldsymbol{\varphi}}^{(j)} = \left((\boldsymbol{I} - \boldsymbol{1}_N) \, \tilde{\boldsymbol{V}}\right) \boldsymbol{\beta}^{(j)} \quad \forall \, j = 1, \ldots, C-1 \quad . \tag{15}$$

A data point $\boldsymbol{x}^*$ is mapped to $(\boldsymbol{k}_*^\mathsf{T}\tilde{\boldsymbol{\varphi}}^{(1)}, \ldots, \boldsymbol{k}_*^\mathsf{T}\tilde{\boldsymbol{\varphi}}^{(C-1)})^\mathsf{T}$ in the null space, where $\boldsymbol{k}_*$ contains values of the kernel function calculated between $\boldsymbol{x}^*$ and all N training samples.

2.3. Related approaches

Beside NFST and KNFST, there exist further null space approaches. The null eigenspace of principal component analysis is analyzed in [11] to select features for one-class SVM. In order to obtain the common features of a single class, the input data is projected on principal components associated with zero eigenvalue. However, it is not guaranteed that such principal components exist, especially in large-scale settings (which is in contrast to the null space of KNFST we use in our approach). Therefore, the author of [11] proposes to use the principal components associated with the small eigenvalues.

There also exists a metric learning approach [5] that is closely related to null space methods. The authors approximate an ideal case, where the metric assigns zero distance to samples of the same class and samples of different classes are infinitely far. The approximation is carried out by a probabilistic formulation together with a convex optimization problem. Again, it is not guaranteed that all samples of the same class are mapped to a single point, since the proposed algorithm only approximates the ideal case. However, combining such a "null space metric" with NFST or KNFST is an interesting topic for future work.

Another metric learning approach [17] has recently been introduced for large-scale image classification, which is able to generalize to new classes. The authors study how their learned metric deals with samples of unseen classes. However, they do not detect those samples of unknown classes automatically. To bridge this gap, we propose a novelty detection method in the next section.

3. Novelty detection with null space methods

In the previous section, we have described NFST and its kernelization based on already existing work [7, 12, 26]. This section explains how to adapt null space methods for novelty detection in both one-class and multi-class scenarios. Instead of creating a multi-class model out of several one-class models as done in previous work (see Sect. 4), we introduce our approach working the other way round. We therefore first show how to perform multi-class novelty detection and then apply this idea to the one-class case. Additionally, we characterize the advantages of our novelty detection approach that come from the model properties.

3.1. Multi-class novelty detection using null spaces

In multi-class novelty detection, we want to calculate a novelty score indicating whether a test sample belongs to one of C known classes, no matter to which class. Throughout the rest of this paper, we refer to the classes known during training as target classes. We calculate a null space of dimension $C-1$ and determine target points $\boldsymbol{t}^{(1)}, \ldots, \boldsymbol{t}^{(C)}$, one point for each target class, corresponding to the projection of class samples in the null space (see Sect. 2).

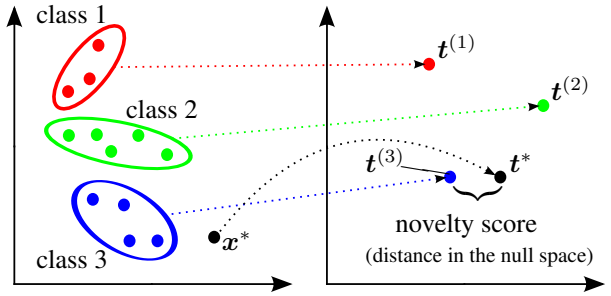


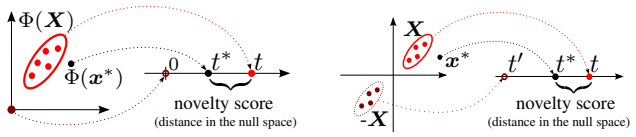
Figure 3. Overview of our *multi-class novelty detection* approach using projections in the joint null space: the novelty score of a test sample $\boldsymbol{x}^*$ is the smallest distance between its projection $\boldsymbol{t}^*$ and the class projections in the null space.

To obtain a single novelty score of a test sample $\boldsymbol{x}^*$, we first map $\boldsymbol{x}^*$ to $\boldsymbol{t}^*$ by projecting $\boldsymbol{x}^*$ into the null space. Applying a pooling step directly in the joint null space of all C classes, we use the smallest distance between $\boldsymbol{t}^*$ and the target points $\boldsymbol{t}^{(1)}, \ldots, \boldsymbol{t}^{(C)}$ as a novelty score (Figure 3):

$$\text{MultiClassNovelty}(\boldsymbol{x}^*) = \min_{1 \le i \le C} \text{dist}(\boldsymbol{t}^*, \boldsymbol{t}^{(i)}) \quad . \quad (16)$$

The larger the score and thus the minimum distance in the null space, the more novel is the test sample. A hard decision can be found by using a threshold between zero and the minimum distance between two target points. Note that an arbitrary distance measure can be incorporated and we use Euclidean distances in our experiments.

It is important to note that: (1) our null space approach is able to perform *joint learning* of multiple classes and *joint novelty detection* with a single model since it is derived from a true multi-class classification technique using a single subspace and (2) our null space approach is able to separate the known classes from *every currently unknown class* caused by the method specific properties. Training a binary SVM for each known class using the samples of the other classes as negatives only leads to separations from other known classes and not from currently unknown ones. In contrast, the separation from every currently unknown class is possible with our approach due to the simple class representations in the null space. Additionally, we are able to treat all classes jointly with their true class labels, while training a binary SVM for each known class treats remaining known classes as a single negative class, which contradicts to the idea of novelty detection. The novelty detection formulation of SVM [20] is only derived for one-class settings (see Sect. 4.1) and multi-class solutions based on this model are discussed in Sect. 4.2.



(a) Separation of the target class from the origin in the kernel feature space.

(b) Separation of the target class from negative data in the input space.

Figure 4. Our *one-class classification* approaches: all samples of the target class are mapped on a single point t in a one-dimensional subspace and the novelty score of a test sample $\boldsymbol{x}^*$ is the distance of its projection t^* to t.

3.2. One-class classification using null spaces

At first glance, one-class classification is not possible with null space methods, because we only have a single target class in a one-class setting. This leads to zero null projection directions, since the number of these directions is $C-1$ (see Sect. 2.1). Therefore, the adaptation to one-class classification settings is not straightforward.

Due to this reason, we propose separating the samples of the target class from the origin in the high-dimensional kernel feature space similar to one-class SVM [20]. Using this idea, we are able to compute a single null projection direction and all class samples are mapped on a single target value t along this direction. To check whether a test sample $\boldsymbol{x}^*$ belongs to the target class, we compute its projection on the null projection direction and obtain the value t^*. As a novelty score of $\boldsymbol{x}^*$ we propose using the absolute difference between t and t^*:

$$\text{OneClassNovelty}(\boldsymbol{x}^*) = |t - t^*| \tag{17}$$

similar to the multi-class case, where a large score indicates novelty. This score is a soft assignment and can be used in experimental evaluations. For practical applications, we obtain a hard decision using a threshold between 0 and $|t|$.

As an alternative strategy, we can also compute a single null projection direction by separating the class samples from "minus data". Following [19], all class samples are replicated with opposite sign to create a second class. Again, all true class samples are mapped on a single target value t and we compute the novelty score similar to our first approach using Eq. (17). Note that all points of the second class are mapped to a second value $t' \neq t$, which is of no interest for computing the novelty score. For practical applications, a threshold between 0 and $|t - t'|$ can be used to get a final decision.

Both one-class approaches are visualized in Figure 4. Their asymptotic runtime for learning is $\mathcal{O}(N^3)$ and thus equal to those of other kernel based methods, such as one-class Gaussian process techniques [8]. Computing the novelty score of a new sample can be done in linear time. However, separating from the origin in the kernel feature space is more suitable when dealing with histogram kernels like the histogram intersection kernel and we use this method for our experiments. In additional experiments with Gaussian kernels, both methods achieved comparable performance.

3.3. Advantages of our novelty detection approach

Our proposed novelty detection approach benefits from the null space, a joint subspace of all training samples where each known class is represented by a single point. In contrast to other subspace methods such as Kernel PCA, additional density estimation or clustering within the obtained subspace can be avoided and a simple distance measure can be applied to get a novelty score. Whereas a pooling step is necessary to combine scores of individual class models from different subspaces, *e.g.*, when applying the one-vs-rest SVM framework, null space methods offer the possibility to treat several classes in a joint manner with a single subspace model. Additionally, our approach separates known classes from every currently unknown class without the necessity of negative samples by using simple representations of known classes in the null space. This is in contrast to binary classifiers treating samples of one class as positives and samples of remaining known classes as negatives.

Using null space methods for novelty detection, we are able to calculate a single feature for each class of the target data and thus are able to compute *features with zero intra-class variance*. This ability is exactly what the authors in [23] claim to be needed for one-class classification. They suggest using features leading to zero variance in target data, if available. Such features are computable with null space methods, even for multiple classes.

In addition, computing features of zero variance within a class totally removes potentially large and complex intra-class variations of the training samples. This means nothing but extracting features that are identical within each class by determining the common properties of class samples. The transformed features obtained using null space methods can therefore be treated as *class-specific features*, since the transformation preserves the joint characteristics within each class. As previously mentioned, such features are perfectly suited for novelty detection from a theoretical point of view [23]. Additionally, this goes beyond the feature selection scheme proposed by [11] for one-class SVM already mentioned in Sect. 2.3, where it is proposed to use principal components associated with small eigenvalues in order to preserve the common features of a single class.

Furthermore, we benefit from the absence of additional hyperparameters such as the outlier ratio in the support vector based methods [20, 21] or the noise variance in the Gaussian process framework [8]. The only parameter that can occur within our approach is the one related to the kernel function, an issue shared by all kernel methods. Hence, additional parameter tuning beyond kernel hyperparameters is not necessary for our proposed novelty detection method.

4. Related work on novelty detection

An overview of basic concepts for novelty detection in signal processing is provided by the review papers of Markou and Singh [15, 16]. In visual object recognition, novelty detection should not be confused with the detection of unseen classes in zero shot learning [9], where knowledge about new objects is used explicitly, *e.g.*, via attributes.

Generally, novelty detection problems can be divided into one-class and multi-class settings depending on the number of known classes during training. Recent work on novelty detection focuses on one-class classification. However, the derived methods can also be used for the multi-class case if combined properly. In the following, we give a short overview of related work for both one-class and multi-class novelty detection scenarios.

4.1. One-class classification

The one-class classification paradigm assumes that the whole data stems from a single underlying class. One-class methods are particularly useful for latent binary classification problems, where only samples from a single class are available. These methods model the distribution of a single class similar to Parzen density estimation [1], where known samples are assumed to be located in high density regions.

An alternative strategy is to enclose class samples with a boundary and measure the distance to it, *e.g.*, the support vector data description (SVDD) [21] estimates the minimal enclosing hypersphere of a class, where the margin between class samples and outliers can additionally be maximized [25]. Note that one-class SVM (1SVM), which separates the samples of a single class from the origin with maximum margin, achieves results equivalent to SVDD when using a kernel that leads to constant self-similarities $\kappa(\boldsymbol{x}, \boldsymbol{x})$ [20]. Since we apply such kernels in our experiments, both methods achieve identical results (see Sect. 5).

The Gaussian process (GP) framework is a probabilistic methodology [18] and it has been shown that Gaussian process regression can be applied for one-class classification problems using the predictive mean (GP-Mean) and the predictive variance (GP-Var) as one-class scores [8].

4.2. Approaches for dealing with multiple classes

In many practical applications, not only a single class but a set of classes is given. For dealing with multiple classes, previous approaches can be divided in three main groups: (1) treating the training data of all given classes as one artificial super-class [10], (2) combining the results of several one-class classifiers learned with training data from each class separately [22], and (3) using the results of multi-class classifiers. The latter has not been studied so far and and we apply the one-vs-rest SVM framework as a baseline. In the following, we give more details about those groups.

Artificial super-class A simple way to perform multi-class novelty detection is to train a single one-class classifier for all available samples of all known categories [10]. This means nothing else but treating multiple classes as one large artificial super-class, which seems unsuitable for categories that are far away from each other in feature space.

Combination of one-class classifiers The authors of [22] propose training a multi-class classification model by combining one-class classifiers learned for each category and define the rejection of a test sample based on a pooling strategy for the individual one-class scores. In this setting, each one-class classifier discriminates a single class from each possible other class. This is in contrast to binary classifiers such as SVM that only discriminate between known classes, *e.g.*, in the one-vs-rest setting. Therefore, the rejection method used in [22] can be applied to novelty detection and we compare our approach to this strategy.

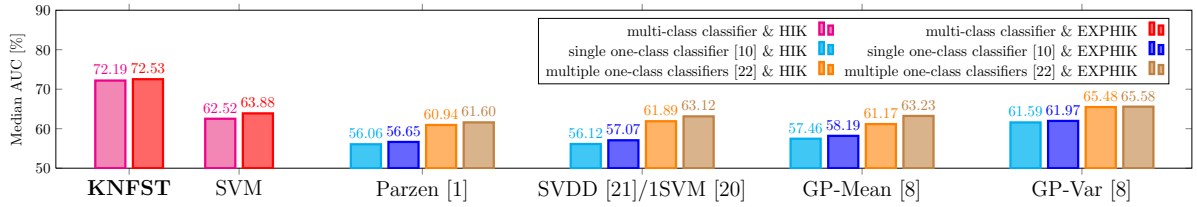
Multi-class classifiers Reject strategies for multi-class classification are related to novelty detection but differ in treating regions between classes. Samples could also be rejected when being close to the decision boundaries between known classes, which is obviously contradicting with the idea of novelty detection and which is a severe problem especially when classes overlap in feature space. Note that this is also the case when pooling binary SVM classifiers. Since these classifiers distinguish between a single class and a fixed set of negative classes, it is not clear how such models behave for samples of unseen categories, *i.e.*, whether such samples always occur in the "negative" half-spaces indicated by the SVM hyperplanes. However, we also compare our approach to the one-vs-rest SVM framework.

5. Experiments

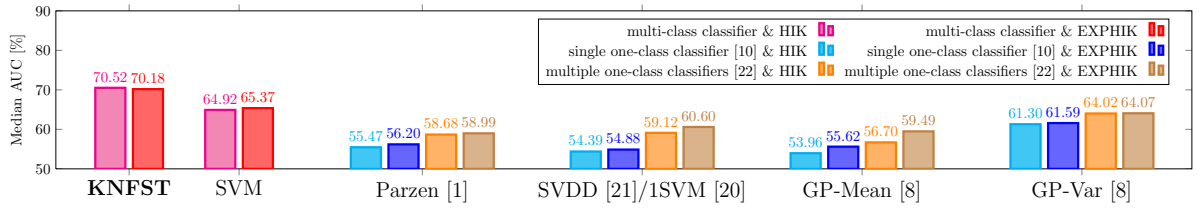
We evaluate our novelty detection approach[1] in visual object recognition on two datasets, Caltech-256 [6] and ImageNet [3]. For the latter, we choose exactly the same 1,000 object classes as done for ILSVRC 2010[2] using the training set (100 samples per class) for learning the models and the validation set (50 samples per class) for testing. We compare the performances of our approach to those of several state-of-the-art approaches reviewed in Sect. 4 using the area under the ROC curve (AUC). In the experiments, we focus on multi-class novelty detection. We also performed experiments in one-class classification treating each class of both datasets as a single target class once. The results of all methods are comparable and do not differ significantly. However, this is not the case when multiple classes are considered and we show that our null space approach outperforms all other methods in this typical scenario important for lifelong learning and automatic object discovery.

[1] MATLAB source code available at: http://www.inf-cv.uni-jena.de/Forschung/paperProjects/Kernel+Null+Space+Methods+for+Novelty+Detection.html

[2] http://www.image-net.org/challenges/LSVRC/2010



(a) Five object categories known during training.



(b) Ten object categories known during training.

Figure 5. Performance in multi-class novelty detection on the Caltech-256 dataset.

5.1. Experimental setup

Multi-class setup We use either five or ten target classes that are known during training. Samples of Caltech-256 target classes are split in training and test set of equal size and all samples not belonging to the target classes are considered as novelties. For each set of target classes, we average over 20 random splits. The experiments on the ImageNet dataset are done with 100 samples per target class for training and 50 samples from each of the 1,000 classes (including the target classes) for testing. Final results are achieved by computing median AUC scores using 50 and 100 randomly picked training sets of the Caltech-256 and ImageNet dataset, respectively.

Features To represent images, we use bag-of-visual-words histograms from densely sampled SIFT [13] features. These are publicly available for both datasets, Caltech-256[3] and ImageNet[4], which allows for easy reproducibility.

Kernels The similarity between two histograms is measured using the histogram intersection kernel (HIK) [14]: $\kappa_{\text{HIK}}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{d=1}^{D} \min(x_d, x'_d)$ or the corresponding generalized rbf-kernel [24]: $\kappa_{\text{EXPHIK}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(2 \cdot \kappa_{\text{HIK}}(\boldsymbol{x}, \boldsymbol{x}') - \kappa_{\text{HIK}}(\boldsymbol{x}, \boldsymbol{x}) - \kappa_{\text{HIK}}(\boldsymbol{x}', \boldsymbol{x}')\right)$.

Methods for comparison As one-class classifiers, we apply the methods reviewed in Sect. 4.1 using code provided by the corresponding authors. We compare our approach with previous work using either a single one-class classifier [10] or combine models separately trained for each class [22] (see Sect. 4.2). Additionally, we apply binary SVM in the one-vs-rest framework representing a standard technique in visual object recognition [14] using LIBSVM [2]. The outlier ratio ν of 1SVM and SVDD, the parameter C of binary SVM as well as the noise variance of the Gaussian process techniques is set to 0.1.

[3] http://homes.esat.kuleuven.be/~tuytelaa/unsup_features.html

[4] http://www.image-net.org/download-features

5.2. Multi-class novelty detection results

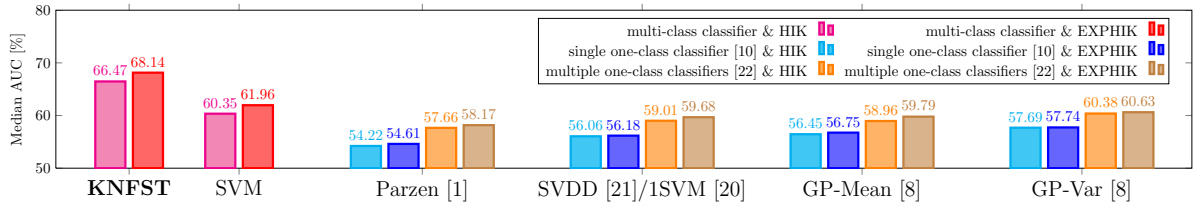
The results on the Caltech-256 dataset and the ImageNet dataset are shown in Figure 5 and Figure 6, respectively. First of all, we empirically verify that modeling multiple classes with a single one-class classifier as proposed in [10] is not appropriate, since learning individual one-class classifiers for each category according to [22] leads to better performances in all our experiments.

Interestingly, the binary SVM approach seems to be more suitable for the task of multi-class novelty detection in terms of higher median AUC scores compared to most approaches based on one-class classifiers. The GP-Var method combined with the pooling approach of [22] is the best among the one-class based methods.

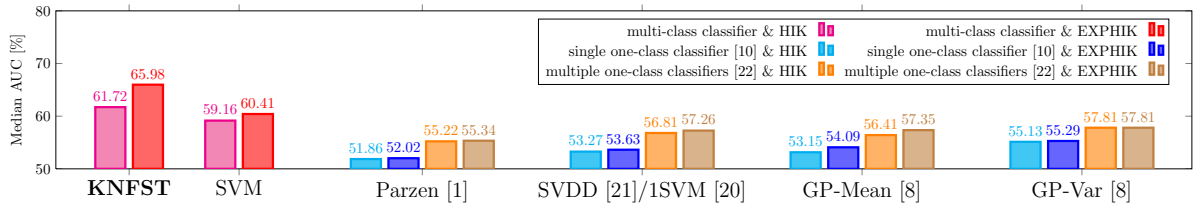
However, our proposed approach with *KNFST even outperforms SVM and GP-Var with a benefit of more than* 5% *in median AUC* on both the Caltech-256 dataset (see Figure 5) and the ImageNet dataset (see Figure 6). KNFST achieves significantly superior results in all our experiments with $p < 10^{-5}$ according to the Wilcoxon rank sum test. This highlights the capability and the relevance of our proposed null space approach for novelty detection.

6. Conclusions and future work

Multi-class novelty detection is a challenging problem, which needs more attention from the research community. Especially in real-world applications, where the number of categories is not fixed in advance, a modern object recognition system should cope with situations, where novel object categories can occur at any time. This paper proposes a new novelty detection approach based on null space projections, which is perfectly suitable for tackling this problem. The benefit of our proposed multi-class novelty detection approach is its ability of separating a set of known



(a) Five object categories known during training.



(b) Ten object categories known during training.

Figure 6. Performance in multi-class novelty detection on the ImageNet dataset.

classes from every currently unknown class. The approach is able to decide about novelty in a single step using a single model, whereas other approaches need to train a model for each known class without considering them jointly. Our experimental results clearly demonstrate the advantage of the joint learning approach using the null space leading to the best performance for multi-class novelty detection compared to all other methods. Additionally, we have addressed one-class classification as a special case of novelty detection. We have shown how to adapt our multi-class approach in order to be able to model a single target class.

Future work will concentrate on novelty detection in large-scale scenarios, where hundreds or thousands of categories are known to the model. Additionally, incorporating metric learning approaches related to null space methods such as [5, 17] is of special interest.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 2, 6

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *Trans. Intell. Syst. Technology*, 2(3):27:1–27:27, 2011. 7

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009. 1, 6

[4] D. H. Foley and J. W. Sammon. An optimal set of discriminant vectors. *IEEE Trans. Comput.*, C-24(3):281–289, 1975. 2

[5] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Proc. NIPS*, pages 451–458, 2005. 4, 8

[6] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 6

[7] Y.-F. Guo, L. Wu, H. Lu, Z. Feng, and X. Xue. Null foley-sammon transform. *Pattern Recog.*, 39(11):2248–2251, 2006. 2, 3, 4

[8] M. Kemmler, E. Rodner, and J. Denzler. One-class classification with gaussian processes. In *Proc. ACCV*, pages 489–500, 2010. 5, 6

[9] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proc. CVPR*, pages 951–958, 2009. 6

[10] T. Landgrebe, P. Paclik, D. M. J. Tax, and R. P. W. Duin. Optimising two-stage recognition systems. In *Multiple Classifier Systems*, pages 206–215, 2005. 6, 7

[11] H. Lian. On feature selection with principal component analysis for one-class svm. *Pattern Recog. Lett.*, 33(9):1027–1031, 2012. 4, 5

[12] Y. Lin, G. Gu, H. Liu, and J. Shen. Kernel null foley-sammon transform. In *Proc. Int. Conf. Comput. Sci. Software Eng.*, pages 981–984, 2008. 3, 4

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. 7

[14] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *Proc. CVPR*, pages 1–8, 2008. 7

[15] M. Markou and S. Singh. Novelty detection: A review-part 1: Statistical approaches. *Signal Process.*, 83(12):2481–2497, 2003. 6

[16] M. Markou and S. Singh. Novelty detection: A review-part 2: Neural network based approaches. *Signal Process.*, 83(12):2499–2521, 2003. 6

[17] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Proc. ECCV*, pages 488–501, 2012. 4, 8

[18] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 01 2006. 6

[19] V. Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4):942–960, 2006. 5

[20] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. 4, 5, 6

[21] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004. 5, 6

[22] D. M. J. Tax and R. P. W. Duin. Growing a multi-class classifier with a reject option. *Pattern Recog. Lett.*, 29(10):1565–1570, 2008. 6, 7

[23] D. M. J. Tax and K.-R. Müller. Feature extraction for one-class classification. In *Proc. ICANN/ICONIP*, pages 342–349, 2003. 5

[24] S. Vempati, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Generalized rbf feature maps for efficient detection. In *Proc. BMVC*, pages 2.1–2.11, 2010. 7

[25] M. Wu and J. Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(11):2088–2092, 2009. 6

[26] W. Zheng, L. Zhao, and C. Zou. Foley-sammon optimal discriminant vectors using kernel approach. *IEEE Trans. Neural Netw.*, 16:1–9, 2005. 3, 4