Novelty Discovery with Kernel Minimum Enclosing Balls

Rafet Sifa[1(✉)] and Christian Bauckhage[1,2]

[1] Fraunhofer Center for Machine Learning, Sankt Augustin, Germany
{rafet.sifa,christian.bauckhage}@iais.fraunhofer.de
[2] University of Bonn, Bonn, Germany

Abstract. We introduce the idea of utilizing ensembles of Kernel Minimum Enclosing Balls to detect novel datapoints. To this end, we propose a novelty scoring methodology that is based on combining outcomes of the corresponding characteristic functions of a set of fitted balls. We empirically evaluate our model by presenting experiments on synthetic as well as real world datasets.

1 Introduction

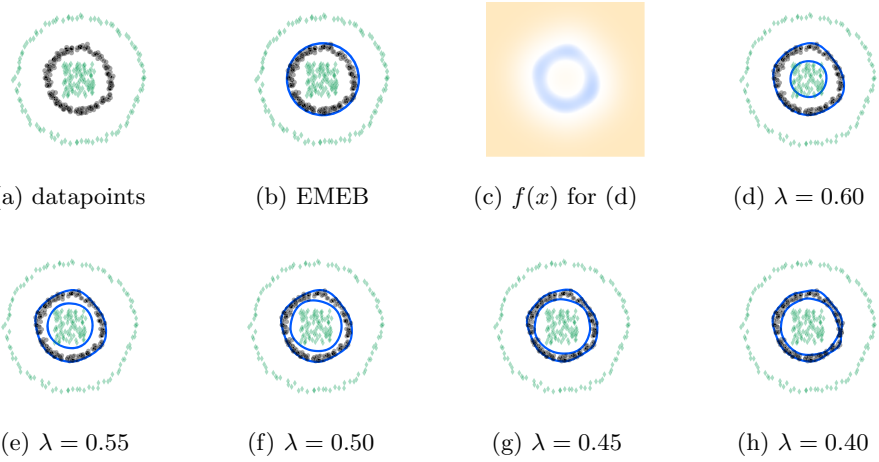
The notion of novelty discovery (or detection) [6] can be described as a one-class classification problem (a.k.a *data domain description* [11]) aiming to learn certain characteristics of the analyzed datasets to be able to separate novel datapoints. It finds many applications in numerous scientific and engineering areas such as fraudulent activity detection in financial applications or detecting rare events in medical monitoring [6]. Although reservoir computing based approaches [4] have been proposed to a variety of classification and regression problems, to the best of our knowledge, corresponding methods that are oriented to tackle one-class problems are scarce. The main contribution of this work is about utilizing Minimum Enclosing Balls [1] for novelty discovery. Minimum Enclosing Balls (MEBs) fall into the class of unsupervised representation learning methods that can be used to extract important characteristics about the considered datasets [1]. The main idea behind the MEBs is about determining the smallest ball encapsulating the *entire* dataset in the data- or feature space, which can be found by formulating the problem as an inequality constrained convex minimization problem with a dual allowing for invoking the kernel trick and this dual can be solved using dynamical processes from reservoir computing [1].

Our contribution is based on the decisions of a set of Kernel Minimum Enclosing Balls (KMEBs) by introducing a compound novelty score, which can allow for, for instance, a majority voting based detection as decision based on single balls might be limiting for novelty detection. In addition, our methodology

Supported by the Competence Center for Machine Learning Rhine Ruhr (ML2R) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01—S18038A).

I. S. Kotsireas and P. M. Pardalos (Eds.): LION 14 2020, LNCS 12096, pp. 414–420, 2020.
https://doi.org/10.1007/978-3-030-53552-0_37

can be easily implemented in neuromorphic architectures and is capable of dealing with nonlinear patterns due to kernelization [1,10]. Figure 1 shows an illustrative example explaining our idea about detecting the novel datapoints (green diamond shaped points in Fig. 1a) given a dataset of normal datapoints (black round points in Fig. 1a), which can neither be detected using euclidean Minimum Enclosing Balls (as seen in Fig. 1b) nor considering probabilistic novelty detection such as the deviation from the sample data mean [6]. Instead by considering the characteristic functions of multiple KMEBs (see example in Fig. 1c) with differently scaled Gaussian kernels we can detect all novel points that the considered balls might not individually be capable of capturing (compare the results of Fig. 1d to the others).



(a) datapoints (b) EMEB (c) $f(x)$ for (d) (d) $\lambda = 0.60$

(e) $\lambda = 0.55$ (f) $\lambda = 0.50$ (g) $\lambda = 0.45$ (h) $\lambda = 0.40$

Fig. 1. A conceptual example illustrating the idea of utilizing Kernel Minimum Enclosing Balls for novelty discovery. (a) shows the data (the inner ring), which is used to compute the ball, and the novel (green diamond) points. It is important to note that neither considering the deviation from the mean vector nor computing the euclidean MEB, which is shown in (b), can in this case isolate the points inside the inner ring. (c) shows a heat-map of the characteristic function from Eq. 7, where colors orange, white and blue respectively indicate positive, zero and negative values. (d–h) shows the dataset and the novel points with the decision boundaries for different Gaussian Kernel scale values λ. Used individually to detect the novel points, the recall values for detecting the novelty are respectively 0.935, 0.995 and 1.000 for the balls in (d), (e) and (f–h). We obtain 1.000 recall when majority-voting over the prediction of the balls (d–h) (i.e. by considering an ensemble of 5 KMEBs with evenly spaced λ values over $[0.4, 0.6]$). (Color figure online)

The remaining of the paper is organized as follows. So as to be self-contained, in Sect. 2 we will formally define the notion of KMEBs, show how we can compute them following a process akin to the ones used in echo state networks and finally show how, once computed, the support vectors of balls can be used to

characterize the interior of the fitted balls. Following that in Sect. 3 we will introduce a new novelty scoring methodology based on the characteristic functions for novelty discovery. In Sect. 4 we will present empirical results to evaluate our approach using real world datasets and in Sect. 5 we will conclude our work.

2 An Overview of Kernel Minimum Enclosing Balls

Given a set of m-dimensional data points $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ (for $\boldsymbol{x}_i \in \mathbb{R}^m$) that are grouped into a column data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$, we aim to find the m-ball $\mathcal{B}(\boldsymbol{c}, r)$ containing each of the given data points in $\mathcal{X}$, where $\boldsymbol{c} \in \mathbb{R}^m$ and $r \in \mathbb{R}$ are respectively the center and the radius of $\mathcal{B}$. Finding MEBs can be cast as a convex optimization problem

$$\begin{aligned} \boldsymbol{c}_*, r_* = \underset{\boldsymbol{c}, r}{\operatorname{argmin}} \quad & r^2 \\ \text{s.t.} \quad & \left\|\boldsymbol{x}_i - \boldsymbol{c}\right\|^2 - r^2 \leq 0 \qquad i \in [1, \ldots, n]. \end{aligned} \tag{1}$$

Upon evaluating the Lagrangian and the KKT conditions, the negated dual of (1), allows for the kernel trick (as the data *only* occurs in form of inner products [1]) and can be written as the minimization problem

$$\begin{aligned} \boldsymbol{\mu}_* = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \;\; & \boldsymbol{\mu}^\intercal \boldsymbol{K}\, \boldsymbol{\mu} - \boldsymbol{\mu}^\intercal \boldsymbol{k} \\ \text{s.t.} \quad & \sum_{i=1}^{n} \mu_i = 1 \quad \wedge \quad \mu_j \geq 0 \;\; \forall \;\; j \in [1, \ldots, n], \end{aligned} \tag{2}$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is a kernel matrix, $\boldsymbol{k}$ contains its diagonal (i.e. $\boldsymbol{k} = \operatorname{diag}[\boldsymbol{K}]$) and $\boldsymbol{\mu} \in \mathbb{R}^n$ contains Lagrange multipliers. The kernel matrix $\boldsymbol{K}$ in (2) is built by considering a Mercer kernel $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ such that $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. An example kernel function that we considered throughout our work is the Gaussian kernel that for scale parameter λ is defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\,\lambda^2}\right)$.

Considering (2), we note that finding Kernel Minimum Enclosing balls boils down to finding optimal $\boldsymbol{\mu}$, which resides in the standard simplex Δ^{n-1} and minimizes a convex function $\mathcal{L}(\boldsymbol{\mu}) = \boldsymbol{\mu}^\intercal \boldsymbol{k} - \boldsymbol{\mu}^\intercal \boldsymbol{K}\, \boldsymbol{\mu}$. Optimization settings of this kind can be easily solved iteratively using the Frank-Wolfe algorithm [3], which itself can be implemented as a recurrent neural network (see examples from [1,2,8,10]). To this end, at each iteration t, the Frank-Wolfe algorithm evaluates the gradient of the negated dual Lagrangian $\mathcal{L}(\boldsymbol{\mu})$ from (2), which amounts to $\nabla\mathcal{L}(\boldsymbol{\mu}) = 2\,\boldsymbol{K}\boldsymbol{\mu} - \boldsymbol{k}$, and finds the vertex of Δ^{n-1} for the update, that minimizes

$$\boldsymbol{\nu}_t = \underset{\boldsymbol{v}_j \in \mathbb{R}^n}{\operatorname{argmin}} \; \boldsymbol{v}_j^\intercal \left[2\,\boldsymbol{K}\boldsymbol{\mu}_t - \boldsymbol{k}\right] \approx \boldsymbol{g}_\beta\big(2\,\boldsymbol{K}\boldsymbol{\mu}_t - \boldsymbol{k}\big), \tag{3}$$

where $\boldsymbol{\nu}_t \in \mathbb{R}^n$ represent the current solution at t, $\boldsymbol{v}_j$ is the jth standard vector $\boldsymbol{v}_j = [\delta_{j1}, \delta_{j2}, \ldots, \delta_{jp}]^T\}$ (here δ_{ji} represents the Kronecker delta) and, finally,

$\boldsymbol{g}_\beta(\boldsymbol{x})$ represents the soft-min operator. This operator is the smooth approximation of argmin$_\cdot$, whose the ith entry defined as $\big(\boldsymbol{g}_\beta(\boldsymbol{x})\big)_i = \frac{e^{-\beta x_i}}{\sum_j e^{-\beta x_j}}$ and has the limit

$$\lim_{\beta\to\infty} \boldsymbol{g}_\beta(\boldsymbol{x}) = \underset{\boldsymbol{v}_j\in\mathbb{R}^n}{\operatorname{argmin}}\, \boldsymbol{v}_j^\intercal \boldsymbol{x} = \boldsymbol{v}_i. \tag{4}$$

Given that we can define the *convergent* iterative Frank-Wolfe updates [1] as

$$\boldsymbol{\mu}_{t+1} \leftarrow (1-\eta_t)\,\boldsymbol{\mu}_t + \eta_t\, \boldsymbol{g}_\beta\big(2\,\boldsymbol{K}\boldsymbol{\mu}_t - \boldsymbol{k}\big), \tag{5}$$

where $\eta_t \in [0,1]$ is a monotonically decreasing step size. Rearranging the rightmost expression in (5) as $\boldsymbol{g}_\beta\big(2\,\boldsymbol{K}\boldsymbol{\mu}_t - \boldsymbol{k}\big) = \boldsymbol{g}_\beta\big(2\,\boldsymbol{K}\boldsymbol{\mu}_t + \bar{\boldsymbol{K}}\bar{\mathbf{1}}\big)$, where $\bar{\boldsymbol{K}} = diag(\boldsymbol{k})$ and $\bar{\mathbf{1}}$ is the vector of -1s defined as $\bar{\mathbf{1}} = [-1,\ldots,-1]^T$, allows us to interpret and implement these updates in terms of echo state networks [4]. That is, we can describe this machinery as a structurally constrained echo state network, in which we have the fixed input vector $\bar{\mathbf{1}}$ containing -1s, the input weight matrix $\bar{\boldsymbol{K}}$, n reservoir neurons with $\boldsymbol{g}_\beta(\cdot)$ and $2\,\boldsymbol{K}$ respectively being the nonlinear activation function and the reservoir weight matrices and η_t acting as a leaking rate for updating the Lagrange multipliers. Once optimal Lagrange multipliers have been found using the updates from (5), we can determine the kernelized radius and the squared magnitude of the center of the fitted ball $\mathcal{B}$ respectively as $r_* = \sqrt{\boldsymbol{\mu}_*^\intercal \boldsymbol{k} - \boldsymbol{\mu}_*^\intercal \boldsymbol{K}\,\boldsymbol{\mu}_*}$ and $\boldsymbol{c}_*^\intercal \boldsymbol{c}_* = \boldsymbol{\mu}_*^\intercal \boldsymbol{K}\,\boldsymbol{\mu}_*$, which will allow us to define a characteristic function defining the interior of $\mathcal{B}$ [1]. Namely, using these equalities we can represent the inequality $\|\boldsymbol{x}-\boldsymbol{c}_*\|^2 \leq r_*^2$ to check whether an arbitrary point $\boldsymbol{x} \in \mathbb{R}^m$ within the ball $\mathcal{B}$ by considering

$$f(\boldsymbol{x}) = \sqrt{K(\boldsymbol{x},\boldsymbol{x}) - 2\,\bar{\boldsymbol{k}}^\intercal \boldsymbol{\mu}_* + \boldsymbol{\mu}_*^\intercal \boldsymbol{K}\,\boldsymbol{\mu}_*} - \sqrt{\boldsymbol{\mu}_*^\intercal \boldsymbol{k} - \boldsymbol{\mu}_*^\intercal \boldsymbol{K}\,\boldsymbol{\mu}_*}, \tag{6}$$

where $\bar{\boldsymbol{k}} \in \mathbb{R}^n$ is defined as $\bar{k}_i = K(\boldsymbol{x},\boldsymbol{x}_i)$ [1]. That is, $f(\boldsymbol{x}) > 0$ holds if $\boldsymbol{x}$ is outside of the ball $\mathcal{B}$, whereas, $f(\boldsymbol{x}) \leq 0$ is the case when $\boldsymbol{x}$ is inside the ball $\mathcal{B}$. Though, $f(\boldsymbol{x}) = 0$ only holds for the points with nonzero Lagrange multipliers that are the support vectors of $\mathcal{B}$ and can be defined as $\mathcal{S} = \{\boldsymbol{x}_i \mid \forall\ i \in [1,\ldots,n] \ \wedge \mu_{i*} > 0\}$. It is worth noting that, we can simplify (6) by grouping the $l \leq n$ points in $\mathcal{S}$ into a column data matrix $\boldsymbol{S} = [\boldsymbol{s}_1,\ldots,\boldsymbol{s}_l] \in \mathbb{R}^{m\times l}$, putting their corresponding multipliers in $\boldsymbol{\sigma} \in \mathbb{R}^l$, letting $\boldsymbol{Q} \in \mathbb{R}^{l\times l}$ be the kernel matrix for the support vectors (i.e. $Q_{ij} = K(\boldsymbol{s}_i,\boldsymbol{s}_j)$) and $\boldsymbol{q} \in \mathbb{R}^l$ to contain its diagonal (i.e. $\boldsymbol{q} = \text{diag}[\boldsymbol{Q}]$), which yields a simpler characteristic function

$$f(\boldsymbol{x}) = \sqrt{K(\boldsymbol{x},\boldsymbol{x}) - 2\,\bar{\boldsymbol{k}}^\intercal \boldsymbol{\sigma} + \boldsymbol{\sigma}^\intercal \boldsymbol{Q}\,\boldsymbol{\sigma}} - \sqrt{\boldsymbol{\sigma}^\intercal \boldsymbol{q} - \boldsymbol{\sigma}^\intercal \boldsymbol{Q}\,\boldsymbol{\sigma}} \tag{7}$$

where as in (6), $\bar{\boldsymbol{k}} \in \mathbb{R}^l$ is evaluated as $\bar{k}_j = K(\boldsymbol{x},\boldsymbol{s}_j)$ and we note that the term $\sqrt{\boldsymbol{\sigma}^\intercal \boldsymbol{q} - \boldsymbol{\sigma}^\intercal \boldsymbol{Q}\,\boldsymbol{\sigma}}$ (which indeed amounts to r_*) is does not depend on $\boldsymbol{x}$.

3 An Ensemble Approach for Novelty Discovery

Having explained how KMEBs are defined and can be computed so that we can determine their interior, we will now turn our attention to novelty discovery

by combining the characteristic functions of a set of balls. We note that, the characteristic function from (7) for a given ball $\mathcal{B}$ can be used to label the points outside of the ball to be the novel points. In this case a query point $\boldsymbol{x}$ is considered novel if $f(\boldsymbol{x}) > 0$ and not novel for $f(\boldsymbol{x}) \leq 0$. Although this approach can capture novel points it might result in very restrictive or too general decision boundaries that respectively might result in detecting every query point to be novel or not novel (see Fig. 1d for the latter case). Both problems, however, can be avoided if we generalize this approach by combining the decisions of multiple balls. One approach for such a combination can be based on uniform voting [7]. That is, given a set of u KMEBs $\mathcal{P} = \{\mathcal{B}_1, \ldots, \mathcal{B}_u\}$, that are trained considering a different setting, and $f_i(\cdot)$ and $[\![\cdot]\!]$ respectively indicating the characteristic function from (7) for ball $\mathcal{B}_i$ and the Iverson bracket, we can assign the *novelty score* of a query point $\boldsymbol{x}$ by evaluating $z(\boldsymbol{x}) = \sum_{i=1}^{u} [\![f_i(\boldsymbol{x}) > 0]\!]$ and, for instance, label $\boldsymbol{x}$ to be novel if $z(\boldsymbol{x}) \geq \lceil \frac{u}{2} \rceil$ (i.e. $\boldsymbol{x}$ is outside of the majority of the balls in $\mathcal{P}$ for an odd u) and not novel if $z(\boldsymbol{x}) < \lceil \frac{u}{2} \rceil$. In the next section, we will empirically evaluate this methodology to detect novelty by showing two conceptual examples on benchmarking datasets.

Table 1. Novelty prediction results in terms of recall (**RC**), precision (**PR**), as well as the harmonic mean and geometric mean of both (respectively referred as **F1** and **GM**) for (a) the CBLC Face and (b) the MNIST datasets to respectively detect non-face images from face images and the images of digit 0 from the ones of 1. We benchmarked methods to detect novelty that consider the deviation from the sample mean (**MDEV**), matrix factorization (**MF**), euclidean MEBs (**EMEB**) and the ensemble of kernelized MEBs (**EKMEB**). The superior prediction results indicate that EKMEB can indeed be used for novelty discovery.

Method	RC	PR	F1	GM
MDEV	0.686	1.000	0.813	0.828
MF	0.711	0.999	0.831	0.843
EMEB	0.790	0.999	0.882	0.889
EKMEB	0.974	0.998	0.986	0.986

(a) MNIST dataset

Method	RC	PR	F1	GM
MDEV	0.043	1.000	0.082	0.207
MF	0.215	0.998	0.354	0.463
EMEB	0.095	1.000	0.173	0.308
EKMEB	1.000	0.949	0.974	0.974

(b) CBLC Faces

4 Empirical Results

We evaluated our method on the MNIST [5] and CBCL-face (`bit.ly/2KwOVV6`) datasets. For the former we trained models on the digit 1 aiming to obtain the 0s, whereas for the latter we leaned balls on faces to detect non-face novel images. So as to evaluate the precision of the detections, we divided the training data into 90/10 splits and the latter split is combined with the novel points, which resulted in training/evaluation datsets of cardinality values 6067/6598 and 2186/4791 for respectively the MNIST and CBCL-face datasets. We note

that for both examples, we constructed ensembles of KMEBs (i.e. distinct $\mathcal{P}$ sets) with the Gaussian Kernel, whose scale values, in our case, were evenly spaced over specified intervals (as in Fig. 1) by considering $u = 5$ KMEBs with λ ranging in $[40, 60]$. We also normalized the datasets to have zero mean and unit variance and always considered $\beta = \infty$ for the softmin function (see (4)).

In Table 1, we compare our method against thresholding the tested points considering the maximum deviation from the sample mean vector [6], euclidean MEBs [1] (where we consider points outside of the ball as novel) and matrix factorization (MF) [8] based reconstruction to validate the use of kernel methods. For the first method, we label points in the test set as novel if the euclidean distance is larger than the furthest point to the sample mean. For the last method, we factorize the matrix with the number of latent factors $k = 50$ using the alternating least squares method [9] and learn a threshold value based on the *worst* reconstruction error ($l2$-norm). Unseen points with reconstruction error exceeding this threshold are considered novel. Table 1a and Table 1b respectively depict the prediction results for the MNIST and CBCL datasets, where we observe the superiority of ensemble KMEBs to detect novel datapoints.

5 Conclusion and Future Work

In this work, we introduced the idea of using ensemble of KEMBs for novelty discovery. We showed how we can construct ensembles of KEMBs and introduced a voting-based approach to detect novel data points. Our empirical evaluation yielded superior results over the use of mean deviation, euclidean MEBs and matrix factorization approaches. Our future work involves studying different ball selection as well as novelty determination strategies and extending the scope of the applications. Another line of future work is related to physical implementation of our methodology and in resource-constrained devices for applications in industrial domains such as for predictive maintenance.

References

1. Bauckhage, C., Sifa, R., Dong, T.: Prototypes within minimum enclosing balls. In: Tetko, I.V., Kůrková, V., Karpov, P., Theis, F. (eds.) ICANN 2019. LNCS, vol. 11731, pp. 365–376. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30493-5_36
2. Bauckhage, C.: A neural network implementation of Frank-Wolfe optimization. In: Lintas, A., Rovetta, S., Verschure, P.F.M.J., Villa, A.E.P. (eds.) ICANN 2017. LNCS, vol. 10613, pp. 219–226. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68600-4_26
3. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Res. **3**, 95–110 (1956)
4. Jäger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science **304**(5667), 78–80 (2004)
5. LeCun, Y., Boottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

6. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. Sig. Process. **99**, 215–249 (2014)
7. Rokach, L.: Ensemble methods for classifiers. In: Maimon, O., Rokach, L. (eds.) Data mining and Knowledge Discovery Handbook, pp. 957–980. Springer, Boston (2005). https://doi.org/10.1007/0-387-25465-X_45
8. Sifa, R.: An overview of Frank-Wolfe optimization for stochasticity constrained interpretable matrix and tensor factorization. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11140, pp. 369–379. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01421-6_36
9. Sifa, R.: Matrix and Tensor Factorization for Profiling Player Behavior. LeanPub, British Columbia (2019)
10. Sifa, R., Paurat, D., Trabold, D., Bauckhage, C.: Simple recurrent neural networks for support vector machine training. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11141, pp. 13–22. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01424-7_2
11. Tax, D.M., Duin, R.P.: Data domain description using support vectors. In: Proceedings of ESANN (1999)