Highlights

- two sparsification methods for indefinite learning models are proposed
- a formulation of an indefinite core vector regression is derived
- sparsification is evaluated on supervised classification and regression problems
Sparsification of core set models in non-metric supervised learning

Frank-Michael Schleif\textsuperscript{a,b,**}, Christoph Raab\textsuperscript{a}, Peter Tino\textsuperscript{b}

\textsuperscript{a}University of Appl. Sc. Würzburg-Schweinfurt, Dept. of Computer Science, Sanderheinrichsleitenweg 20, 97074 Würzburg, Germany
\textsuperscript{b}University of Birmingham School of Computer Science, Edgbaston, B15 2TT, Birmingham, UK

\textbf{ABSTRACT}

Supervised learning employing positive semi definite kernels has gained wide attraction and lead to a variety of successful machine learning approaches. The restriction to positive semi definite kernels and a hilbert space is common to simplify the mathematical derivations of the respective learning methods, but is also limiting because more recent research indicates that non-metric, and therefore non positive semi definite, data representations are often more effective. This challenge is addressed by multiple approaches and recently dedicated algorithms for so called indefinite learning have been proposed. Along this line, the Krėin space Support Vector Machine (KSVM) and variants are very efficient classifiers for indefinite learning problems, but with a non-sparse decision function. This very dense decision function prevents practical applications due to a costly out of sample extension. We focus on this problem and provide two post processing techniques to sparsify models as obtained by a Krėin space SVM approach. In particular we consider the indefinite Core Vector Machine and indefinite Core Vector Regression Machine which are both efficient for psd kernels, but suffer from the same dense decision function, if the Krėin space approach is used. We evaluate the influence of different levels of sparsity and employ a Nyström approach to address large scale problems. Experiments show that our algorithm is similar efficient as the non-sparse Krėin space Support Vector Machine but with substantially lower costs, such that also problems of larger scale can be processed.

\textcopyright 2019 Elsevier Ltd. All rights reserved.

Learning of classification models for indefinite kernels received substantial interest with the advent of domain specific similarity measures. Indefinite kernels are a severe problem for most kernel based learning algorithms because classical mathematical assumptions such as positive definiteness, used in the underlying optimization frameworks are violated. As a consequence e.g. the classical Support Vector Machine (SVM) (Vapnik, 2000) has no longer a convex solution - in fact, most standard solvers will not even converge for this problem (Loosli et al., 2016). Researchers in the field of e.g. psychology (Hodgetts and Hahn, 2012), vision (Scheirer et al., 2014; Xu et al., 2011) and machine learning (Duin and Pekalska, 2010) have criticized the typical restriction to metric similarity measures. In (Duin and Pekalska, 2010) it is shown that many real life problems are better addressed by e.g. kernel functions which are not restricted to be based on a metric. Non-metric measures (leading to kernels which are not positive semi-definite (non-psd)) are common in many disciplines. The use of divergence measures (Schnitzer et al., 2012; Zhang et al., 2009) is very popular for spectral data analysis in chemistry, geo- and medical sciences (Mwebaze et al., 2010; van der Meer, 2006), and are in general not metric. Also the popular Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) algorithm provides a non-metric alignment score which is often used as a proximity measure between two one-dimensional functions of different length. In image processing and shape retrieval indefinite proximities are often obtained by means of the inner distance (Ling and Jacobs, 2007) - another non-metric measure. Further prominent examples for genuine non-metric proximity measures can be found in the field of bioinformatics where classical sequence alignment algorithms (e.g. smith-waterman score (Gusfield, 1997)) produce non-metric proximity values. Multiple authors argue that the non-metric part of the data contains valuable information and should not be removed (Scheirer et al., 2014; Pekalska and Duin, 2005). Furthermore, it has been shown (Loosli et al., 2016; Schleif and Tiño, 2015) that work-arounds such as eigenspectrum modifications are often inappropriate or undesirable, due to a loss of information and problems with the out-of sample extension. Nevertheless they are still often used and can serve as a baseline approach. Due
Table 1: Overview of the different datasets. We provide the dataset size (N) and the origin of the indefiniteness. For vectorial data the indefiniteness is caused artificial by using the tanh kernel.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#samples</th>
<th>Proximity measure and data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonatas</td>
<td>1068</td>
<td>normalized compression distance on midi files (Schleif and Tiño, 2015)</td>
</tr>
<tr>
<td>Delft</td>
<td>1500</td>
<td>dynamic time warping (Schleif and Tiño, 2015)</td>
</tr>
<tr>
<td>a1a</td>
<td>1605</td>
<td>tanh kernel (Luss and d’Aspremont, 2009)</td>
</tr>
<tr>
<td>zongker</td>
<td>2000</td>
<td>template matching on handwritten digits (Pekalska and Haasdonk, 2009)</td>
</tr>
<tr>
<td>prodom</td>
<td>2604</td>
<td>pairwise structural alignment on proteins (Pekalska and Haasdonk, 2009)</td>
</tr>
<tr>
<td>PolydistH57</td>
<td>4000</td>
<td>Hausdorff distance (Pekalska and Haasdonk, 2009)</td>
</tr>
<tr>
<td>chromo</td>
<td>4200</td>
<td>edit distance on chromosomes (Pekalska and Haasdonk, 2009)</td>
</tr>
<tr>
<td>Mushrooms</td>
<td>8124</td>
<td>tanh kernel (Srisuphab and Mitrpanont, 2009)</td>
</tr>
<tr>
<td>swiss-10k</td>
<td>≈10k</td>
<td>smith waterman alignment on protein sequences (Schleif and Tiño, 2015)</td>
</tr>
<tr>
<td>checker-100k</td>
<td>100.000</td>
<td>tanh kernel (indefinite)</td>
</tr>
<tr>
<td>skin</td>
<td>245.057</td>
<td>tanh kernel (indefinite)(UCI, 2016)</td>
</tr>
<tr>
<td>checker</td>
<td>1 Mill</td>
<td>tanh kernel (indefinite)</td>
</tr>
</tbody>
</table>

1. Learning with non-psd kernels

Learning with non-psd kernels can be a challenging problem and may occur very quickly as discussed before, if domain specific measure are used or simply due to noise. The metric violations cause negative eigenvalues in the eigenspectrum of the kernel matrix $K$, leading to non-psd similarity matrices or indefinite kernels. Many learning algorithms are based on kernel formulations which have to be symmetric and psd. The mathematical meaning of a kernel is the inner product in some Hilbert space (Shawe-Taylor and Cristianini, 2004). However, it is often loosely considered simply as a pairwise “similarity” measure between data items, leading to a similarity matrix $S$.

If a particular learning algorithm requires the use of Mercer kernels and the similarity measure does not fulfill the kernel conditions, steps must be taken to ensure a valid model.

1.1. Eigenspectrum approaches

A natural way to address the indefiniteness problem and to obtain a psd similarity matrix is to correct the eigenspectrum of the original similarity matrix $S$. Popular strategies include flipping, clipping and shift correction. The non-psd similarity matrix $S$ is decomposed by an eigen decomposition: $S = U\Lambda U^\top$, where $U$ contains the eigenvectors of $S$ and $\Lambda$ contains the corresponding eigenvalues. One can now adapt the eigenvalues to get rid of the negative eigenvalues and to end up with a psd kernel.

Clip eigenvalue correction.: All negative eigenvalues in $\Lambda$ are set to 0. Spectrum clip leads to the nearest psd matrix $S$ in terms of the Frobenius norm (Higham, 1988).

Flip eigenvalue correction.: All negative eigenvalues in $\Lambda$ are set to $\lambda_i := |\lambda_i| \forall i$ which at least keeps the absolute values of the negative eigenvalues and can be relevant if these eigenvalue contain important information (Pekalska and Duin, 2005).

Shift eigenvalue correction.: The shift operation was already discussed earlier by different researchers (Filippone, 2009) and modifies $\Lambda$ such that $\Lambda := \Lambda - \min_i \lambda_i$. Spectrum shift enhances all the self-similarities by the amount of $\nu$ and does not change the similarity between any two different data points, but it may also increase the intrinsic dimensionality of the data space and amplify noise contributions.

In the experiments we will only compare with the clip and flip approach. The latter one is also an algorithmic part of the Krėin space SVM model. If one of the former corrections is
applied to the input kernel any standard kernel based learning method like SVM can be used. One major drawback of these approaches is the rather complicated out of sample extension to new test points but also that the data representation may have changed completely, leading to inferior results.

1.2. Krēin space SVM

The Krēin Space SVM (K SVM) (Loosli et al., 2016), replaced the classical SVM minimization problem by a stabilization problem in the Krēin space. The respective equivalence between the stabilization problem and a standard convex optimization problem was shown in (Loosli et al., 2016). Let \( x_i \in X, i \in \{1, \ldots, N\} \) be training points in the input space \( X \), with labels \( y_i \in \{-1, 1\} \), representing the class of each point. The input space \( X \) is often considered to be \( \mathbb{R}^d \), but can be any suitable space due to the kernel trick. For a given positive \( C \), SVM is the minimum of the following regularized empirical risk functional

\[
J_C(f, b) = \min_{f \in H, b \in \mathbb{R}^2} \frac{1}{2} \|f\|_H^2 + C \cdot H(f, b) \tag{1}
\]

Using the solution of Equation (1) as \( (f^*_C, b^*_C) \) one can introduce \( \tau = H(f^*_C, b^*_C) \) and the respective convex quadratic program (QP)

\[
\min_{f \in H, b \in \mathbb{R}^2} \frac{1}{2} \|f\|_H^2 \text{ s.t. } \sum_{i=1}^N \max(0, 1 - y_i(f(x_i) + b)) \leq \tau \tag{2}
\]

where we detail the notation in the following. This QP can be interpreted as a projection which still holds in RKKS and is used as a regularization technique (Loosli et al., 2016). This allows to define SVM in RKKS (viewed as Hilbert space) as the orthogonal projection of the null element onto the set \( \{f \in H | H(f, b) \leq \tau \} \)

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

If \( H_+ \) and \( H_- \) are reproducing kernel hilbert spaces (RKHS), \( K \) is a reproducing kernel Krēin space (RKKS). For details on RKHS and RKKS see e.g. (Pekalska and Dün, 2005).

In this case the uniqueness of the functional decomposition (the nature of the RKHSs \( H_+ \) and \( H_- \)) is not guaranteed. In (Ong et al., 2004) the reproducing property is shown for a RKKS \( K \). There is a unique symmetric kernel \( k(x, y) \) with \( k(x, y) \in K \) such that the reproducing property holds for all \( f \in K, f(x) = \langle f, k(x) \rangle_K \) and \( k = k_+ - k_- \). where \( k_+ \) and \( k_- \) are the reproducing kernels of the RKHSs \( H_+ \) and \( H_- \).

As shown in (Ong et al., 2004) for any symmetric non-positive kernel \( k \) that can be decomposed as the difference of two positive kernels \( k_+ \) and \( k_- \), a RKKS can be associated to it. In (Loosli et al., 2016) it was shown how the classical SVM problem can be reformulated by means of a stabilization problem. This is necessary because a classical norm as used in Eq. (2) does not exist in the RKKS but instead the norm is reinterted as a projection which still holds in RKKS and is used as a regularization technique (Loosli et al., 2016). This allows to define SVM in RKKS (viewed as Hilbert space) as the orthogonal projection of the null element onto the set \( \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \)

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]

where \( \partial_b \) denotes the sub differential with respect to \( b \). The set \( S \) leads to a unique solution for SVM in a Krēin space (Loosli et al., 2016). As detailed in (Loosli et al., 2016) one finally obtains a stabilization problem which allows one to formulate a SVM in a Krēin space.

\[
S = \{f \in K, b \in \mathbb{R} | H(f, b) \leq \tau \} \quad \text{and} \quad 0 \in \partial_b H(f, b)
\]
It is well-known that this problem is NP hard in general, and a variety of approximate solution strategies exist in the literature. Here, we rely on a popular and very efficient approximation offered by orthogonal matching pursuit (OMP) (Geoffrey M. Davis, 1994; Pati et al., 1993). Given an acceptable error $\epsilon > 0$ or a maximum number $n$ of non-vanishing components of the approximation, a greedy approach is taken: the algorithm iteratively determines the most relevant direction and the optimum coefficient for this axes to minimize the remaining residual error.

**Algorithm 2 OMP to approximate the $\alpha$ vector.**

1. **OMP:**
2. $I := \emptyset$; $r := y := K\tilde{\alpha}$; % initial residuum
3. while $|I| < n$ do
4. $I_0 := \text{argmax}_{|I|} |[Kr]|$; % find relevant direction + index
5. $I := I \cup \{I_0\}$ % track relevant indices
6. $\tilde{\gamma} := (K_r)^+ \cdot y$ % restricted (inverse) projection
7. $r := y - (K_r) \cdot \tilde{\gamma}$ % residuum of the approximated decision function
8. end while
9. return $\tilde{\gamma}$ (as the new sparse $\tilde{\alpha}$)

In line 2 of Alg. 2 we define the initial residuum to be the vector $K\tilde{\alpha}$ as part of the decision function. In line 4 we identify the most contributing dimension (assuming an empirical feature space representation of our kernel - it becomes the dictionary). Then in line 6 we find the current approximation of the sparse $\tilde{\alpha}$-vector - called $\tilde{\gamma}$ to avoid confusion, where $^+$ indicates the pseudo inverse. In line 7 we update the residuum by removing the approximated $K\tilde{\alpha}$ from the original unapproximated one. A Nyström based approximation of the Algorithm 2 is straightforward using the concepts provided in (Gisbrecht and Schleif, 2015; Schleif and Gisbrecht, 2013). There it is also shown that the Nyström approximation holds for non-psd kernels, with a simplified proof given in (Oglic and Gärtner, 2019). With the Nyström technique a symmetric matrix psvd (Williams and Seeger, 2000) or non-pdf (Gisbrecht and Schleif, 2015; Schleif and Gisbrecht, 2013) is approximated by a low-rank approach using a subset of the original datapoints, called landmarks. As shown in (Williams and Seeger, 2000; Gisbrecht and Schleif, 2015) this approximation is exact if the rank of original data is smaller or equal to the number of landmarks. The landmarks are often chosen randomly, with more advanced strategies proposed e.g. in (Musco and Musco, 2017).

### 2.2. Sparsification of iCVM by late subsampling

The parameters $\tilde{\alpha}$ are dense as already noticed in (Loosli et al., 2016). A naive sparsification by using only $\tilde{\alpha}$, with large absolute magnitude is not possible as can be easily checked by counter examples. One may now approximate $\tilde{\alpha}$ by using the (for this scenario slightly modified) OMP algorithm from the former section or by the following strategy, both compared in the experiments.

As a second sparsification strategy we can use the approach suggested by (Schleif and Tiño, 2017), to restrict the projection operator and hence the transformation matrix of iCVM to

**Algorithm 1 Krèin Space SVM (KSV) - adapted from (Loosli et al., 2016).**

**Krèin SVM:**

\[ [U, D] := \text{EigenDecomposition}(K) \]
\[ \tilde{K} := USD^\top \text{ with } S := \text{sign}(D) \]
\[ [\alpha, b] := \text{SVMSolver}(\tilde{K}, Y, C) \]
\[ \tilde{\alpha} := US^\top \alpha \quad \text{(now } \tilde{\alpha} \text{ is dense)} \]
\[ \text{return } \tilde{\alpha}, b; \]

contains the eigenvectors, $D$ is a diagonal matrix of the eigenvalues and $S$ is a matrix containing only \{1, -1\} on the diagonal as obtained from the respective function sign.

As pointed out in (Loosli et al., 2016), this solver produces an exact solution for the stabilization problem. The main weakness of this Algorithm is, that it requires the user to pre-compute the whole kernel matrix and to decompose it into eigenvectors/eigenvalues. Further, today’s SVM solvers have a theoretical, worst case complexity of $\approx O(N^2)$. The other point to mention is that the final solution $\tilde{\alpha}$ is not sparse. The iCVM from (Schleif and Tiño, 2017) has a similar derivation and leads to a related decision function, again with a dense $\tilde{\alpha}$, but the model fitting costs are $\approx O(N)$. 

### 2. Sparsification of iCVM

#### 2.1. Sparsification of iCVM by OMP

We can formalize the objective to approximate the decision function, which is defined by the $\tilde{\alpha}$ vector, obtained by KSVM or iCVM (both are structural identical), by a sparse alternative with the following mathematical problem:

\[ \min \|\tilde{\alpha}\|_0 \quad \text{s.t. } \sum_m \tilde{\alpha}_m \Phi(x_m)\top \Phi(x) \approx f(x) \]

Fig. 1: Prediction results for the protein dataset using a varying level of sparsity and the OMP sparsity methods. For comparison the prediction accuracy of the non-sparse model is shown by a straight line.

to the $\alpha$ parameters obtained from the stabilization problem in the Krèin space, which can be solved using classical optimization tools on the flipped kernel matrix. This permits to apply the obtained model from the Krèin space directly on the non-positive input kernel without any further modifications. The algorithm is shown in Alg. 1. There are four major steps: 1) an eigen-decomposition of the full kernel matrix, with cubic costs (which can be potentially restricted to a few dominating eigenvalues - referred to as KSVM-L); 2) a flipping operation; 3) the solution of an SVM solver on the modified input matrix; 4) the application of the projection operator obtained from the eigen-decomposition on the $\alpha$ vector of the SVM model. $U$ in Alg. 1 contains the eigenvectors, $D$ is a diagonal matrix of the eigenvalues and $S$ is a matrix containing only \{1, -1\} on the diagonal as obtained from the respective function sign.

As pointed out in (Loosli et al., 2016), this solver produces an exact solution for the stabilization problem. The main weakness of this Algorithm is, that it requires the user to pre-compute the whole kernel matrix and to decompose it into eigenvectors/eigenvalues. Further, today’s SVM solvers have a theoretical, worst case complexity of $\approx O(N^2)$. The other point to mention is that the final solution $\tilde{\alpha}$ is not sparse. The iCVM from (Schleif and Tiño, 2017) has a similar derivation and leads to a related decision function, again with a dense $\tilde{\alpha}$, but the model fitting costs are $\approx O(N)$. 

### 2. Sparsification of iCVM

#### 2.1. Sparsification of iCVM by OMP

We can formalize the objective to approximate the decision function, which is defined by the $\tilde{\alpha}$ vector, obtained by KSVM or iCVM (both are structural identical), by a sparse alternative with the following mathematical problem:

\[ \min \|\tilde{\alpha}\|_0 \quad \text{s.t. } \sum_m \tilde{\alpha}_m \Phi(x_m)\top \Phi(x) \approx f(x) \]
a subset of the original training data. We refer to this approach as iCVM-sparse-sub.

To get a consistent solution we have to recalculate parts of the eigen-decomposition as shown in Alg. 3. To obtain the respective subset of the training data we use the samples which are core vectors\(^2\). The number of core vectors is guaranteed to be very small (Tsang et al., 2006) and hence even for a larger number of classes the solution remains widely sparse. The suggested approach is given in Alg. 3. We assume that the original projection function (line 6 of Algorithm 3, detailed in (Loosli et al., 2016), is smooth and can be potentially restricted to a small number of construction points with low error. We observed that in general few construction points are sufficient to keep high accuracy, as seen in the experiments.

3. Indefinite Core-Vector-Regression - iCVR

As already indicated in (Schleif and Tiño, 2017) the Kr"ein space approach considered before can also be used in similar minimum enclosing ball (MEB) based optimization problems. In particular we will consider the sparsification in the context of core vector regression for indefinite kernels, subsequently referred to as iCVR.

Assume points \(x_i \in \mathbb{R}^d, i \in [1, \ldots, N]\) and real-valued outputs \(y_i \in \mathbb{R}\) are given. Further, we assume a kernel function \(k\) (for the moment it is assumed this kernel is a psd kernel) is given with a feature map \(\Phi\). A kernel regression trains a function of the following form: \(x \mapsto w^T \Phi(x) + b\), where \(w\) is a normal vector of a decision plane and \(b\) a bias term. The training objective is to get as many points as possible approximately right while preserving a large margin. In the classical core vector regression (CVR) (Tsang et al., 2006) an \(\epsilon\)-tube formalization is used to achieve this objective. For an \(\epsilon\)-tube a data point is correctly predicted iff its image is within \(\epsilon\) of the desired value. The corresponding dual core vector regression is described by (for details see (Tsang et al., 2006)):

\[
\max_{\alpha, \alpha' \geq 0, \alpha + \alpha' = 1} -\frac{1}{2} \sum_{i,j} (\alpha_i - \alpha'_j)(\alpha_j - \alpha'_j)(k_{ij} + 1)
- \frac{1}{2} \sum_{i=1}^{N} \alpha_i^2/C - \frac{1}{2} \sum_{i=1}^{m} (\alpha'_j)^2/C + \sum_{i=1}^{N} (\alpha_i - \alpha'_j)y_i
\]

This problem is of the form:

\[
\max_{\alpha, \alpha' \geq 0, \alpha + \alpha' = 1} -\frac{1}{2} \left(\alpha^T\right)^T \tilde{K} \left(\alpha^T\right) + \left(\alpha^T\right)^T \left(\begin{array}{c} y \end{array}\right)
\]

where

\[
\tilde{K} = \left(\begin{array}{cc}
(K + 11^T) + C \cdot I & -(K + 11^T) \\
-(K + 11^T) & (K + 11^T) + C \cdot I
\end{array}\right)
\]

\(\tilde{K}\) is a valid kernel and as shown in (Tsang et al., 2006) a core / MEB algorithm can be used to solve Eq. (4). If the underlying kernel function \(k\) is indefinite also \(\tilde{K}\) becomes indefinite. We can now use the same argumentation as for iCVM (Schleif and Tiño, 2017) and following the work in (Loosli et al., 2016) to modify the kernel \(\tilde{K}\) by a flipping operation, to calculate a valid CVR model. Using the projection approach of (Loosli et al., 2016) the obtained solution vector can again be mapped into the Kr"ein space to obtain a model for iCVR. This final solution does not need any kernel modification for new test points to be applied. The whole algorithmic workflow to derive a iCVR model is described in Algorithm 4.

4. Experiments - iCVM

This part contains a series of experiments that show that our approach leads to a substantially lower complexity, while keeping similar prediction accuracy compared to the non-sparse approach. To allow for large datasets with two much hassle we provide sparse results only for the MEB approaches, namely iCVM and iCVR. The modified OMP approach will work also for sparse KSVBM or KSVR but the late sampling sparsification is not well suited if many support vectors are given in the original model, asking for a sparse SVM implementation. We follow the experimental design given in (Loosli et al., 2016). Methods that require to modify test data are excluded as also done in (Loosli et al., 2016). Finally we compare the experimental complexity of the different solvers. The used data are explained in Table 1. Additional larger data sets have been added to motivate our approach in the line of learning with large scale indefinite kernels.

\(^2\)A similar strategy for KSVBM may be possible but is much more complicated because typically quite many points are support vectors and special sparse SVM solvers would be necessary.
4.1. Experimental setting

For each dataset, we have run 20 times the following procedure: a random split to produce a training and a testing set, a 5-fold cross validation to tune each parameter (the number of parameters depending on the method) on the training set, and the evaluation on the testing set. If \( N > 1000 \) we use \( m = 200 \) randomly chosen landmarks from the given classes to approximate the kernel matrix using the Nyström technique. If the input data are vectorial data we used a tanh kernel with parameters \( [\alpha = 1, r = 1] \) to obtain an indefinite kernel. Where tanh is given as: \( k(x,y) = \tanh(\alpha < x, y> + r) \).

4.2. Results

In Table 2 we show the results for large scale data (having at least 1000 points) using iCVM with sparsification. We observe much smaller models, especially for larger datasets with often comparable prediction accuracy with respect to the non-sparse model. The runtimes are similar to the non-sparse case but in general slightly higher due to the extra eigen-decompositions on a reduce set of the data as shown in Algorithm 3. But the focus is not on a faster runtime (which is linear for iCVM and iCVR), but on a simple, sparse model and hence an easy out of sample extension. A typical result for the protein data set using the OMP-sparsity technique and various values for sparsity is shown in Figure 1.

5. Experiments - iCVR

We show the effectiveness of iCVR on a number of simulated and real life benchmark regression problems and compare with solutions as obtained by using standard CVR but for flipped (all signs of the eigenspectrum become positive) and clipped eigenspectra (negative eigenvalues are set to 0) of the respective kernel matrices. Data are given as \( X \in \mathbb{R}^{|D|} \). Target function values \( y_i \in \mathbb{R} \). The following one-dimensional simulated datasets have been used:

- (SIM1) basic sinc sample, with 200 samples, \( f(x) = \sin(x/\pi) + 0.05 \cdot \sigma \) where \( \sigma \) is gaussian noise and \( x \) is linearly spread in \([-30,30]\)
- (SIM2) Friedman function, with 200 samples, \( f(x) = 10 \cdot \sin(\pi \cdot \sigma_1 \cdot \sigma_2) + 20 \cdot (\sigma_3 - 0.5)^2 + 10 \cdot \sigma_4 + 5 \cdot \sigma_5 + \sigma_7 \) and uniform noise \( \sigma_1, \ldots, \sigma_5, \sigma_7 \) is gaussian noise
- (SIM3) The Mackey glass data, with 12000 samples, in 1 dimension as detailed in (Mackey and Glass, 1977)

Further we used the following real life regression datasets.

- (DS1) Abalone - age prediction, with 4177 samples, \( D = 8 \) taken from (Lichman, 2013)
- (DS2) Forest fires, with 517 samples, \( D = 13 \), dimension 13 was used as output variable, taken from (Cortez and Morais, 2007)
- (DS3) Breast cancer (radius) prediction, with 569 samples, \( D = 32 \), dimension 3 was used as output variable, taken from (Lichman, 2013) (wdbc)
- (DS4) White wine quality (scored 0-10), with 4898 samples, \( D = 12 \), dimension 12 was used as output variable, taken from (Cortez et al., 2009)\textsuperscript{3}

The indefiniteness was caused using a Manhattan kernel \( K_m = -||X - X^\top|| \). The regression profiles for SIM1-SIM3 are depicted in Figure 2. In the experiments we apply the iCVR approach on the given datasets and compare it with the standard CVR algorithm were the indefinite input kernel was corrected by applying a flip or clip eigenspectrum transformation.

In Figure 3 a plot of the output function for SIM2 and its prediction using iCVR and CVR on a clipped kernel is shown. The plot shows substantial prediction errors on the clipped kernel in contrast to the prediction of iCVR with the indefinite mahalanobis kernel. Considering the results shown in Table 3 we observe that the clipping is in general worse than flipping or the iCVR. The sparse models of iCVR are in general only slightly worse than the non-sparse model. In parts we can even see a better performance of the sparse iCVR model (see last column) compared to the iCVM. This may be due to a denoising effect, caused by the implicit low rank approach used in OMP.

It should be noted that an application of the standard CVR on the indefinite kernels is not possible, which was also experimentally verified, because the obtained problem becomes non-convex and the solver is unable to provide a solution to the optimization problem.

6. Complexity analysis

The original KSVM has runtime costs (with full eigen-decomposition) of \( O(N^3) \) and memory storage \( O(N^2) \), where \( N \) is the number of points. The iCVM or respectively iCVR involves an extra Nyström approximation of the kernel matrix to obtain \( K_{(N,m)} \) and \( K^{-1}_{(m,m)} \), if not already given. If we have \( m \) landmarks, \( m \ll N \), this gives memory costs of \( O(mN) \) for the first matrix and \( O(m^2) \) for the second, due to the matrix inversion. Further a Nyström approximated eigendecomposition has to be done to apply the eigenspectrum flipping operator. This leads to runtime costs of \( O(N \times m^2) \). The runtime costs for the sparse iCVM/iCVR are \( O(N \times m^2) \) and the memory complexity is the same as for iCVM/iCVR. Due to the used Nyström approximation the prior costs only hold if \( m \ll N \), which is the case for many datasets as shown in the experiments.

The application of a new point to a KSVM, iCVM or iCVR model requires the calculation of kernel similarities to all \( N \) training points, for the sparse iCVM/iCVR this holds only in the worst case. In general the sparse iCVM/iCVR provides a simpler out of sample extension as shown in Table 2, but is data dependent.

The (i)CVM/(i)CVR model generation has not more than \( N \) iterations or even a constant number of 59 points, if the probabilistic sampling trick is used (Tsang et al., 2006; Smola and Schölkopf, 2000). As show in (Tsang et al., 2006) the classical SVM has runtime costs of \( O(1/\epsilon^2) \). The evaluation of a kernel function using the Nyström approximated kernel can be done

\textsuperscript{3}Available at: http://www3.dsi.uminho.pt/pcortez/wine/
Table 2: Prediction errors (mean ± std.-dev.) on the test sets. The percentage of projection points (pts) is calculated using the unique set over core vectors over all classes in comparison to all training points. All sparse-OMP models use only 10 points in the final models. Best results are shown in bold. Best sparse results are underlined. Datasets with substantially reduced prediction accuracy are marked by ⊙ (anova p < 5%).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>iCVM (sparse-sub)</th>
<th>pts</th>
<th>iCVM (sparse-OMP)</th>
<th>iCVM (non-sparse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Šonatas</td>
<td>12.64 ± 1.71</td>
<td>76.84%</td>
<td>22.56 ± 4.16⊙</td>
<td>13.01 ± 3.82</td>
</tr>
<tr>
<td>Delft</td>
<td>16.53 ± 2.79⊙</td>
<td>52.48%</td>
<td>3.27 ± 0.6</td>
<td>3.20 ± 0.84</td>
</tr>
<tr>
<td>a1a</td>
<td>39.50 ± 2.88⊙</td>
<td>1.25%</td>
<td>27.85 ± 2.8</td>
<td>20.56 ± 1.34</td>
</tr>
<tr>
<td>zongker</td>
<td>29.20 ± 2.48⊙</td>
<td>52.81%</td>
<td>7.50 ± 1.7</td>
<td>6.40 ± 2.11</td>
</tr>
<tr>
<td>prodom</td>
<td>2.89 ± 1.17</td>
<td>26.31%</td>
<td>3.12 ± 0.11</td>
<td>0.87 ± 0.64</td>
</tr>
<tr>
<td>PolydistH57</td>
<td>6.12 ± 1.38</td>
<td>12.92%</td>
<td>29.35 ± 8⊙</td>
<td>0.70 ± 0.19</td>
</tr>
<tr>
<td>chromo</td>
<td>11.50 ± 1.17</td>
<td>33.76%</td>
<td>3.74 ± 0.58⊙</td>
<td>6.10 ± 0.63</td>
</tr>
<tr>
<td>Mushrooms</td>
<td>7.84 ± 2.21</td>
<td>6.46%</td>
<td>18.39 ± 5.7⊙</td>
<td>2.54 ± 0.56</td>
</tr>
<tr>
<td>swiss-10k</td>
<td>35.90 ± 2.52⊙</td>
<td>17.03%</td>
<td>6.73 ± 0.72⊙</td>
<td>12.08 ± 3.47</td>
</tr>
<tr>
<td>checker-100k</td>
<td>8.54 ± 2.35⊙</td>
<td>2.26%</td>
<td>19.54 ± 2.1⊙</td>
<td>9.66 ± 2.32</td>
</tr>
<tr>
<td>skin</td>
<td>9.38 ± 3.30⊙</td>
<td>0.06%</td>
<td>9.43 ± 2.41⊙</td>
<td>4.22 ± 1.11</td>
</tr>
<tr>
<td>checker</td>
<td>8.94 ± 0.84⊙</td>
<td>0.24%</td>
<td>1.44 ± 0.3⊙</td>
<td>9.38 ± 2.73</td>
</tr>
</tbody>
</table>

Fig. 2: Plots of the simulated data.

Fig. 3: Zoom in a plot of the Friedman output function (green, line). We also show the predicted output using CVR on a clipped mahalanobis kernel (black dashed+ dotted) and a prediction of the output function using iCVR on the indefinite mahalanobis kernel (red, dashed).

with cost of $O(m^2)$ in contrast to constant costs if the full kernel is available. Accordingly, If we assume $m \ll N$ the overall runtime and memory complexity of iCVM/iCVR is linear in $N$, this is two magnitudes less as for KSVM for reasonable large $N$ and for low rank input kernels.

7. Discussions and Conclusions

As discussed in (Loosli et al., 2016), there is no good reason to enforce positive-definiteness in kernel methods. A very detailed discussion on reasons for using KSVM or iCVM is given in (Loosli et al., 2016), explaining why a number of alternatives or pre-processing techniques are in general inappropriate. Our experimental results show that an appropriate Krēin space model provides very good prediction results and using one of the proposed sparsification strategies this can also be achieved for a sparse model in most cases. The proposed iCVM-sparse-OMP is only slightly better than the former iCVM-sparse-sub model with respect to the prediction accuracy but has very few final modeling vectors, with an at least competitive prediction accuracy in the vast majority of data sets. Similar observations are found for the iCVR in comparison to CVR with flipping or clipping. As is the case for KSVM, the presented approach can be applied without the need for transformation of test points, which is a desirable property for practical applications.
Table 3: Mean square error (mean ± std-dev.) in the 10-fold crossvalidation. The percentage of projection points (pts) is calculated using the unique set over core vectors over all classes in comparison to all training points. All sparse-OMP models use only 50 points in the final models. Best results are shown in bold.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>iCVR (non-sparse)</th>
<th>iCVR-flip</th>
<th>iCVR-clip</th>
<th>iCVR (sparse-sub)</th>
<th>pts</th>
<th>iCVR (sparse-OMP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM1</td>
<td>0.25 ± 0.12</td>
<td>0.44 ± 0.43</td>
<td>0.46 ± 0.50</td>
<td>0.33 ± 0.13</td>
<td>17.25%</td>
<td>0.25 ± 0.12</td>
</tr>
<tr>
<td>SIM2</td>
<td>0.14 ± 0.16</td>
<td>0.15 ± 0.18</td>
<td>0.15 ± 0.16</td>
<td>0.15 ± 0.16</td>
<td>50%</td>
<td>0.15 ± 0.18</td>
</tr>
<tr>
<td>SIM3</td>
<td>0.01 ± 0.00</td>
<td>0.01 ± 0.00</td>
<td>0.01 ± 0.00</td>
<td>0.01 ± 0.00</td>
<td>18.68%</td>
<td>0.06 ± 0.01</td>
</tr>
<tr>
<td>DS1</td>
<td>0.83 ± 0.09</td>
<td>0.81 ± 0.07</td>
<td>2.46 ± 4.64</td>
<td>0.85 ± 0.06</td>
<td>7.83%</td>
<td>0.77 ± 0.08</td>
</tr>
<tr>
<td>DS2</td>
<td>1.34 ± 0.57</td>
<td>1.19 ± 0.38</td>
<td>2.15 ± 0.61</td>
<td>1.57 ± 1.08</td>
<td>5.03%</td>
<td>1.12 ± 0.15</td>
</tr>
<tr>
<td>DS3</td>
<td>0.00 ± 0.00</td>
<td>0.01 ± 0.00</td>
<td>0.02 ± 0.00</td>
<td>23.04%</td>
<td>0.05 ± 0.00</td>
<td></td>
</tr>
<tr>
<td>DS4</td>
<td>1.17 ± 0.28</td>
<td>1.24 ± 0.23</td>
<td>1.20 ± 0.16</td>
<td>1.29 ± 0.19</td>
<td>5.22%</td>
<td>0.75 ± 0.05</td>
</tr>
</tbody>
</table>

Acknowledgments

We thank Gaelle Bonnet-Loosli for providing support with the Kérin Space SVM and R. Duin, Delft University for various support with distorts and protos. PT was supported by the EC Horizon 2020 ITN SUNDIAL (Survey Network for Deep Imaging Analysis and Learning), Project ID: 721463. FMS, CR are supported by the FuE program of the StMWi, project OBeRAn, grant number IUK-1709-0011//IUK530/010

References


Deep Imaging Analysis and Learning), Project ID: 721463.

Conflicts of interest

There is no conflict of interest.