Metric and non-metric proximity transformations at linear costs

Andrej Gisbrecht

Theoretical Computer Science

University of Bielefeld

Center of Excellence,

Universitätsstrasse 21-23,

33615, Bielefeld

Frank-Michael Schleif*

School of Computer Science

University of Birmingham

Birmingham

B15 2TT

United Kingdom

November 7, 2014

Abstract

Domain specific (dis-)similarity or proximity measures used e.g. in alignment algorithms of sequence data, are popular to analyze complex data objects and to cover domain specific data properties. *Without an underlying vector space* these data are given as pairwise (dis-)similarities only. The few available methods for such data focus widely on similarities and do not scale to large data sets. Kernel methods are very effective for *metric similarity* matrices, also at large scale, but costly transformations are necessary starting with non-metric (dis-)similarities. We propose an integrative combination of Nyström approximation, potential double centering and eigenvalue correction to obtain valid kernel matrices at *linear costs* in the number of samples. By the proposed approach effective kernel approaches, become accessible. Experiments with several larger (dis-)similarity data sets show that the proposed method achieves much better runtime performance than the standard strategy while keeping competitive model accuracy.

*Corresponding author: schleify@cs.bham.ac.uk

The main contribution is an efficient and accurate technique, to convert (potentially non-metric) large scale *dissimilarity matrices* into approximated positive semi-definite kernel matrices at linear costs.

Keywords: dissimilarity learning, nystroem approximation, double centering, pseudo-euclidean, indefinite kernel

1 Introduction

In many application areas such as bioinformatics, text mining, image retrieval, spectroscopy domains or social networks the available electronic data are increasing and get more complex in size and representation. In general these data are not given in vectorial form and *domain specific* (dis-)similarity measures are used, as a replacement or complement to Euclidean measures. These data are also often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series [7, 31, 42] are of this type. These data are inherently compositional and a feature representation leads to information loss. As an alternative, tailored dissimilarity measures such as pairwise alignment functions, kernels for structures or other domain specific similarity and dissimilarity functions can be used as the interface to the data. But also for vectorial data, non-metric proximity measures are common in some disciplines. An example of this type is the use of divergence measures [10] which are very popular for spectral data analysis in chemistry, geo- and medical sciences [41, 43], and are not metric in general. In such cases, machine learning techniques which can deal with pairwise non-metric similarities or dissimilarities are attractive [46].

The paper is organized as follows. First we give a brief review of related work. Subsequently we review common transformation techniques for dissimilarity data and discuss the influence of non-Euclidean measures, by eigenvalue corrections. Thereafter we discuss alternative methods for processing small dissimilarity data. We extend this discussion to approximation strategies and give an alternative derivation of the Nyström approximation together with a convergence proof, also for indefinite kernels. This allows us to apply the Nyström technique to similarities as well as for dissimilarities. Thus, we can link both strategies effectively to use kernel methods for the analysis of larger (non-)metric dissimilarity data. Then we show the effectiveness of the proposed approach by different supervised learning tasks aligned with various error measures. We also discuss differences and com-

mons to some known approaches supported by experiments on simulated data[1].

2 Related work

Similarity and dissimilarity learning or for short proximity learning has attracted wide attention over the last years, pioneered by work of [24] and major contributions in [46] and different other research groups. As will be detailed more formally in the next section, the learning of proximities is challenging under different aspects: in general there is no underlying vector space, the proximities may be non-Euclidean, the data may not be metric. As mentioned before a symmetric matrix of metric similarities between objects is essentially a kernel and can be analyzed by a multitude of kernel methods [55]. But complex preprocessing steps are necessary, as discussed in the following, to apply them on non-metric (dis-)similarities. Some recent work discussed non-metric *similarities* in the context of kernel approaches by means of indefinite kernels see e.g. [39, 47], resulting in non-convex formulations. Other approaches try to make the kernel representation positive semi definite (psd) or learn an alternative psd proxy matrix close to the original one [7, 8], but with high computational costs. For dissimilarity matrices only few approaches have been published [40, 6] both with quadratic to cubic computational costs in the number of samples. In fact, as discussed in the work of [46], non-Euclidean proximities can encode important information in the Euclidean as well as in non-Euclidean parts of space, represented by the positive and negative eigenvalues of the corresponding similarity matrix, respectively. Thus, transformations of similarities to make them psd, by e.g. truncating the negative eigenvalues, may be inappropriate [49]. This however is very data dependent and for a large number of datasets negative eigenvalues may be actually noise effects while for other data sets the negative eigenvalues carry relevant information [33, 34]. Often non-psd kernels are still used with kernel algorithms but actually on a heuristical basis, since corresponding error bounds are provided only for psd kernels in general. As we will see in the experiments for strongly non-psd data it may happen that standard kernel methods fail to converge due to the violation of underlying assumptions.

Another strategy is to use a more general theory of learning with similarity functions proposed in [2]. Which can be used to identify descriptive or discriminative models based on a available similarity function under some

---

[1]This article contains extended and improved results and is based on [54]

3

conditions [30]. A practical approach of the last type for classification problems was provided in [29]. The model is defined on a fixed randomly chosen set of landmarks per class and a transfer function. Thereby the landmarks are a small set of columns (or rows) of a kernel matrix which are used to formulate the decision function. The weights of the decision function are then optimized by standard approaches. The results are however in general substantially worse than those provided in [7] where the datasets are taken from.

In the following we will focus on non-metric proximities and especially *dissimilarities*. Native methods for the analysis of matrix dissimilarity data have been proposed in [25, 46, 51, 21], but are in general based on non-convex optimization schemes and with quadratic to linear memory and runtime complexity, the later employing some of the approximation techniques discussed subsequently and additional heuristics. The strategy to correct non-metric dissimilarities is addressed in the literature only for smaller data sets. And there exist basically three approaches to make them metric. The first one is to modify the (symmetric) dissimilarity matrix such that all triangle equations in the data are fulfilled [6], which is called the *metric-nearness* problem. The second strategy is to learn again a metric proxy matrix [40]. Both strategies are quite costly and not used at large scale. The third approach is based on converting the dissimilarities to similarities, by double centering followed by an eigenvalue correction of the similarities and back conversion to dissimilarities. These steps scale quadratic and cubic, respectively. We focus on the last approach and provide a runtime and memory efficient solution for problems at large scale[2].

The approximation concepts used in the following are based on the Nyström approximation which was introduced to machine learning by the work of [67]. In [17] the Nyström approximation was used to simplify the normalized Cut problem, which can be considered as a clustering problem. This work was however valid for *psd similarity* matrices, only. An extension to *non-psd* similarities was addressed in [3], but the derivation can still lead to an invalid matrix approximation [3]. Our proposal derives valid eigenvector and eigenvalue estimates also for non-psd proximity matrices.

Large (dis-)similarity data are common in biology like the famous *UniProt-*

---

[2]With large we refer to a sample size $N \in [1e3 - 1e6]$. We do not focus on very big data - which are (not yet) considered in the area of proximity learning.

[3]The derivation of Z on p 535 for negative eigenvalues in Λ leads to complex values and hence invalid results. However the strategy proposed in the corresponding paper may have removed the negative eigenvalues in Λ, due to a rank reduction, explaining the experimental results. But the cut-off of negative eigenvalues can again be criticized [49]

/SwissProt-database with $\approx 500,000$ entries or *GenBank* with $\approx 135,000$ entries, but there are many more (dis-)similarity data as discussed in the work based on [46, 48]. These growing data sets request effective and generic modeling approaches.

Here we will show how potentially non-metric (dis-)similarities can be effectively processed by standard kernel methods by correcting the proximity data with linear costs. The proposed strategies permit the effective application of many kernel methods for these type of data under very mild conditions.

Especially for metric dissimilarities the approach keeps the known guarantees, like generalization bounds (see e.g. [13]). For non-psd data we give a convergence proof, but the corresponding bounds are still open, yet our experiments are promising.

3 Transformation techniques for (dis-)similarities

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$. We assume, there exists a dissimilarity measure such that $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a dissimilarity matrix measuring the pairwise dissimilarities $D_{ij} = d(\mathbf{v}_j, \mathbf{v}_i)^2$ between all pairs $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V}$ [4]. Any reasonable (possibly non-metric) distance measure is sufficient. We assume zero diagonal $d(\mathbf{v}_i, \mathbf{v}_i) = 0$ for all i and symmetry $d(\mathbf{v}_i, \mathbf{v}_j) = d(\mathbf{v}_j, \mathbf{v}_i)$ for all i, j.

3.1 Transformation of dissimilarities and similarities into each other

Every dissimilarity matrix $\mathbf{D}$ can be seen as a distance matrix computed in some, not necessarily Euclidean, vector space. The matrix of the inner products computed in this space is the corresponding similarity matrix $\mathbf{S}$. It can be computed from $\mathbf{D}$ directly by a process referred to as double centering [46]:

$$\mathbf{S} = -\mathbf{J}\mathbf{D}\mathbf{J}/2$$
$$\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$$

with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. Similarly, it is possible to construct the dissimilarity matrix element-wise from the matrix of inner products $\mathbf{S}$

$$D_{ij} = S_{ii} + S_{jj} - 2S_{ij}.$$

---

[4]We assume D_{ij} to be squared to simplify the notation.

As we can see, both matrices $\mathbf{D}$ and $\mathbf{S}$ are closely related to each other and represent the same data, up to translation, which is lost by the double-centering step. If the mean estimate, used in the double centering step, is inaccurate the conversion of $\mathbf{D}$ to $\mathbf{S}$ is inaccurate as well, which can have a negative impact on e.g. a classifier based on $\mathbf{S}$.

The data stems from an Euclidean space, and therefore the distances d_{ij} are Euclidean, if and only if $\mathbf{S}$ is positive semi-definite (psd) [4]. This is the case, when we observe only non-negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$ associated to $\mathbf{D}$. Such psd matrices $\mathbf{S}$ are also referred to as kernels and there are many classification techniques, which have been proposed to deal with such data, like the support vector machine (SVM) [61]. In the case of non-psd similarities, the mercer kernel based techniques are no longer guaranteed to work properly and additional transformations of the data are required or the methods have to be modified substantially, effecting the overall runtime efficiency or desired properties like convexity [45, 26]. To define these transformations we need first to understand the pseudo-Euclidean space.

3.2 Pseudo-Euclidean embedding

Given a symmetric dissimilarity with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space is always possible [24].

Definition 1 (Pseudo-Euclidean space [46]). *A pseudo-Euclidean space $\xi = \mathbb{R}^{(p,q)}$ is a real vector space equipped with a non-degenerate, indefinite inner product $\langle .,. \rangle_\xi$. ξ admits a direct orthogonal decomposition $\xi = \xi_+ \oplus \xi_-$ where $\xi_+ = \mathbb{R}^p$ and $\xi_- = \mathbb{R}^q$ and the inner product is positive definite on ξ_+ and negative definite on ξ_-. The space ξ is therefore characterized by the signature (p,q).*

A symmetric bi-linear form in this space is given by

$$\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{p+q} x_i y_i = \mathbf{x}^\top \mathbf{I}_{p,q} \mathbf{y}$$

where $\mathbf{I}_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Given the eigendecomposition of a similarity matrix $\mathbf{S} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^\top$ we can compute the corresponding vectorial representation $\mathbf{V}$ in the pseudo-Euclidean space by

$$\mathbf{V} = \mathbf{U}_{p+q} \left| \boldsymbol{\Lambda}_{p+q} \right|^{1/2} \tag{1}$$

where $\mathbf{\Lambda}_{p+q}$ consists of p positive and q negative non-zero eigenvalues and $\mathbf{U}_{p+q}$ consists of the corresponding eigenvectors. It is straightforward to see that $D_{ij} = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j \rangle_{p,q}$ holds for every pair of data points. Similarly to the signature (p, q) of a space ξ, we describe our finite data sets, given by a matrix $\mathbf{D}$ or $\mathbf{S}$, by the extended signature $(p, q, N-p-q)$ which represents the number of positive eigenvalues p, the number of negative eigenvalues q and the number of the remaining zero eigenvalues in the similarity matrix.

3.3 Dealing with pseudo-Euclidean data

In [7] different strategies were analyzed to obtain valid kernel matrices for a given similarity matrix $\mathbf{S}$, most popular are: *flipping, clipping, vector-representation, shift correction*. The underlying idea is to remove negative eigenvalues in the eigenspectrum of the matrix $\mathbf{S}$. One may also try to learn an alternative psd kernel representation with maximum alignment to the original non-psd kernel matrix [7, 8, 36] or split the proximities based on positive and negative eigenvalues as discussed in [46, 47].

The *flip*-operation takes the absolute eigenvalues of the matrix $\mathbf{S}$. This corresponds to ignoring the separation of the space ξ into ξ_+ and ξ_- and instead computing in the space $\mathbb{R}^{p+q}$. This approach preserves the variation in the data and could be revoked for some techniques after the training by simply reintroducing the matrix $\mathbf{I}_{p,q}$ into the inner product.

The *shift*-operation increases all eigenvalues by the absolute value of the minimal eigenvalue. This approach performs a non-linear transformation in the pseudo-Euclidean space, emphasizing ξ_+ and nearly eliminating ξ_-.

The *clip*-operation sets all negative eigenvalues to zero. This approach corresponds to ignoring the space ξ_- completely. As discussed in [49], depending on the data set, this space could carry important information and removing it would make some tasks, as e.g. classification, impossible.

After the transformation of the eigenvalues, the corrected matrix $\mathbf{S}^*$ is obtained as $\mathbf{S}^* = \mathbf{U}\mathbf{\Lambda}^*\mathbf{U}^\top$, with $\mathbf{\Lambda}^*$ as the modified eigenvalue matrix using one of the above operations. The obtained matrix $\mathbf{S}^*$ can now be considered as a valid kernel matrix $\mathbf{K}$ and kernel based approaches can be used to operate on the data.

The analysis in [49] indicates that for non-Euclidean dissimilarities some corrections like above may change the data representation such that information loss occurs. This however is not yet systematically explored and very data dependent, best supported by domain knowledge about the data or the used proximity measure.

Alternatively, techniques have been introduced which directly deal with

possibly non-metric dissimilarities. Using the Equation (1) the data can be embedded into the pseudo-Euclidean space. Classical vectorial machine learning algorithms can then be adapted to operate directly in the pseudo-Euclidean space. This can be achieved by e.g. defining a positive definite inner product in the space ξ. Variations of this approach are also possible whereby an explicit embedding is not necessary and the training can be done implicitly, based on the dissimilarity matrix only [46]. A further strategy is to employ so called relational or proximity learning methods as discussed e.g. in [21]. The underlying models consist of prototypes, which are implicitly defined as a weighted linear combination of training points:

$$\mathbf{w}_j = \sum_i \alpha_{ji}\mathbf{v}_i \text{ with } \sum_i \alpha_{ji} = 1. \qquad \mathbb{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_c\}$$

But this explicit representation is not necessary because the algorithms are based only on a specific form of distance calculations using the matrix $\mathbf{D}$ and the potentially unknown vector space V is not needed. The basic idea is an implicit computation of distances $d(\cdot, \cdot)$ during the model calculation based on the dissimilarity matrix $\mathbf{D}$ using weights α:

$$d(\mathbf{v}_i, \mathbf{w}_j)^2 = [\mathbf{D} \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^\top \mathbf{D}\alpha_j. \qquad (2)$$

details in [21]. As shown e.g. in [27] the mentioned methods do not rely on a metric dissimilarity matrix $\mathbf{D}$, but it is sufficient to have a symmetric $\mathbf{D}$ in a pseudo-Euclidean space, with constant self-dissimilarities.

The *dissimilarity space* approach is another technique which does not embed the data into the pseudo-Euclidean space [46]. Instead, one selects a representative set of points $\mathbf{w}_i \in \mathbb{W}$ and considers for every point the dissimilarities to the set $\mathbb{W}$ as features, resulting in a vectorial representation $\mathbf{x}_i = [d(\mathbf{v}_i, \mathbf{w}_1), d(\mathbf{v}_i, \mathbf{w}_2), d(\mathbf{v}_i, \mathbf{w}_3), ...]^\top$. This corresponds to an embedding into an Euclidean space with the dimensionality equal to the size of the selected set of points. These vectors can then be processed using any vectorial approaches. A negative point of this representation is the change of the original data representation which may disturb the structure of the data. It is also highly reliable on a good representative set, since highly correlated sampled points generate similar features and the correlation information is lost in the embedded space.

3.4   Complexity

The methods discussed before are suitable for data analysis based on similarity or dissimilarity data where the number of samples N is rather small,

Figure 1: Schema of the relation between similarities and dissimilarities.

e.g. scales by some thousand samples. For large N, most of the techniques discussed above become infeasible. All techniques which use the full (dis-)similarity matrix, have $\mathcal{O}(N^2)$ memory complexity and thus at least $\mathcal{O}(N^2)$ computational complexity.

Double centering, if done naively, is cubic, although after simplifications it can be computed in $\mathcal{O}(N^2)$. Transformation from $\mathbf{S}$ to $\mathbf{D}$ can be done element-wise, but if the full matrix is required it is still quadratic.

All the techniques relying on the full eigenvalue decomposition, e.g. for eigenvalue correction or for explicit pseudo-Euclidean embedding, have an $\mathcal{O}(N^3)$ computational complexity.

The only exception is the dissimilarity space approach. If it possible to select a good representative set of a small size, one can achieve linear computational and memory complexity. The technique becomes quadratic as well, if all data points are selected as the representative set.

Other then this, only for *metric, similarity data* (psd kernels) efficient approaches have been proposed before, e.g. the Core-Vector Machine (CVM) [59] or low-rank linearized SVM [69] for classification problems or an approximated kernel k-means algorithm for clustering [9].

A schematic view of the relations between $\mathbf{S}$ and $\mathbf{D}$ and its transformations is shown in Figure 1, including the complexity of the transformations. Some of the steps can be done more efficiently by known methods, but with additional constraints or in atypical settings as discussed in the following.

9

In the following, we discuss techniques to deal with larger sample sets for potentially non-metric similarity and especially dissimilarity data. We show how standard kernel methods can be used, assuming that for non-metric data, the necessary transformations have no severe negative influence on the data accuracy. Basically also core-set techniques [1] become accessible for large potentially non-metric (dis-)similarity data in this way, but at the cost of multiple additional intermediate steps. In particular, we investigate the Nyström approximation technique, as low rank linear time approximation technique; we will show its suitability and linear time complexity for similarities as well as dissimilarities, applied on the raw data as well as for the eigenvalue correction.

4 Nyström approximation

As shown in [67], given a symmetric positive semi-definite kernel matrix $\mathbf{K}$, it is possible to create a low rank approximation of this matrix using the Nyström technique [44]. The idea is to sample m points, the so called landmarks, and to analyze the small $m \times m$ kernel matrix $\mathbf{K}_{m,m}$ constructed from the landmarks. The eigenvalues and eigenvectors from the matrix $\mathbf{K}_{m,m}$ can be used to approximate the eigenvalues and eigenvectors of the original matrix $\mathbf{K}$. This allows to represent the complete matrix in terms of a linear part of the full matrix only. The final approximation takes the simple form

$$\hat{\mathbf{K}} = \mathbf{K}_{N,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,N},\tag{3}$$

where $\mathbf{K}_{N,m}$ is the kernel matrix between N data points and m landmarks and $\mathbf{K}_{m,m}^{-1}$ is the Moore-Penrose pseudoinverse of the small matrix.

This technique has been proposed in the context of Mercer kernel methods in [67] with related proofs and bounds given in [13] and very recent results in [22]. It can be applied in conjunction with algorithms using the kernel matrix in multiplications with other matrices or vectors only. Due to the explicit low rank form as in Equation (3) it is possible to select the order of multiplication, thus reducing the complexity from quadratic in the number of data points to a linear one.

4.1 Eigenvalue decomposition of a Nyström approximated matrix

In some applications it might be useful to compute the exact eigenvalue decomposition of the approximated matrix $\hat{\mathbf{K}}$, e.g. to compute the pseudo-

inverse of this matrix. We will show now, how this decomposition can be computed in linear time [5]. The psd matrix approximated by Equation (3) can be written as

$$\hat{\mathbf{K}} = \mathbf{K}_{N,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,N}$$
$$= \mathbf{K}_{N,m}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\top}\mathbf{K}_{N,m}^{\top}$$
$$= \mathbf{B}\mathbf{B}^{\top},$$

where we defined $\mathbf{B} = \mathbf{K}_{N,m}\mathbf{U}\mathbf{\Lambda}^{-1/2}$ with $\mathbf{U}$ and $\mathbf{\Lambda}$ being the eigenvectors and eigenvalues of $\mathbf{K}_{m,m}$, respectively. Further it follows

$$\hat{\mathbf{K}}^2 = \mathbf{B}\mathbf{B}^{\top}\mathbf{B}\mathbf{B}^{\top}$$
$$= \mathbf{B}\mathbf{V}\mathbf{A}\mathbf{V}^{\top}\mathbf{B}^{\top},$$

where $\mathbf{V}$ are the orthonormal eigenvectors of the matrix $\mathbf{B}^{\top}\mathbf{B}$ and $\mathbf{A}$ the matrix of its eigenvalues. The corresponding eigenequation can be written as $\mathbf{B}^{\top}\mathbf{B}\mathbf{v} = a\mathbf{v}$. Multiplying it with $\mathbf{B}$ from left we get the eigenequation for $\hat{\mathbf{K}}$

$$\mathbf{B}\mathbf{B}^{\top}(\mathbf{B}\mathbf{v}) = a(\mathbf{B}\mathbf{v}).$$

It is clear, that $\mathbf{A}$ must be the matrix of eigenvalues of $\hat{\mathbf{K}}$. The matrix $\mathbf{B}\mathbf{v}$ is the matrix of the corresponding eigenvectors, which are orthogonal but not necessary orthonormal. The normalization can be computed from the decomposition

$$\hat{\mathbf{K}} = \mathbf{B}\mathbf{V}\mathbf{V}^{\top}\mathbf{B}^{\top}$$
$$= \mathbf{B}\mathbf{V}\mathbf{A}^{-1/2}\mathbf{A}\mathbf{A}^{-1/2}\mathbf{V}^{\top}\mathbf{B}^{\top}$$
$$= \mathbf{C}\mathbf{A}\mathbf{C}^{\top},$$

where we defined $\mathbf{C} = \mathbf{B}\mathbf{V}\mathbf{A}^{-1/2}$ as the matrix of orthonormal eigenvectors of $\hat{\mathbf{K}}$. Thus, $\hat{\mathbf{K}} = \mathbf{C}\mathbf{A}\mathbf{C}^{\top}$ is the orthonormal eigendecomposition of $\hat{\mathbf{K}}$.

4.2 Convergence proof

The Nyström approximation was proposed for the psd matrices and thus, it was not accessible for distance matrices and similarities coming from non-psd kernel functions. First developments to apply the Nyström technique to indefinite matrices were presented in [20, 21]. Although supported with

---

[5]A similar strategy was used to construct large eigenmaps from *psd* similarity matrices as recently shown [58] but our approach applies also to non-psd matrices.

experiments, a formal proof was lacking. Here we present a proof, that shows, that the Nyström approximated, possible indefinite, kernel converges in the operator norm to the true underlying kernel as long as the number of landmarks is large enough. Generalization bounds will be a subject of future work.

Let K be an integral operator and its kernel $k \in L^2(\Omega^2)$ be a continuous symmetric function (not necessarily psd, i.e. it does not have to reproduce a Hilbert space):

$$Kf(x) := \int_\Omega k(x,y)f(y)d\mu(y).$$

Without loss of generality let Ω be an interval $[a,b] \subset \mathbb{R}$ with measure 1. Then K is a compact operator in a Hilbert space $\mathfrak{H}$

$$\|K\|_{L^2 \to L^2} := \sup_{\|f\| \le 1} \|Kf\|_{L^2} \le \|k\|_{L_2},$$

with the operator norm $\|.\|_{L^2 \to L^2}$ and the L_2-norm $\|.\|_{L_2}$.

We define a measurement operator T_m which divides the space Ω into m spaces Ω_j, each with the measure $1/m$. It converts functions $f \in \mathfrak{H}$ to functions $f_m \in \mathfrak{H}_m$ which are piece-wise constant on each Ω_j. The corresponding integral kernel of T_m is defined as:

$$t_m(x,y) := \begin{cases} m & x,y \in \Omega_j \text{ for any } j \\ 0 & \text{else.} \end{cases}$$

It follows for an $x \in \Omega_j$ that

$$T_m f(x) = \int_\Omega t_m(x,y)f(y)d\mu(y) = m \int_{\Omega_j} f(y)d\mu(y),$$

where we can see, that the right hand side is the mean value of $f(y)$ on Ω_j and thus constant for all $x \in \Omega_j$. This way, the operator T_m allows us to approximate a function $f(x)$ by measuring it at m places $f(x_j)$ and assuming that it is constant in between. Measuring the operator K we get

$K_m := T_m \circ K$ with the integral kernel

$$\int_\Omega t_m(x,z)k(z,y)d\mu(z) = \sum_{j=1}^m \int_{\Omega_j} t_m(x,z)k(z,y)d\mu(z)$$

$$= \sum_{j=1}^m 1_{\Omega_j}(x)m \int_{\Omega_j} k(z,y)d\mu(z)$$

$$= \sum_{j=1}^m 1_{\Omega_j}(x)k_j(y)$$

$$=: k_m(x,y),$$

where $1_{\Omega_j}(x)$ is the indicator function which is 1 if $x \in \Omega_j$ and 0 elsewhere and we defined $k_j = m \int_{\Omega_j} k(z,y)d\mu(z)$.

We can now analyze the convergence behavior of K_m to K. $\forall x \in \Omega_j$ and $\forall y \in \Omega$ we get

$$|k_m(x,y) - k(x,y)| =$$

$$= \left| m \int_{\Omega_j} k(z,y)d\mu(z) - m \int_{\Omega_j} k(x,y)d\mu(z) \right|$$

$$\leq m \int_{\Omega_j} |k(z,y) - k(x,y)| \, d\mu(z).$$

Since k is continuous on the interval $[a,b]$, it is uniformly continuous and we can bound

$$|k(z,y) - k(x,y)| \leq \mathcal{D}(\Omega_j) := \sup_{\substack{x_1, x_2 \in \Omega_j \\ y \in \Omega}} |k(x_1,y) - k(x_2,y)|$$

$$\leq \delta_m := \max_j \mathcal{D}(\Omega_j)$$

and therefore

$$\sup_{\substack{x \in \Omega \\ y \in \Omega}} |k_m(x,y) - k(x,y)| \leq \delta_m.$$

For $m \to \infty$ the Ω_j become smaller and $\delta_m \to 0$, thus kernel k_m converges to k. For the operators K and K_m it follows

$$\|K_m - K\|_{L^2 \to L^2} \to 0$$

which shows that K_m converges to K in the operator norm, if the number of measurements goes to infinity.

Applying K_m on f results in

$$K_m f(x) = \int_\Omega k_m(x,y)f(y)d\mu(y)$$
$$= \sum_{j=1}^{m} 1_{\Omega_j}(x) \int_\Omega k_j(y)f(y)d\mu(y)$$
$$= \sum_{j=1}^{m} a_j 1_{\Omega_j}(x)$$

where $a_j := \int_\Omega k_j(y)f(y)d\mu(y)$ is a constant with respect to x. It is clear that $K_m f$ is always in the linear hull of $1_{\Omega_1}(x), ..., 1_{\Omega_m}(x)$ and the image of the operator $\Im K_m = \text{span}\{1_{\Omega_1}(x), ..., 1_{\Omega_m}(x)\}$ is m dimensional. Since the coefficients a_j are finite, K_m is a compact operator and because the sequence of K_m converges to K, we see that K is in fact a compact operator.

According to the "Perturbation of bounded operators" theorem [62], if a sequence K_m converges to K in the operator norm, then for an isolated eigenvalue λ of K there exist isolated eigenvalues λ_m of K_m such that $\lambda_m \to \lambda$ and the corresponding spectral projections converge in operator norm. This theorem allows us to estimate the eigenvalues and eigenfunctions of the unknown operator K by computing the eigendecomposition of the measured operator K_m.

The eigenfunctions and eigenvalues of the operator K_m are given as the solutions of the eigenequation

$$K_m f = \lambda f. \tag{4}$$

We know that the left hand side of the equation is in the image of K_m and therefore an eigenfunction f must have the form

$$f(x) = \sum_{i=1}^{m} f_i 1_{\Omega_i}(x) \tag{5}$$

14

where f_i are constants. For the left side of the Equation (4) it follows

$$K_m f(x) = \int_\Omega \sum_{j=1}^m 1_{\Omega_j}(x) k_j(y) f(y) d\mu(y)$$

$$= \sum_{j=1}^m 1_{\Omega_j}(x) \int_\Omega k_j(y) \sum_{i=1}^m f_i 1_{\Omega_i}(y) d\mu(y)$$

$$= \sum_{j=1}^m \sum_{i=1}^m 1_{\Omega_j}(x) f_i \int_{\Omega_i} k_j(y) d\mu(y)$$

$$= \sum_{j=1}^m \sum_{i=1}^m 1_{\Omega_j}(x) \frac{1}{m} f_i k_{ji}$$

and we defined $k_{ji} = m \int_{\Omega_i} k_j(y) d\mu(y) = m^2 \int_{\Omega_i} \int_{\Omega_j} k(y, z) d\mu(y) d\mu(z)$ which represents our measurement of the kernel k around the i-th and j-th points. If we combine the above equation with the Equation (4) for an $x \in \Omega_j$ we get

$$\sum_{i=1}^m \frac{1}{m} k_{ji} f_i = \lambda f_j.$$

This equation is a weighted eigenequation and we can turn it into a regular eigenequation by defining $\tilde{\lambda} = m\lambda$ and $\tilde{f}_i = f_i/\sqrt{m}$. Thus, we get

$$\sum_{i=1}^m k_{ji} \tilde{f}_i = \tilde{\lambda} \tilde{f}_j.$$

Hence $\tilde{\lambda}$ and $\tilde{f}$ are the eigenvalues and eigenvectors of matrix (k_{ji}). Note, that f_i are scaled to guarantee the normalization of $\tilde{f}$

$$1 = \int_\Omega f(x) f(x) d\mu(x)$$

$$= \int_\Omega \sum_{i=1}^m f_i^2 1_{\Omega_i}(x) d\mu(x)$$

$$= \sum_{i=1}^m f_i^2 \int_{\Omega_i} d\mu(x)$$

$$= \sum_{i=1}^m \left(\frac{f_i}{\sqrt{m}}\right)^2.$$

The eigendecomposition takes the form

$$(k_{ji}) = \sum_{l=1}^{m} \tilde{\lambda}^l \tilde{f}^l (\tilde{f}^l)'$$

and for a single measured element we get

$$k_{ij} = \sum_{l=1}^{m} \tilde{\lambda}^l \tilde{f}_i^l \tilde{f}_j^l.$$

According to the spectral theorem [66] the eigendecomposition of k is

$$k(x,y) = \sum_{l=1}^{\infty} \gamma^l \phi^l(x)\phi^l(y)$$

where γ^l and ϕ^l are the eigenvalues and eigenfunctions, respectively. Since K is a compact operator, γ^l is a null sequence. Thus, the sequence of operators $\tilde{K}_m$ with the kernel $\tilde{k}_m(x,y) = \sum_{l=1}^{m} \gamma^l \phi^l(x)\phi^l(y)$ converges to K in the operator norm for $m \to \infty$ [66] and we can approximate

$$k(x,y) \approx \sum_{l=1}^{m} \gamma^l \phi^l(x)\phi^l(y)$$
$$= \sum_{l=1}^{m} \int_{\Omega} k(x,z)\phi^l(z)d\mu(z)\frac{1}{\gamma^l}\int_{\Omega} k(y,z')\phi^l(z')d\mu(z'),$$

where we assume that none of the γ^l are zero. Further, due to the "Perturbation of bounded operators" theorem, the eigenvalues λ^l converge to γ^l and the corresponding eigenspaces converge in the operator norm and we can approximate

$$k(x,y) \approx \sum_{l=1}^{m} \int_{\Omega} k(x,z)f^l(z)d\mu(z)\frac{1}{\lambda^l}\int_{\Omega} k(y,z')f^l(z')d\mu(z').$$

16

Taking into account the Equation (5) the above formula turns into

$$k(x,y) \approx \sum_{l=1}^{m} \int_{\Omega} k(x,z) \sum_{i=1}^{m} f_i^l 1_{\Omega_i}(z) d\mu(z)$$

$$\cdot \frac{1}{\lambda^l} \int_{\Omega} k(y,z') \sum_{j=1}^{m} f_j^l 1_{\Omega_j}(z') d\mu(z')$$

$$= \sum_{l=1}^{m} \sum_{i=1}^{m} f_i^l \int_{\Omega_i} k(x,z) d\mu(z) \frac{1}{\lambda^l} \sum_{j=1}^{m} f_j^l \int_{\Omega_j} k(y,z') d\mu(z')$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} k_i(x) \left(\sum_{l=1}^{m} \frac{f_i^l}{\sqrt{m}} \frac{1}{m\lambda^l} \frac{f_j^l}{\sqrt{m}} \right) k_j(y)$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} k_i(x) \left(k^{-1} \right)_{ij} k_j(y),$$

where k^{-1} is the pseudo-inverse of the matrix consisting of elements k_{ij}. It is now clear, that after measuring $k_i(x)$ at N places and writing the above formula in matrix form, we retain the original Nyström approximation as in Equation (3).

Note, that the approximation of $k(x,y)$ consists of two approximations. The first one is the approximation of the rank of the matrix and the second one is the approximation of the eigenfunctions and eigenvalues. Although we don't know the exact eigenvalues and eigenfunctions of kernel $k(x,y)$, the approximation is exact if the kernel has a rank $\leq m$ [6]. This fact is known for the Nyström approximation and can be validated by simple matrix transformations. The reason is, that if the rank of a kernel is m then it can be represented as an inner product in a pseudo-Euclidean space and m linearly independent landmarks build a basis which spans this space. The position of any new point x is then fully determined by $k(x,x_i)$, with x_i being the landmarks, so that all inner products between any points are determined and the matrix $\mathbf{K}$ can be computed precisely.

The Nyström approximation involves the computation of $\mathbf{K}_{N,m}$ and inversion of $\mathbf{K}_{m,m}$ with the corresponding complexities of $\mathcal{O}(mN)$ and $\mathcal{O}(m^3)$, respectively. The multiplication of both matrices as well as multiplication of the approximated matrix with other matrices, required for further processing

---

[6]If the true rank is larger than m the eigenvalues do not match the true once and errors occur - like with any other approach. However the presented approach can also keep negative eigenvalues, given they are within the top m eigenvalues.

Figure 2: Updated schema from Figure 1 using the discussed approximation. The costs are now substantially smaller, provided $m \ll N$.

and training, has the complexity of $\mathcal{O}(m^2 N)$. Thus, the overall complexity of the Nyström technique is given by $\mathcal{O}(m^2 N)$.

5 Transformations of (dis-)similarities with linear costs

The Nyström approximation was proposed originally to deal with large psd similarity matrices with kernel approaches in mind by [67]. To apply these techniques on indefinite similarity and dissimilarity matrices additional transformations, as discussed in section 3, are required. Unfortunately, these transformations have quadratic or even cubic time complexity, making the advantage gained by the Nyström approximation pointless. Since we can now apply the Nyström technique on arbitrary symmetric matrices, it is not only possible to approximate the dissimilarities directly, but also to perform the transformations in linear time. Thus, we can apply relational and kernel techniques on similarities and dissimilarities including eigenvalue corrections if necessary.

In this section we will elaborate how the transformations discussed in section 3 can be done in linear time if applied for the Nyström-approximated matrices. The updated costs are shown on the Figure 2.

18

5.1 Transformation of dissimilarities and similarities into each other

Given a dissimilarity matrix $\mathbf{D}$, there are two ways to construct the approximated matrix $\hat{\mathbf{S}}$. First, we can transform $\mathbf{D}$ to $\mathbf{S}$ using double centering and then apply Nyström approximation to $\mathbf{S}$. Obviously, this approach has quadratic time complexity due to the double centering step. Second, we can approximate $\mathbf{D}$ to $\hat{\mathbf{D}}$ first and then apply double centering. As we will show in the following, this transformation requires only linear computational time.

As mentioned before, from the dissimilarity matrix $\mathbf{D}$ we can compute the corresponding similiarity matrix using double centering. This process is noted as $\mathbf{S}(\mathbf{D})$ in the following:

$$\mathbf{S}(\mathbf{D}) = -\mathbf{J}\mathbf{D}\mathbf{J}/2$$

where $\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$ with identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$. Expanding the right side of the equation we get

$$
\begin{aligned}
\mathbf{S}(\mathbf{D}) &= -\frac{1}{2}\mathbf{J}\mathbf{D}\mathbf{J} \\
&= -\frac{1}{2}\left(\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\mathbf{D}\left(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\right) \\
&= -\frac{1}{2}\left(\mathbf{D} - \frac{1}{N}\mathbf{D}\mathbf{1}\mathbf{1}^\top - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\mathbf{D} + \frac{1}{N^2}\mathbf{1}\mathbf{1}^\top\mathbf{D}\mathbf{1}\mathbf{1}^\top\right).
\end{aligned}
$$

Approximating $\mathbf{S}(\mathbf{D})$ requires computation of a linear part of each summand, but still involves summation over the full matrix $\mathbf{D}$.

Alternatively, by approximating $\mathbf{D}$ first, we get

$$
\begin{aligned}
\mathbf{S} \overset{Ny}{\approx} \mathbf{S}(\hat{\mathbf{D}}) = -\frac{1}{2}\bigg[& \mathbf{D}_{N,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,N} - \frac{1}{N}\mathbf{D}_{N,m} \\
& \cdot (\mathbf{D}_{m,m}^{-1} \cdot (\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top - \frac{1}{N}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m}) \cdot \mathbf{D}_{m,m}^{-1}) \\
& \cdot \mathbf{D}_{m,N} + \frac{1}{N^2}\mathbf{1}((\mathbf{1}^\top\mathbf{D}_{N,m}) \cdot \mathbf{D}_{m,m}^{-1} \cdot (\mathbf{D}_{m,N}\mathbf{1}))\mathbf{1}^\top\bigg].
\end{aligned} \quad (6)
$$

This equation can be rewritten for each entry of the matrix $\mathbf{S}(\hat{\mathbf{D}})$

$$
\begin{aligned}
\hat{S}_{ij}(\hat{\mathbf{D}}) \;=\; -\frac{1}{2}\Bigg[& \mathbf{D}_{i,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,j} \\
& -\frac{1}{N}\sum_{k} \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,j} \\
& -\frac{1}{N}\sum_{k} \mathbf{D}_{i,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,k} \\
& +\frac{1}{N^2}\sum_{kl} \mathbf{D}_{k,m} \cdot \mathbf{D}_{m,m}^{-1} \cdot \mathbf{D}_{m,l} \Bigg],
\end{aligned}
$$

as well as for the sub-matrices $\mathbf{S}_{m,m}(\hat{\mathbf{D}})$ and $\mathbf{S}_{N,m}(\hat{\mathbf{D}})$, in which we are interested for the Nyström approximation

$$
\begin{aligned}
\mathbf{S}_{m,m}(\hat{\mathbf{D}}) \;=\; -\frac{1}{2}\Bigg[& \mathbf{D}_{m,m} - \frac{1}{N}\mathbf{1}\cdot\sum_{k}\mathbf{D}_{k,m} \\
& -\frac{1}{N}\sum_{k}\mathbf{D}_{m,k}\cdot\mathbf{1}^{\top} \\
& +\frac{1}{N^2}\mathbf{1}\cdot\sum_{kl}\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,l}\cdot\mathbf{1}^{\top} \Bigg]
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{S}_{N,m}(\hat{\mathbf{D}}) \;=\; -\frac{1}{2}\Bigg[& \mathbf{D}_{N,m} - \frac{1}{N}\mathbf{1}\cdot\sum_{k}\mathbf{D}_{k,m} \\
& -\frac{1}{N}\sum_{k}\mathbf{D}_{N,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,k}\cdot\mathbf{1}^{\top} \\
& +\frac{1}{N^2}\mathbf{1}\cdot\sum_{kl}\mathbf{D}_{k,m}\cdot\mathbf{D}_{m,m}^{-1}\cdot\mathbf{D}_{m,l}\cdot\mathbf{1}^{\top} \Bigg].
\end{aligned}
$$

Now the matrix $\mathbf{S}(\hat{\mathbf{D}})$ can be approximated via the matrix $\hat{\mathbf{S}}(\hat{\mathbf{D}})$ using the matrices $\mathbf{S}_{m,m}(\hat{\mathbf{D}})$ and $\mathbf{S}_{N,m}(\hat{\mathbf{D}})$. This requires only a linear part of $\mathbf{D}$ and involves linear computation time.

Comparing this approach to the quadratic computation of $\mathbf{S}_{N,m}$, we see, that the first three summands are identical and only the forth summand is different. This term involves summation over the full dissimilarity matrix

and, depending on the approximation quality of $\hat{\mathbf{D}}$, might vary. The deviation is added to each pairwise similarity resulting in a non-linear transformation of the data. If m corresponds to the rank of $\mathbf{D}$ then double centering is exact and no information loss occurs during the approximation. Otherwise, the information loss increases with smaller m for both approaches and the error is made by approximating $\mathbf{S}$ in the first case and by approximating $\mathbf{D}$ in the second case. If the Nyström approximation is feasible for a given data set, then the second approach allows to perform the transformation in linear instead of quadratic time.

It should be mentioned that a similar transformation is possible with the landmark multidimensional scaling (L-MDS) [12] which is widely known in the visualization community and typically used to embed data into a low $2-3$ dimensional space. Embeddings to higher dimensions are possible but not considered, in general. The idea of L-MDS is to sample a small amount m of points, the so called landmarks, compute the corresponding dissimilarity matrix followed by a double centering on this matrix. Finally the data are projected to a low dimensional space using an eigenvalue decomposition. The remaining points can then be projected into the same space, taking into account the distances to the landmarks, and applying a triangulation. From this vectorial representation of the data one can easily retrieve the similarity matrix as a scalar product between the points.

It was shown, that L-MDS is also a Nyström technique by [52], but compared to our proposed approach in Equation (6) L-MDS makes not only an error in the forth summand, but also in the second and the third. Additionally, and more importantly, by projecting into *Euclidean space* it makes an implicit clipping of the eigenvalues. As discussed above and will be shown later, this might disturb data significantly, leading to qualitatively worse results. Thus, our proposed method can be seen as a generalization of L-MDS and should be used instead.

Similarly to the transformation from $\mathbf{D}$ to $\hat{\mathbf{S}}$, there are two ways to transform $\mathbf{S}$ to $\hat{\mathbf{D}}$. First, transform the full matrix $\mathbf{S}$ to $\mathbf{D}$ using $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$ and then apply the Nyström approximation

$$\hat{\mathbf{D}} = \mathbf{D}_{N,m}\mathbf{D}_{m,m}^{-1}\mathbf{D}_{N,m}^{\top}. \tag{7}$$

Second, approximate $\mathbf{S}$ with $\hat{\mathbf{S}}$ and then transform it to $\hat{\mathbf{D}}$. The first approach requires quadratic time, since it transforms the full matrix. In the second approach only $\mathbf{D}_{N,m}$ is computed, thus making it linear in time and memory. Obviously, both approaches produce the same results, but the second one is significantly faster. The reason is, that for the computation of

$\hat{\mathbf{D}}$ only the matrix $\mathbf{D}_{N,m}$ is required and it is not necessary to compute the rest of $\mathbf{D}$.

5.2   Eigenvalue correction

For non-Euclidean data, the corresponding similarity matrix is indefinite. We would like to make the data Euclidean in order to avoid convergence issues, or to be able to use kernel methods. A strategy to obtain a valid kernel matrix from similarities is to apply an eigenvalue correction as discussed in section 3.3. This however can be prohibitive for large matrices, since to correct the whole eigenvalue spectrum, the whole eigenvalue decomposition is needed, which has $\mathcal{O}(N^3)$ complexity. The Nyström approximation can again decrease computational costs dramatically. Since we can now apply the approximation on an arbitrary symmetric matrix, we can make the correction afterwards, reducing the complexity to a linear one, as we will show now.

Given non-metric dissimilarities $\mathbf{D}$, we can first approximate them and then convert to approximated similarities $\hat{\mathbf{S}}(\hat{\mathbf{D}})$ using the Equation (6). For similarities $\hat{\mathbf{S}}$ given directly or obtained from $\hat{\mathbf{S}}(\hat{\mathbf{D}})$, we need to compute the eigenvalue decomposition in linear time. As we have shown in the section 4.1, it is possible to compute the exact eigenvalue decomposition of a Nyström-approximated psd matrix in linear time, given the corresponding similarity matrix has indeed rank m. Since $\hat{\mathbf{S}}$ is indefinite, we can not apply the above technique directly. Instead, since in a squared matrix the eigenvectors stay the same, we first compute

$$\begin{aligned}
\hat{\mathbf{S}}^2 &= \mathbf{S}_{N,m}\mathbf{S}_{m,m}^{-1}\left(\mathbf{S}_{m,N}\cdot\mathbf{S}_{N,m}\right)\mathbf{S}_{m,m}^{-1}\mathbf{S}_{m,N} \\
&= \mathbf{S}_{N,m}\tilde{\mathbf{S}}_{m,m}\mathbf{S}_{N,m}^{\top} \\
&= \mathbf{C}\tilde{\mathbf{A}}\mathbf{C}^{\top}.
\end{aligned}$$

The resulting matrix can be computed in linear time and is psd. This means, we can determine its eigenvalue decomposition as described in section 4.1:

$$\hat{\mathbf{S}}^2 = \mathbf{C}\tilde{\mathbf{A}}\mathbf{C}^{\top},$$

where $\tilde{\mathbf{A}}$ are the eigenvalues of $\hat{\mathbf{S}}^2$ and $\mathbf{C}$ are the eigenvectors of both $\hat{\mathbf{S}}^2$ and $\hat{\mathbf{S}}$.

Using the eigenvectors $\mathbf{C}$, the eigenvalues $\mathbf{A}$ of $\hat{\mathbf{S}} = \mathbf{C}\mathbf{A}\mathbf{C}^{\top}$ can be retrieved via

$\mathbf{A} = \mathbf{C}^{\top}\hat{\mathbf{S}}\mathbf{C}$. Then we can correct the eigenvalues $\mathbf{A}$ by some technique as discussed in section 3.3 to $\mathbf{A}^*$. The corrected approximated matrix $\hat{\mathbf{S}}^*$ is

then simply

$$\hat{\mathbf{S}}^* = \mathbf{C}\mathbf{A}^*\mathbf{C}^\top. \tag{8}$$

Thus, using a low rank representation of a similarity matrix we can compute its eigenvalue decomposition and perform eigenvalue correction in linear time. If it is desirable to work with the corrected dissimilarities, then using the Equation (7), it is possible to transform the corrected similarity matrix $\hat{\mathbf{S}}^*$ back to dissimilarities resulting in the corrected and approximated matrix $\hat{\mathbf{D}}^*$.

5.3   Out-of-sample extension

Usually models are learned by a training set and we expect them to generalize well on the new unseen data, or the test set. In such cases we need to provide an out-of-sample extension, i.e. a way to apply the model on the new data. This might be a problem for the techniques dealing with (dis-)similarities. For example, in proxy approaches the out of sample extension is in general handled by solving another costly optimization problem [8, 40]. If the matrices are corrected, we need to correct the new (dis-)similarities as well to get consistent results. Fortunately this can be easily done in the Nyström framework.

If we compare the Equations (3) and (8) we see that the correction is performed on a different decomposition of $\hat{\mathbf{S}}$, i.e.:

$$\mathbf{S}_{N,m}\mathbf{S}_{m,m}\mathbf{S}_{N,m}^\top = \hat{\mathbf{S}} = \mathbf{C}\mathbf{A}\mathbf{C}^\top. \tag{9}$$

If we correct $\mathbf{A}$ it is not clear what happens on the left side of the above equation. Therefore, to compute the out-of-sample extension we need to find a simple transformation from one decomposition to the other. Taking a linear part $\hat{\mathbf{S}}_{N,m}$ from the equation 9 we get

$$\mathbf{S}_{N,m} = \mathbf{C}_{N,m}\mathbf{A}\mathbf{C}_{m,m}^\top,$$

which leads after simple transformation to

$$\mathbf{C}_{N,m} = \mathbf{S}_{N,m}\left(\mathbf{A}\mathbf{C}_{m,m}^\top\right)^{-1}.$$

Plugging the above formula into Equation (8) we get

$$
\begin{aligned}
\hat{\mathbf{S}}^* &= \mathbf{S}_{N,m} \left(\mathbf{A}\mathbf{C}_{m,m}^\top \right)^{-1} \mathbf{A}^* \left(\left(\mathbf{A}\mathbf{C}_{m,m}^\top \right)^{-1} \right)^\top \mathbf{S}_{N,m}^\top \\
&= \mathbf{S}_{N,m}(\mathbf{C}_{m,m}^\top)^{-1}\mathbf{A}^{-1}\mathbf{A}^*\mathbf{A}^{-1}\mathbf{C}_{m,m}^{-1}\mathbf{S}_{N,m}^\top \\
&= \mathbf{S}_{N,m}(\mathbf{C}_{m,m}^\top)^{-1}(\mathbf{A}^*)^{-1}\mathbf{C}_{m,m}^{-1}\mathbf{S}_{N,m}^\top \\
&= \mathbf{S}_{N,m} \left(\mathbf{C}_{m,m}\mathbf{A}^*\mathbf{C}_{m,m}^\top \right)^{-1} \mathbf{S}_{N,m}^\top
\end{aligned}
$$

and we see that we simply need to extend the matrix $\mathbf{S}_{N,m}$ by uncorrected similarities between the new points and the landmarks to obtain the full approximated and *corrected* similarity matrix, which then can be used by the algorithms to compute the out-of-sample extension. The same approach can be applied to the dissimilarity matrices. Here we first need to transform the new dissimilarities to similarities using Equation (6), correct them and then transform back to dissimilarities.

In [7] a similar approach is taken. First, the whole similarity matrix is corrected by means of a projection matrix. Then this projection matrix is applied to the new data, so that the corrected similarity between old and new data can be computed. This technique is in fact the Nyström approximation, where the whole similarity matrix $\mathbf{S}$ is treated as the approximation matrix $\mathbf{S}_{m,m}$ and the old data, together with the new data build the matrix $\mathbf{S}_{N,m}$. Rewriting this in the Nyström framework makes it clear and more obvious, without the need to compute the projection matrix and with an additional possibility to compute the similarities between the new points.

5.4 Proof of concept

We close this section by a small experiment on the ball dataset as proposed in [15]. It is an artificial dataset based on the surface distances of randomly positioned balls of two classes having a slightly different radius. The dataset is non-Euclidean with substantial information encoded in the negative part of the eigenspectrum. We generated the data with 300 samples per class leading to an $N \times N$ dissimilarity matrix $\mathbf{D}$, with $N = 600$. Now the data have been processed in four different ways to obtain a valid kernel matrix $\mathbf{S}$. First encoding, denoted as $SIM1$, was constructed by converting $\mathbf{D}$ to $\mathbf{S}$ with double centering and computing the full eigenvalue decomposition. The negative eigenvalues were then corrected by flipping. This approach, which we will refer to as the **standard approach** in the following, has a complexity of $\mathcal{O}(N^3)$.

Table 1: Test set results of a 10-fold SVM run on the ball dataset using the different encodings.

	$SIM1$	$SIM2$	$SIM3$	$SIM4$
Test-Accuracy	100 ± 0	88.83 ± 3.15	51.50 ± 6.64	50.67 ± 3.94

Further, we generated an approximated similarity matrix $\hat{\mathbf{S}}^*$ by using the proposed approach, flipping in the eigenvalue correction and 10 landmarks for the Nyström approximation. This dataset is denoted as $SIM2$ and was obtained with a complexity of $\mathcal{O}(m^2 N)$. The third dataset $SIM3$ was obtained in the same way but the eigenvalues were clipped. The dataset $SIM4$ was obtained using landmark MDS with the same landmarks as for $SIM2$ and $SIM3$. The data are processed by a Support Vector Machine in a 10-fold crossvalidation. The results on the test sets are shown in the Table 1.

As mentioned, the data contain substantial information in the negative fraction of the eigenspectrum, accordingly one may expect that these eigenvalues should not be removed. This is also reflected in the results. L-MDS removed the negative eigenvalues and the classification model based on these data shows random prediction accuracy. The SIM3 encoding is a bit better. Also in this case the negative eigenvalues are removed but the limited amount of class separation information, encoded in the positive fraction was better preserved, probably due to the different calculation of the matrix $\hat{\mathbf{S}}_{mm}$. The SIM2 data used the flipping strategy and shows already quite good prediction accuracy, taking into account that the kernel matrix is only approximated by 10 landmarks and the relevant (original negative) eigenvalues are of small magnitude.

As a last point it should be mentioned that corrections like clipping, flipping and their effect on the data representation are still under discussion and considered to be not always optimal [46]. Additionally the selection of landmark points is discussed in [68, 32] Further, for very large data sets (e.g. some 100 million points) the Nyström approximation may still be too costly and some other strategies have to be found as suggested in [35].

6   Experiments

We now apply the priorly derived approach to six non-metric dissimilarity and similarity data and show the effectiveness for a classification task. The considered data are (1) the imbalanced SwissProt similarity data as de-

scribed in [31] consisting of protein sequence alignment scores, (2) the balanced chromosome dissimilarity data taken from [42] with scores of aligned gray value images, (3) the imbalanced proteom dissimilarity data set from [14], (4) the balanced Zongker digit dissimilarity data from [14, 28] which is based on deformable template matchings of 2000 handwritten NIST digits. Further the balanced Delft gestures data base (DS5) taken from [14] and the WoodyPlants50 (Woody) (DS6) from the same source is used. DS5 represents a sign-language interpretation problem with dissimilarities computed using a dynamic time warping procedure on the sequence of positions [37]. The DS6 dataset contains of shape dissimilarities between leaves collected in a study on woody plants [38]. Further details about the data can be found in Table 2.

Table 2: Overview of the considered datasets and their properties.

Data set	Name	# samples	# classes	Signature
DS1	SwissProt	10988	30	[8488,2500,0]
DS2	Chromosom	4200	21	[2258,1899,43]
DS3	Proteom	2604	53	[1502,682,420]
DS4	Zongker	2000	10	[961,1038,1]
DS5	Delft	1500	20	[963,536,1]
DS6	Woody	791	14	[602,188,1]

All datasets are non-metric, multiclass and contain a large number of objects, such that a regular eigenvalue correction with a prior double centering for dissimilarity data, as discussed before, is already very costly but can still be calculated to get comparative results.

6.1 Classification performance

The data are analyzed in various ways, employing either the clipping eigenvalue correction, the flipping eigenvalue correction, or by not-correcting the eigenvalues [7]. To be effective for the large number of objects we also apply the Nyström approximation as discussed before using $10, 50, 100$ and all points as landmarks. If the data have high rank > 100, they are potentially not well suited for approximations and approximation errors are unavoidable. Landmarks have been selected randomly from the data. Other sampling strategies have been discussed in [16, 68, 56], however with additional

---

[7]Shift correction was found to have a negative impact on the model as already discussed in [7].

Table 3: Signature and average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5), Woody (DS6) using a Nyström approximation with $10, 50, 100, full$ landmarks and no, clip or flip eigenvalue correction.

	10 / Clip	10 / Flip	10 / No	10 L-MDS
DS1	[9, 0, 10979]	[10, 0, 10978]	[9, 1, 10978]	
	30.67 ± 5.07*	$\mathbf{31.65 \pm 5.41}$*	5.93 ± 5.23	26.47 ± 6.27
DS2	[9, 0 ,4191]	[10, 0, 4190]	[9,1, 4190]	
	67.61 ± 6.49	$\mathbf{74.83 \pm 3.23}$*	18.79 ± 14.08	67.086 ± 6.09
DS3	[9, 0 ,2595]	[10, 0, 2594]	[9, 1, 2594]	
	59.33 ± 6.87*	$\mathbf{62.43 \pm 7.30}$*	2.52 ± 2.33	56.74 ± 6.26
DS4	[8, 0 ,1992]	[10, 0, 1996]	[8, 2, 1990]	
	42.51 ± 10.51*	$\mathbf{44.92 \pm 11.07}$*	10.63 ± 3.15	32.83 ± 9.49
DS5	[9, 0 ,1491]	[10, 0, 1490]	[9, 1, 1490]	
	73.75 ± 5.12	$\mathbf{78.76 \pm 4.60}$*	15.12 ± 13.05	73.86 ± 5.72
DS6	[9, 0 ,782]	[10, 0, 781]	[9, 1, 781]	
	75.96 ± 4.89	$\mathbf{79.51 \pm 5.33}$*	38.86 ± 14.14	76.03 ± 4.77

meta parameters, which we would like to avoid for clarity of the proposed approach. Also the impact of the Nyström approximation with respect to kernel methods has been discussed recently in [11], but this is out of the focus of the presented approach. For comparison we also show the results as obtained by using Landmark-MDS, which naturally applies a clipping and, as mentioned before, makes various simplifications in the conversion step, which can lead to inaccuracies in the data representation. The prediction accuracies of a 10-fold crossvalidation for $m = \{10, 50, 100\}$ are shown in Table 3-5. The influence of N with respect to a fixed number of landmarks is studied in the experiment shown in Figure 3. A runtime analysis, comparing to the standard approach, is shown in Figure 6. The results of the standard approach where no approximations are used but only eigenvalue corrections on the full matrix are provided in Table 6. We also provide results for the dissimilarity-space representation using a linear and an elm kernel [18] in Table 7. As mentioned before this representation does not need any approximations or eigenvalue corrections but the out of sample extension is costly if many landmarks are chosen, or the selection of the landmarks has to be optimized using e.g. a wrapper approach [50]. Here we use all points as landmarks to simplify the evaluation.

To get comparable experiments, the same randomly drawn landmarks are used in each of the corresponding sub-experiments (along a column in the table). New landmarks are only drawn for different Nyström approx-

Figure 3: Top: box-plots of the classification performance for different sample sizes of DS1 using the proposed approach with 500 landmarks. Bottom: The same experiment but with the standard approach. Obviously our approach does not sacrifice performance for computational speed.

Table 4: Signature and average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5), Woody (DS6) using a Nyström approximation with $10, 50, 100, full$ landmarks and no, clip or flip eigenvalue correction.

	50 / Clip	50 / Flip	50 / No	50 L-MDS
DS1	[49, 0 ,10939]	[50, 0, 10930]	[49, 1, 10931]	
	76.21 ± 5.13	76.49 ± 3.73	69.05 ± 5.01	$\mathbf{76.59 \pm 4.65}$
DS2	[49, 0 ,4151]	[50, 0, 4150]	[49,1, 4150]	
	94.05 ± 1.17	93.94 ± 1.28	83.66 ± 25.43	$\mathbf{94.11 \pm 1.21}$
DS3	[48, 0 ,2556]	[50, 0, 2554]	[49, 1, 2550]	
	93.08 ± 2.25	$\mathbf{93.82 \pm 1.59}^*$	3.53 ± 3.25	92.35 ± 2.08
DS4	[34, 0 ,1979]	[50, 0, 1950]	[34, 16, 1950]	
	$80.79 \pm 3.94^*$	$\mathbf{85.35 \pm 3.42}^*$	9.82 ± 2.08	73.57 ± 6.71
DS5	[48,0,1452]	[50, 0, 1450]	[48, 2, 1450]	
	$\mathbf{95.31 \pm 1.82}$	94.72 ± 2.25	24.99 ± 27.56	95.31 ± 1.89
DS6	[49, 0 ,742]	[50, 0, 741]	[49, 1, 741]	
	88.55 ± 4.11	$\mathbf{89.30 \pm 3.72}$	81.40 ± 23.63	88.46 ± 4.35

Table 5: Signature and average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5), Woody (DS6) using a Nyström approximation with $10, 50, 100, full$ landmarks and no, clip or flip eigenvalue correction.

	100 / Clip	100 / Flip	100 / No	100 L-MDS
DS1	[99, 0 ,10889]	[100, 0, 10888]	[99, 1, 10888]	
	87.62 ± 2.11	87.63 ± 1.85	$\mathbf{88.17 \pm 2.19}$	87.50 ± 2.24
DS2	[91, 0 ,4109]	[100, 0, 4100]	[91,9,4100]	
	95.00 ± 1.11	94.71 ± 1.68	11.29 ± 7.68	$\mathbf{95.18 \pm 1.07}$
DS3	[96, 0 ,2506]	[99, 0, 2505]	[97, 2, 2505]	
	96.48 ± 1.34	$\mathbf{96.96 \pm 1.17}$	13.75 ± 9.90	96.29 ± 1.27
DS4	[63, 0 ,1937]	[100, 0, 1900]	[62, 38, 1900]	
	$83.47 \pm 4.31^*$	$\mathbf{87.42 \pm 3.15}^*$	10.55 ± 2.43	80.34 ± 7.73
DS5	[91, 0 ,1401]	[100, 0, 1400]	[92, 8, 1400]	
	$\mathbf{96.07 \pm 1.56}$	94.74 ± 4.23	23.33 ± 18.62	96.01 ± 1.69
DS6	[96, 0 ,695]	[100, 0, 691]	[96, 4, 691]	
	90.69 ± 3.38	$\mathbf{90.71 \pm 3.20}$	38.11 ± 23.74	90.51 ± 3.65

Figure 4: Spearman rank correlation (left) and the crossvalidation accuracy (right) for the three largest data sets using the proposed approach with an interleaved double centering and Nyström approximation on the dissimilarity data.

(a) Swiss correlation

(b) Swiss accuracy

(c) Chromosom correlation

(d) Chromosom accuracy

(e) Proteom correlation

(f) Proteom accuracy

Table 6: Average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5), Woody (DS6) using the standard approach (no-approximations) and the flip, clip or no-eigenvalue correction on the full matrix. This has $\mathcal{O}(N^3)$ complexity.

Data set	clip	flip	no
DS1	95.45 ± 0.88	95.39 ± 1.01	95.40 ± 0.59
DS2	97.12 ± 0.89	97.17 ± 0.99	96.93 ± 0.66
DS3	99.42 ± 0.66	99.42 ± 0.45	99.38 ± 0.61
DS4	95.65 ± 1.13	96.25 ± 0.75	25.25 ± 4.78
DS5	98.33 ± 1.67	98.00 ± 0.94	96.13 ± 1.43
DS6	92.54 ± 2.27	93.17 ± 2.48	89.63 ± 3.58

Table 7: Average test set accuracy for SwissProt (DS1), Chromosome (DS2), Proteom (DS3), Zongker (DS4), Delft gestures (DS5), Woody (DS6) using the dissimilarity space representation and a linear kernel or an elm kernel.

Data set	linear	elm
DS1	26.01 ± 5.49	72.09 ± 0.96
DS2	76.76 ± 1.11	89.88 ± 0.96
DS3	68.36 ± 2.48	85.37 ± 2.86
DS4	93.70 ± 2.04	95.05 ± 1.71
DS5	87.73 ± 3.83	91.67 ± 2.58
DS6	28.83 ± 6.97	89.38 ± 4.48

imations and for sample sizes shown in Figure 3. Classification rates are calculated in a 10-fold crossvalidation with 10 repeats using the Core-Vector-Machine (CVM) [59]. The crossvalidation does not include a new draw of the landmarks, to cancel out the selection bias of the Nyström approximation, accordingly CVM use the same kernel matrices. However, our objective is not maximum classification performance (which is only one possible application) but to demonstrate the effectiveness of our approach for dissimilarity data of larger scale.

First, one observes that the eigenvalue correction has a strong, positive effect on the classification performance consistent with earlier findings [7]. Best results over a row are highlighted in bold at the various result tables. If the difference is significantly better than L-MDS a $\star$ has been added. Raising the number of landmarks improves the classification performance for

Figure 5: Logarithmic representation of the eigenspectrum of the unapproximated and double centered matrix for the larger datasets DS1 - DS3.

(a) Swiss eigenspectrum (b) Chromosom eigenspectrum (c) Proteom eigenspectrum

the experiments with eigenvalue correction. Using kernels without eigenvalue correction has in general a negative impact. While an increase in the number of landmarks leads to a better approximation of the dataset and may therefore improve the classification accuracy it can also raise the influence of negative eigenvalues, damping the performance[8]. We found that flipping is in general superior to clipping. For $m = 10$ flipping was consistently better than clipping or L-MDS. With an increase of m the approximation error of L-MDS vanishes and the results become more and more similar to the clipping results. But for DS4 L-MDS is also inferior if $m = 100$, which shows that for some data L-MDS gives bad results, due to its approximation errors even for rather large m. Especially for DS3,DS4 and DS6 we observe that the proposed method gives much better results.

In Table 7 we also show the crossvalidation results by use of the priorly mentioned dissimilarity space representation. For simplicity we use an N dimensional feature space and analyse the obtained vector representation by means of a linear kernel and a defacto parameter free elm kernel as proposed by [18]. For the majority of the experiments the obtained results are significantly worse with the exception of DS4. Also for DS5 a comparison with the Nyström approximation at $m = 100$ gives still acceptable results. It should be noted that the results of the elm-kernel experiments are consistently better compared to the linear kernel, indicating the high non-linearity of the data. Obviously the dissimilarity space representation is in general no reasonable alternative. Additionally it becomes very costly for out-of-sample extensions if the number of considered features is large.

---

[8]Comparing signatures at different Nyström approximations also shows that many eigenvalues are close to zero and are sometimes counted as positive,negative or zero.

Figure 6: Runtime analysis of the proposed vs the standard approach for the larger considered dissimilarity data sets. All eigenvalues of the data sets have been processed by flipping.

(a) Chromosom

(b) Delft gestures

(c) Proteom

(d) Zongker

(e) SwissProt

In another experiment, see Figure 4 we analyzed the proximity preservation of the approximated and corrected matrix with respect to the unapproximated and corrected matrix. One would expect that for very low Nyström rates (high approximation), only the dominating eigenvalues are kept and the approximation suffers mainly when the eigenspectra are very smooth. At increasing Nyström rates (lower approximation), first more and more small eigenvalues (also negative ones) are kept leading to a more complex data set and accordingly also a more complex proximity preservation task. Finally if the Nyström rates are high (almost no approximation) one would expect a perfect preservation. This effect is indeed observed in Figure 4. We used the Spearman's rank correlation to measure how far the ranks of the proximities (e.g. distances) are preserved between the two approaches, namely our proposal and a full double centering, followed by a full eigenvalue correction. Low correlation indicates that the data relations are not well preserved whereas small correlation errors indicate that most likely only local neighborhood relations are confused. Comparing the correlation results (left plots in Figure 4) with the prediction accuracy on the test data (right plots in Figure 4) we see that only strong variations in the correlation lead to strong misclassifications. This agrees with our expectation that the data are potentially clustered and local errors in the data relation have only a weak or no effect on the classification model. Similar results were found if we compare our approach to data which have been first double-centered without approximations and where only the eigenvalue correction is done using the Nyström approach.

From the analysis we can conclude that the proposed approach is quite effective to keep the global relations in the data space also for quite high approximations, which is relevant for classification and clustering the data. The local neighborhood relations are kept only for approximation rates of above 60%. As one can see from smooth eigenspectra in Figure 5, the rank of the data sets is rather high, accordingly only for large m the approximation can keep detail information, effecting the local relationships of the data points. Thus, if the different classes are close to each other and have complex nonlinear boundaries, decreasing the number of landmarks leads to an increased classification error. In practice, as can be seen on the Figure 4, the number of the landmarks needs to be very small to take effect. It is thus possible to approximate the matrices by selecting m sufficiently small, without sacrificing the classification accuracy.

6.2 Runtime performance

As shown exemplary in Figure 3 the classification performance on eigenvalue-corrected data is approximately the same for our proposed strategy and the standard approach. But the runtime performance is drastically better for an increase in the number of samples. To show this we selected subsets from the considered data with different sizes from 1000 to the maximal number, while the number of landmarks is fixed by $L = 500$ and calculated the runtime and classification performance using the CVM classifier in a 10-fold crossvalidation. The eigenvalues have been flipped in this experiment. The results of the proposed approach compared to the standard approach are shown in the plots of Figure 6. For larger N the runtime of the standard method (red/dashed line) is two magnitudes larger on log-scale compared to the proposed approach.

7 Large scale experiments

As a final experiment we analyze the proposed approach for large scale non-metric proximity data. With respect to the work presented in the former sections a valid application of kernel methods for such data is not yet possible. Neither the classical eigenvalue correction approach [7] nor the learning of a proximity kernel [8] scales to larger data sets with $N \gg 1e3$ samples, the problem becomes even more challenging if the data are given as dissimilarities such that a double centering is needed to keep a corresponding representation. Due to the large number of samples a full matrix reconstruction is not any longer possible to calculate error measures like the Spearman rank correlation accordingly we only provide test set errors obtained within a 10 fold crossvalidation using a CVM. In our experiments we consider:

- The SwissProt protein database [5] but now at *larger scale* in the version of 11/2010, restricted to ProSite labeled sequences with at least $1,000$ entries per label. We obtain 46 ProSite labels and $82,525$ sequences which are compared by the Smith-Waterman alignment algorithm as provided in [57]. We refer to this data as DS-L-1. The obtained similarity scores are symmetric but non-metric, accordingly standard kernel methods can not be used directly in a valid form. We take $1,000$ landmarks, randomly taken from the selected classes. The dataset has 2 larger negative eigenvalues in the approximated matrix.

- The Pavia remote sensing data consist of 42.776 spectra (DS-L-2). The dataset is taken from [19]. We use the symmetrized Kullback-Leibler

35

Data	size	type	flip	clip	No	L-MDS (clip)
DS-L-1	80k	S	**96.24 ± 0.29%**	96.22 ± 0.28%	failed	96.14 ± 0.27%
DS-L-2	40k	D	**82.56 ± 0.60%**	79.80 ± 0.94%	failed	81.18 ± 1.17%
DS-L-3	50k	D	**88.11 ± 0.68%***	85.06 ± 0.73%	failed	81.37 ± 0.62%
Ball-Large	30k	D	**93.59 ± 0.63%***	50.28 ± 0.80%	28.50 ± 0.76%	50.13 ± 0.97%

Table 8: Crossvalidation results of the large scale data sets (D - dissimilarities, S - similarities) using flip, clip or no eigenvalue correction.

Divergence, which is also known as the spectral information divergence (SID) in remote sensing and frequently used as an effective *non-metric* measure to compare spectral data [60] and use 10% randomly chosen points as landmarks.

- The Salina data of 54129 points (DS-L-3) also taken from [19] with the same measure and settings as for DS-L-2

- The ball dataset with 30,000 samples (Ball-Large). Landmarks are selected randomly as 10% from the dataset.

For all of these data sets a standard kernel approach is costly in calculating the whole similarity matrix and it would be basically impossible to get an eigenvalue correction in a reasonable time. Modern kernel classifiers like the Core-Vector Machine (CVM)[59] do not need to evaluate all the kernel similarities but our similarities are non-metric and an accurate online eigenvalue correction is not available. However we can use our presented approach approximating the score matrix as well as performing an eigenvalue correction. The calculation of the final approximated kernel function and eigenvalue correction by the presented approach takes only some minutes.

The obtained approximated and now positive semi definite similarity matrices can be used by a Core-Vector Machine in a 10 fold crossvalidation to generate a classification model with a good mean prediction accuracy see Table 8. An additional benefit of the CVM approach is that it naturally leads to very sparse models. Accordingly the out of sample extension to new sequences requires only few score calculations to the sequences of the training set.

8   Conclusions

In this article we addressed the analysis of potentially non-metric proximity data and especially the relation between dissimilarity and similarity data.

We proposed effective and *accurate* transformations across the different representations. The results show that our approach can be understood as a generalization of Landmark MDS. L-MDS did not show any significant superior results compared to our method, but instead was often found to be significantly worse. This finding also persisted if the number of landmarks was raised to a rather large value.

Dedicated learning algorithms for dissimilarities and kernels are now accessible for both types of data. The specific coupling of double centering and Nyström approximation permits to compute an exact eigenvalue decomposition in linear time which is a valuable result for many different methods depending on the exact calculation of eigenvalues and eigenvectors of a proximity matrix. While our strategy is very effective e.g. to improve supervised learning of non-metric dissimilarities by kernel methods, it is however also limited again by the Nyström approximation, which itself may fail to provide sufficient approximation and accordingly further research in this line is of interest. Nevertheless, dedicated methods for arbitrary proximity data as addressed in [49] will also be subject of future work. For non-psd data the error introduced by the Nyström approximation and the eigenvalue correction is not yet fully understood and bounds similar as proposed in [13] are still an open issue. It is also of interest to extend our approach to other types of matrix approximation schemes as e.g. the CUR algorithm and others [64, 65, 63]. In future work we will also analyze in more detail the handling of extremely large (dis-)similarity sets [53, 23] and analyze our approach in the context of unsupervised problems [70].

Acknowledgments

9 Appendix

Definition: The norm of an operator $K : L^2(\Omega) \to L^2(\Omega)$ is defined as

$$\|K\|_{L^2 \to L^2} = \sup_{\|f\| \leq 1} \|Kf\|_{L^2}$$

and the norm of a function $f \in L^2(\Omega)$ is defined as

$$\|f\|_{L^2} = \left(\int_\Omega |f(x)|^2 d\mu(x) \right)^{1/2}.$$

Theorem: The sequence of operators K_m converges uniformly to K in the operator norm if

$$\sup_{\substack{x \in \Omega \\ y \in \Omega}} |k_m(x,y) - k(x,y)| \leq \delta_m$$

and $\delta_m \to 0$ for $m \to \infty$.

Proof: The uniform convergence is given if $\|K_m - K\|_{L^2 \to L^2} \to 0$ for $m \to \infty$. Thus, we need to compute this quantity. Following the computations in [66], we can write for the norm of Kf

$$
\begin{aligned}
\|Kf\|_{L^2}^2 &= \int_\Omega |Kf(x)|^2 d\mu(x) \\
&= \int_\Omega \left| \int_\Omega k(x,y) f(y) d\mu(y) \right|^2 d\mu(x) \\
&\leq \int_\Omega \left(\int_\Omega |k(x,y)||f(y)| d\mu(y) \right)^2 d\mu(x) \\
&\leq \int_\Omega \left(\int_\Omega |k(x,y)|^2 d\mu(y) \right) \left(\int_\Omega |f(y)|^2 d\mu(y) \right) d\mu(x) \\
&= \int_\Omega \int_\Omega |k(x,y)|^2 d\mu(x) d\mu(y) \|f\|_{L^2}^2
\end{aligned}
$$

where we used Hölder's inequality and Fubini's theorem. It follows

$$\begin{aligned}
\|K_m - K\|_{L^2 \to L^2}^2 &= \sup_{\|f\| \leq 1} \|(K_m - K)f\|_{L^2}^2 \\
&= \sup_{\|f\| \leq 1} \int_\Omega |(K_m - K)f(x)|^2 d\mu(x) \\
&\leq \sup_{\|f\| \leq 1} \int_\Omega \int_\Omega |k_m(x,y) - k(x,y)|^2 d\mu(x) d\mu(y) \|f\|_{L^2}^2 \\
&\leq \int_\Omega \int_\Omega \delta_m^2 d\mu(x) d\mu(y) \\
&= \delta_m^2
\end{aligned}$$

and since $\delta_m \to 0$ for $m \to \infty$, we have $\|K_m - K\|_{L^2 \to L^2} \to 0$ for $m \to \infty$.
□

References

[1] Mihai Badoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Comput. Geom.*, 40(1):14–22, 2008.

[2] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1-2):89–112, 2008.

[3] Serge Belongie, Charless Fowlkes, Fan R. K. Chung, and Jitendra Malik. Spectral partitioning with indefinite kernels using the Nyström extension. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *ECCV (3)*, volume 2352 of *Lecture Notes in Computer Science*, pages 531–542. Springer, 2002.

[4] Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, 1984.

[5] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider. The swiss-prot protein knowledgebase and its supplement trembl in 2003,. *Nucleic Acids Research*, 31:365–370.

[6] Justin Brickell, Inderjit S. Dhillon, Suvrit Sra, and Joel A. Tropp. The metric nearness problem. *SIAM J. Matrix Analysis Applications*, 30(1):375–396, 2008.

[7] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *JMLR*, 10:747–776, 2009.

[8] Yihua Chen, Maya R. Gupta, and Benjamin Recht. Learning kernels from indefinite similarities. In *In Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, page 19, 2009.

[9] Radha Chitta, Rong Jin, Timothy C. Havens, and Anil K. Jain. Approximate kernel k-means: solution to large scale kernel clustering. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 895–903, 2011.

[10] A. Cichocki and S.-I. Amari. Families of alpha- beta- and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

[11] Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. *JMLR - Proceedings Track*, 9:113–120, 2010.

[12] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 705–712, 2002.

[13] Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[14] R. P.W. Duin. PRTools, march 2012.

[15] Robert P. W. Duin and Elzbieta Pekalska. Non-euclidean dissimilarities: Causes and informativeness. In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR&SPR 2010, Cesme, Izmir, Turkey, August 18-20, 2010. Proceedings*, pages 324–333, 2010.

[16] Ahmed K. Farahat, Ali Ghodsi, and Mohamed S. Kamel. A novel greedy algorithm for nyström approximation. *JMLR - Proceedings Track*, 15:269–277, 2011.

[17] Charless Fowlkes, Serge Belongie, Fan R. K. Chung, and Jitendra Malik. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004.

[18] Benoît Frénay and Michel Verleysen. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, 74(16):2526–2531, 2011.

[19] Computational Intelligence Group from the Basque University. Hyperspectral remote sensing scenes, june 2014.

[20] A. Gisbrecht, B. Mokbel, and B. Hammer. The Nystrom approximation for relational generative topographic mappings. In *NIPS Workshop*, 2010.

[21] A. Gisbrecht, B. Mokbel, F.-M. Schleif, X. Zhu, and B. Hammer. Linear time relational prototype based learning. *Journal of Neural Systems*, 22(5):online, 2012.

[22] Alex Gittens and Michael W. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. *CoRR*, abs/1303.1849, 2013.

[23] Alex Gittens and Michael W. Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *ICML (3)*, volume 28 of *JMLR Proceedings*, pages 567–575. JMLR.org, 2013.

[24] L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17(5):575 – 582, 1984.

[25] Thore Graepel and Klaus Obermayer. A stochastic self-organizing map for proximity data. *Neural Computation*, 11(1):139–155, 1999.

[26] B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

[27] Barbara Hammer and Alexander Hasenfuss. Topographic mapping of large dissimilarity data sets. *Neural Computation*, 22(9):2229–2284, 2010.

[28] A.K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

[29] Purushottam Kar and Prateek Jain. Similarity-based learning via data driven embeddings. In *Proc. of Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Granada, Spain*, pages 1998–2006, 2011.

[30] Purushottam Kar and Prateek Jain. Supervised learning with similarity functions. In *Proc. of Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, Nevada, United States*, pages 215–223, 2012.

[31] T. Kohonen and P. Somervuo. How to make large self-organizing maps for nonvectorial data. *Neural Networks*, 15(8-9):945–952, 2002.

[32] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. *Journal of Machine Learning Research*, 13:981–1006, 2012.

[33] Julian Laub. *Non-metric pairwise proximity data*. PhD thesis, 2004.

[34] Julian Laub, Volker Roth, Joachim M. Buhmann, and Klaus-Robert Müller. On the information and representation of non-euclidean pairwise data. *Pattern Recognition*, 39(10):1815–1826, 2006.

[35] Mu Li, James T. Kwok, and Bao-Liang Lu. Making large-scale nyström approximation possible. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 631–638, 2010.

[36] Wu-Jun Li, Zhihua Zhang, and Dit-Yan Yeung. Latent wishart processes for relational kernel learning. *JMLR - Proceedings Track*, 5:336–343, 2009.

[37] J.F. Lichtenauer, E.A. Hendriks, and M.J.T. Reinders. Sign language recognition by combining statistical dtw and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, 2008.

[38] Haibin Ling and David W. Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, 2007.

[39] S. Liwicki, S. Zafeiriou, G. Tzimiropoulos, and M. Pantic. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 23(10):1624–1636, 2012.

[40] F. Lu, S. Keles abd S. J. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12332–12337, 2005.

[41] E. Mwebaze, P. Schneider, F.-M. Schleif, J.R. Aduwo, J.A. Quinn, S. Haase, T. Villmann, and M. Biehl. Divergence based classification in learning vector quantization. *NeuroComputing*, 74:1429–1435, 2010.

[42] M. Neuhaus and H. Bunke. Edit distance based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.

[43] N.Q. Nguyen, C.K. Abbey, and M.F. Insana. Objective assessment of sonographic: Quality ii acquisition information spectrum. *IEEE Transactions on Medical Imaging*, 32(4):691–698, 2013.

[44] E. J. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54(1):185–204, 1930.

[45] C.S. Ong, X. Mary, S. Canu, and A.J. Smola. Learning with non-positive kernels. pages 639–646, 2004.

[46] E. Pekalska and R. Duin. *The dissimilarity representation for pattern recognition*. World Scientific, 2005.

[47] Elsbieta Pekalska and Bernard Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1017–1032, 2009.

[48] Elzbieta Pekalska and Robert P. W. Duin. Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(6):729–744, 2008.

[49] Elzbieta Pekalska, Robert P. W. Duin, Simon Günter, and Horst Bunke. On not making dissimilarities euclidean. In *Structural, Syntactic, and*

Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings, pages 1145–1154, 2004.

[50] Elzbieta Pekalska, Robert P. W. Duin, and Pavel Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208, 2006.

[51] Elzbieta Pekalska, Pavel Paclík, and Robert P. W. Duin. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research*, 2:175–211, 2001.

[52] J. Platt. Fastmap, metricmap, and landmark mds are all nyström algorithms, 2005.

[53] F.-M. Schleif. Proximity learning for non-standard big data. In *Proceedings of ESANN 2014*, pages 359–364, 2014.

[54] F.-M. Schleif and A. Gisbrecht. Data analysis of (non-)metric proximities at linear costs. In *Proceedings of SIMBAD 2013*, pages 59–74, 2013.

[55] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis and Discovery*. Cambridge University Press, 2004.

[56] Si Si, Cho-Jui Hsieh, and Inderjit S. Dhillon. Memory efficient kernel approximation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Proceedings*, pages 701–709. JMLR.org, 2014.

[57] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[58] Ameet Talwalkar, Sanjiv Kumar, Mehryar Mohri, and Henry Rowley. Large-scale svd and manifold learning. *Journal of Machine Learning Research*, 14:3129–3152, 2013.

[59] Ivor W. Tsang, András Kocsor, and James T. Kwok. Simpler core vector machines with enclosing balls. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 911–918, 2007.

[60] F. v. d. Meer. The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery. *International Journal of Applied Earth Observation and Geoinformation*, 8(1):3–17, 2006.

[61] V.N. Vapnik. *The nature of statistical learning theory*. Statistics for engineering and information science. Springer, 2000.

[62] Ulrike von Luxburg, Olivier Bousquet, and Mikhail Belkin. On the convergence of spectral clustering on random samples: The normalized case. In *Learning Theory, 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1-4, 2004, Proceedings*, pages 457–471, 2004.

[63] Shusen Wang, Chao Zhang, Hui Qian, and Zhihua Zhang. Improving the modified nyström method using spectral shifting. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 611–620. ACM, 2014.

[64] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 14:2729–2769, 2013.

[65] Shusen Wang and Zhihua Zhang. Efficient algorithms and error analysis for the modified nystrom method. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Proceedings*, pages 996–1004. JMLR.org, 2014.

[66] Dirk Werner. *Funktionalanalysis*. Springer-Verlag Berlin Heidelberg, 2011.

[67] Christopher K. I. Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 682–688, 2000.

[68] Kai Zhang and James T. Kwok. Clustered Nyström method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, 21(10):1576–1587, 2010.

[69] Kai Zhang, Liang Lan, Zhuang Wang, and Fabian Moerchen. Scaling up kernel svm on limited resources: A low-rank linearization approach. *JMLR - Proceedings Track*, 22:1425–1434, 2012.

[70] Kai Zhang, Ivor W. Tsang, and James T. Kwok. Improved Nyström low-rank approximation and error analysis. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1232–1239. ACM, 2008.