Incremental Probabilistic Classification Vector Machine with linear costs

F.-M. Schleif[1], H. Chen[2], and P. Tino[1]
1 - University of Birmingham,
School of Computer Science,
B15 2TT Birmingham, UK
2 - University of Science and Technology of China (USTC)
Hefei, Anhui, China
Email: {schleify|p.tino}@cs.bham.ac.uk
Email:hchen@ustc.edu.cn

***Abstract*—The probabilistic classification vector machine is a very effective and generic *probabilistic* and sparse classifier. A recently published incremental version improved the runtime complexity to quadratic costs. We derive the Nyström approximation for asymmetric matrices to obtain *linear* runtime and memory complexity for the incremental probabilistic classification vector machine while keeping similar prediction performance.**

I. Introduction

The Probabilistic Classification Vector Machine (PCVM) was introduced in [1] as a sparse probabilistic kernel classifier pruning unused basis functions during training. The PCVM was found to be very successful and is a high ranked classification algorithm[1] with a recently published incremental version called (EPCVM) [2]. Initially, the PCVM model is generated with N basis functions and has a complexity of $\mathcal{O}(N^3)$, where N is the number of samples. The EPCVM has improved this complexity to $\mathcal{O}(N^2)$ by iteratively adding relevant basis function. This update is based on a sparse relevance learning concept as originally introduced for the Relevance Vector Machine (RVM) [3]. The relevance learning used in EPCVM operates on a (in general) quadratic kernel matrix, containing the input similarities. While the EPCVM has significantly improved the scalability of PCVM the update of the relevance parameters is still costly with quadratic costs in memory and runtime. We propose to approximate the input matrix using the Nyström approximation [4] and to redefine the relevance learning for this approximated matrix. This operation remains exact if the rank of the matrix equals the number of independent landmarks points. The Nyström approximation is a very popular approach used not only in classification but also for other kernel algorithms [5], [6]. A simple application into EPCVM is not possible and some more elaborated modifications are needed.

In Section II we review PCVM and EPCVM and derive the Nyström approximation for asymmetric (potentially rectangular) matrices. We also outline various modifications of the original EPCVM formulation to ensure that memory and runtime complexity remains linear as discussed in Section III. An accurate *probabilistic* model is obtained making EPCVM efficient also for problems at larger scale. We evaluate the derived algorithm on a variety of classical vectorial benchmark data from medium to larger scale in comparison to state of the art classifiers in Section IV. Additionally we show the efficiency of Ny-EPCVM and EPCVM for data representations based on non-mercer kernels. The latter one is very important in a variety of application fields where the data are measured by non-standard often non-metric similarity measures. Section V concludes with a brief summary and discussion of our results.

[1]IEEE TNN outstanding paper award 2009. PCVM Code available at: http://staff.ustc.edu.cn/~hchen/software.htm

II. Probabilistic Classification Vector Learning for large scale

A. Probabilistic classification vector machine

In the following we briefly review PCVM [1]. As other kernel methods PCVM uses a kernel regression model $\sum_{i=1}^{N} w_i \phi_{i,\theta}(\mathbf{x}) + b$ to which a link function is applied, with w_i being the weights of the basis functions $\phi_{i,\theta}(\mathbf{x})$ and b as a bias term. The basis functions will correspond to kernels evaluated at data items. Consider binary classification and a data set of input-target training pairs $D = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $y_i \in \{-1, +1\}$. The implementation of PCVM [2] uses the probit link function, i.e.

$$\Psi(x) = \int_{-\infty}^{x} \mathcal{N}(t|0,1)dt,$$

where $\Psi(x)$ is the cumulative distribution of the normal distribution $\mathcal{N}(0,1)$. Parameters are optimized by an Expectation Maximization (EM) scheme. After incorporating the probit link function, the PCVM model becomes:

$$l(\mathbf{x};\mathbf{w},b) = \Psi\left(\sum_{i=1}^{N} w_i \phi_{i,\theta}(\mathbf{x}) + b\right) = \Psi\left(\Phi_\theta(\mathbf{x})\mathbf{w} + b\right) \quad (1)$$

Where $\Phi_\theta(\mathbf{x})$ is a vector of basis function evaluations for data item $\mathbf{x}$.

In the PCVM formulation [1], a truncated Gaussian prior N_t with mode at 0 is introduced for each weight w_i. Its support is restricted to $[0, \infty)$ for entries of the positive class and $(-\infty, 0]$ for entries of the negative class as shown in Eq. (4). A zero-mean Gaussian prior is adopted for the bias b. The priors

978-1-4799-1959-8/15/$31.00 ©2015 IEEE

are assumed to be mutually independent.

$$p(\mathbf{w}|\alpha) = \prod_{i=1}^{N} p(w_i|\alpha_i) = \prod_{i=1}^{N} N_t(w_i|0,\alpha_i^{-1}), \quad (2)$$

$$p(b|\beta) = \mathcal{N}(b|0,\beta^{-1}), \quad (3)$$

where α_i and β are inverse variances:

$$p(w_i|\alpha_i) = \begin{cases} 2\mathcal{N}(w_i|0,\alpha_i^{-1}) & \text{if } y_i w_i > 0 \\ 0 & \text{otherwise} \end{cases} = 2\mathcal{N}(w_i|0,\alpha_i^{-1}) \cdot \delta(y_i w_i). \quad (4)$$

where $\delta(\cdot)$ is the indicator function $\mathbf{1}_{x>0}(x)$.

We follow the standard probabilistic formulation and assume that $z_\theta(\mathbf{x}) = \Phi_\theta(\mathbf{x})\mathbf{w} + b$ is corrupted by an additive random noise ϵ , where $\epsilon \sim \mathcal{N}(0,1)$. According to the probit link model, if $h_\theta(\mathbf{x}) = \Phi_\theta(\mathbf{x})\mathbf{w} + b + \epsilon \geq 0, y = 1$ and if $h_\theta(\mathbf{x}) = \Phi_\theta(\mathbf{x})\mathbf{w} + b + \epsilon < 0, y = -1$. We obtain:

$$p(y = 1|\mathbf{x},\mathbf{w},b) = p(\Phi_\theta(\mathbf{x})\mathbf{w}+b+\epsilon \geq 0) = \Psi(\Phi_\theta(\mathbf{x})\mathbf{w}+b). \quad (5)$$

$h_\theta(\mathbf{x})$ is a latent variable because ϵ is an unobservable variable. We collect evaluations of $h_\theta(\mathbf{x})$ at training points in a vector $\mathbf{H}_\theta(\mathbf{x}) = (h_\theta(\mathbf{x_1}),\ldots,h_\theta(\mathbf{x_N}))^\top$. In the expectation step the expected value $\bar{\mathbf{H}}_\theta$ of $\mathbf{H}_\theta$ with respect to the posterior distribution over the latent variables is calculated (given old values $\mathbf{w}^{\text{old}}, b^{\text{old}}$). In the maximization step the parameters are updated through

$$\mathbf{w}^{\text{new}} = M(M\Phi_\theta^\top(\mathbf{x})\Phi_\theta(\mathbf{x})M + I_N)^{-1} \quad (6)$$

$$M(\Phi_\theta^\top(\mathbf{x})\bar{\mathbf{H}}_\theta - b\Phi_\theta^\top(\mathbf{x})\mathbf{I}) \quad (7)$$

$$\mathbf{b}^{\text{new}} = t(1+tNt)^{-1}t(\mathbf{I}^\top\bar{\mathbf{H}}_\theta - \mathbf{I}^\top\Phi_\theta(\mathbf{x})\mathbf{w}) \quad (8)$$

where I_N is a N-dimensional identity matrix and I a all-ones vector, the diagonal elements in the diagonal matrix M are:

$$m_i = \begin{cases} \sqrt{2}w_i & \text{if } \quad y_i w_i \geq 0 \\ 0 & \text{else} \end{cases} \quad (9)$$

and the scalar $t = \sqrt{2}|b|$. For further details see [1].

B. Laplace Approximation for EPCVM

In [2] an extension of PCVM was proposed which calculates the same model but with an incremental set of basis functions employing automatic relevance determination. In the EPCVM the model is generalized by applying the logistic sigmoid link function $\sigma(x) = \frac{1}{1+\exp(-x)}$, and adopting the Bernoulli distribution for $p(\mathbf{t}|\mathbf{w})$, the likelihood is then written as follows:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} \sigma_n^{t_n} \left[1-\sigma_n\right]^{1-t_n},$$

where $\sigma_n = \sigma\left(\lambda \sum_{i=0}^{N} w_i \phi_{y_i}(\mathbf{x}_n)\right)$ and we assume $y_0 = 1$ to facilitate the representation, $\mathbf{t} = (t_1,\cdots,t_N)^T$ is a vector of targets, $t_n = \frac{y_n+1}{2} \in \{0,1\}$ is the probabilistic target.

According to Bayes' theorem, the posterior distribution of weights $\mathbf{w}$ can be obtained with the current values of α as follows:

$$p(\mathbf{w}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)}{p(\mathbf{t}|\alpha)}.$$

with $p(\mathbf{w}|\alpha)$ given in Eq. (2) and $p(\mathbf{t}|\alpha) = \prod_{i=0}^{N} \exp\left(\frac{\alpha_i \mathbf{w}_i^2}{2}\right)$, where the bias term has been included for $i = 0$. After incorporating the truncated Gaussian prior, the integral in Bayesian inference is intractable. In order to obtain the posterior, Laplace approximation will be employed to approximate the posterior. Laplace approximation is a deterministic approximation algorithm using a Gaussian to represent a given probability.

The most probable weight setting under the posterior, MAP estimate of $\mathbf{w}$, $\mathbf{w}_{MAP}$ can be obtained by maximizing the log of $p(\mathbf{w}|\mathbf{t})$ with respect to the parameters $\mathbf{w}$:

$$\begin{aligned} Q &= \log\{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)\} - \log p(\mathbf{t}|\alpha) \\ &= \sum_{n=1}^{N}[t_n \log\sigma_n + (1-t_n)\log(1-\sigma_n)] - \frac{1}{2}\sum_{i=0}^{N}\alpha_i w_i^2 \\ &\quad + \sum_{i=1}^{N}\log\delta(w_i) - \text{const.} \end{aligned}$$

As the indicator function $\delta(\cdot)$ is not differentiable, a sigmoid link function with $\beta = 3$ is employed to replace it, i.e. approximate $\delta(w_i)$ by $\xi_\beta(w_i) = \sigma(\beta w_i)$, the gradient is

$$\frac{\partial Q}{\partial \mathbf{w}} = \lambda\Phi^T(\mathbf{t}-\sigma) - \mathbf{A}\mathbf{w} + \mathbf{k},$$

where $\sigma = [\sigma_1,\cdots,\sigma_N]^T$, $\sigma_n = \sigma\left(\lambda\sum_{i=0}^{N} w_i\phi_{y_i}(\mathbf{x}_n)\right)$, $\mathbf{A} = diag(\alpha_0,\alpha_1,\cdots,\alpha_N)$ is the $(N+1)\times(N+1)$ diagonal matrix, $\mathbf{k} = [0,\beta(1-\sigma(\beta w_1)),\cdots,\beta(1-\sigma(\beta w_N))]^T$ is the $N+1$ vector.

Setting the gradient to zero and we obtain

$$\mathbf{w}_{MAP} = \mathbf{A}^{-1}\left(\lambda\mathbf{\Phi}^T(\mathbf{t}-\sigma) + \mathbf{k}\right). \quad (10)$$

The Hessian can be explicitly computed as follows:

$$\frac{\partial^2 Q}{\partial \mathbf{w}^2} = -(\mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi} + \mathbf{A} + \mathbf{D}),$$

where $\mathbf{B} = diag(b_1,\cdots,b_N)$ and $\mathbf{D}$ are diagonal matrices, where $b_i = \lambda^2\sigma_n(1-\sigma_n)$ and $\mathbf{D} = diag(0,d_1,\cdots,d_N) = diag(0,\sigma(\beta w_1)(1-\sigma(\beta w_1))\beta^2,\cdots,\sigma(\beta w_N)(1-\sigma(\beta w_N))\beta^2)$, respectively.

Hence, the posterior covariance is

$$\mathbf{\Sigma}_{MAP} = (\mathbf{\Phi}^T\mathbf{B}\mathbf{\Phi} + \mathbf{A} + \mathbf{D})^{-1}. \quad (11)$$

By incorporating the indicator function, i.e. $\mathbf{k}$ and $\mathbf{D}$ in Equations (10) and (11) one prevents the weight from negative values, i.e. complying with truncated prior. A more detailed derivation is given in [2].

C. Hyperparameters Optimization for EPCVM

Originally the PCVM was optimized by a *top-down* approach including all basis functions in the beginning, and then pruning irrelevant basis functions when the corresponding $\alpha_n's$ tending to infinity. However, the *top-down* approach will typically consume a lot of computational resources, especially in the beginning of the training. In order to make the algorithm more computationally efficient in [2] a *constructive* approach was proposed, based on marginal likelihood maximization to

include basis functions step by step starting from an empty model.

The previous sections presented the training algorithm of EPCVM with fixed hyperparameter α. In order to sequentially update α for a practical algorithm, we can maximize the type-II marginal likelihood $p(\mathbf{D}|\alpha)$. The fast algorithm to optimize the type-II marginal likelihood is to decompose $p(\mathbf{D}|\alpha)$ into two parts, one part denoted by $p(\mathbf{D}|\alpha_{\setminus i})$, that does not depend on α_i and another that does, i.e.,

$$p(\mathbf{D}|\alpha) = p(\mathbf{D}|\alpha_{\setminus i}) + l(\alpha_i), \tag{12}$$

where $l(\alpha_i)$ is a function that depends on α_i.

The updating rule for α_i can be obtained with the derivation of marginal likelihood [7]. The procedure leads to a practical algorithm for optimizing the hyperparameters that has significant speed advantages. To make this approach scale invariant each basic function has to be normalized to one (the columns of ϕ). As a consequence the design matrix becomes asymmetric. Further this matrix is also of large scale leading to substantial memory consumption. Subsequently we review the Nyström approximation as an efficient technique for low-rank matrix approximation. The classical Nyström approximation is defined only for symmetric positive semi definite (psd) matrices and we provide a derivation also for asymmetric (potentially rectangular) matrices.

D. Nyström approximation

The Nyström approximation technique has been proposed in the context of kernel methods in [4] and is used in the following to derive an approximated EPCVM called Ny-EPCVM. Here, we give a short review of this technique before it is employed in EPCVM. One well known way to approximate a $N \times N$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel matrix $K = U\Lambda U^T$, where U is a matrix, whose columns are orthonormal eigenvectors, and Λ is a diagonal matrix consisting of eigenvalues $\Lambda_{11} \geq \Lambda_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\tilde{K} \approx U_{N,m}\Lambda_{m,m}U_{m,N}$, where the indices refer to the size of the corresponding submatrix restricted to the largest m eigenvalues. The Nyström method approximates a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(N^3)$ operation.

By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions φ_i and non negative eigenvalues λ_i in the form

$$k(\mathbf{x}, \mathbf{y}) = \sum_i \lambda_i \varphi_i(\mathbf{x})\varphi_i(\mathbf{y}).$$

The eigenfunctions and eigenvalues of a kernel are defined as the solution of the integral equation

$$\int k(\mathbf{y}, \mathbf{x})\varphi_i(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \lambda_i\varphi_i(\mathbf{y}),$$

where $p(\mathbf{x})$ is the probability density of $\mathbf{x}$. This integral can be approximated based on the Nyström technique by an i.i.d. sample $\{\mathbf{x}^k\}_{k=1}^m$ from $p(\mathbf{x})$:

$$\frac{1}{m}\sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}^k)\varphi_i(\mathbf{x}^k) \approx \lambda_i\varphi_i(\mathbf{y}).$$

Using this approximation we denote with $K^{(m)}$ the corresponding $m \times m$ Gram sub-matrix and get the corresponding matrix eigenproblem equation as:

$$K^{(m)}U^{(m)} = U^{(m)}\Lambda^{(m)}$$

with $U^{(m)} \in \mathbb{R}^{m\times m}$ is column orthonormal and $\Lambda^{(m)}$ is a diagonal matrix.

Now we can derive the approximations for the eigenfunctions and eigenvalues of the kernel k

$$\lambda_i \approx \frac{\lambda_i^{(m)} \cdot N}{m}, \quad \varphi_i(\mathbf{y}) \approx \frac{\sqrt{m/N}}{\lambda_i^{(m)}}\mathbf{k}_y^\top \mathbf{u}_i^{(m)}, \tag{13}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $U^{(m)}$. Thus, we can approximate φ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}^1, \mathbf{y}), ..., k(\mathbf{x}^m, \mathbf{y}))$. For a given $N \times N$ Gram matrix K we randomly choose m rows and respective columns. The corresponding indices are called landmarks, and should be chosen such that the data distribution is sufficiently covered. A specific analysis about selection strategies was recently given in [8]. We denote these rows by $K_{m,N}$. Using the formulas (13) we obtain $\tilde{K} = \sum_{i=1}^m 1/\lambda_i^{(m)} \cdot K_{m,N}^T(\mathbf{u}_i^{(m)})^T(\mathbf{u}_i^{(m)})K_{m,N}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Thus we get, $K_{m,m}^{-1}$ denoting the inverse[2],

$$\tilde{K} = K_{N,m}K_{m,m}^{-1}K_{m,N}. \tag{14}$$

as an approximation of K. This approximation is exact, if $K_{m,m}$ has the same rank as K.

E. A Nyström approximation for asymmetric matrices

The original Nyström approximation was proposed for psd symmetric kernel matrices only [4] with approximation bounds detailed recently in [9]. An asymmetric Nyström approximation can be derived in various ways. Here we show an approach which directly links back to a symmetric approximation, keeping the known approximation guarantees and another more simple approach, already discussed in [10], which works on the generalized Nyström approximation [11]. The first strategy makes use of a singular value decomposition (SVD). For an arbitrary matrix K an SVD can be constructed such that $K = U \cdot S \cdot V'$, where U and V contain the so called left or right singular vectors of K and S contains the singular values. This decomposition can be also obtained for asymmetric matrices.

The singular value decomposition based on a *Nyström approximated* similarity matrix $\tilde{K} = K_{Nm}K_{m,m}^{-1}K_{Nm}^\top$ with m landmarks, calculates the left singular vectors of $\tilde{K}$ as the eigenvectors of $\tilde{K}\tilde{K}^\top$ and the right singular vectors of $\tilde{K}$ as the eigenvectors of $\tilde{K}^\top\tilde{K}$. The matrices $\tilde{K}\tilde{K}^\top$ and $\tilde{K}^\top\tilde{K}$ are obviously symmetric matrices which can be approximated by the regular Nyström approximation with the guarantees as shown in [9].

The non-zero singular values of $\tilde{K}$ are then found as the square roots of the non-zero eigenvalues of both $\tilde{K}^\top\tilde{K}$ or $\tilde{K}\tilde{K}^\top$. Accordingly one only has to calculate a new

[2]If $K_{m,m}^{-1}$ has not full rank, e.g. due to identical points in the training data, one may also use the Moore-Penrose pseudoinverse

Nyström approximation of the matrix $\tilde{K}\tilde{K}^\top$ using e.g. the same landmark points as for the input matrix K. Subsequently an eigenvalue decomposition (EVD) is calculated on the approximated matrix $\zeta = \tilde{K}\tilde{K}^\top$ as shown later on. This eigenvalue calculation should not be based on the small matrix $K_{m,m}$ which can lead to inaccuracies due to the sub-sampling, nor on the full matrix K which would be very costly. Instead it is desirable to use a slightly more complicated way but with the benefit of providing exact [3] estimates of the eigenvalues.

For a matrix approximated by Eq. (14) it is possible to compute its exact eigenvalue decomposition in *linear* time [4]. To compute the eigenvectors and eigenvalues of a potentially *indefinite* matrix we first compute its squared form, since the eigenvectors in the squared matrix stay the same and only the eigenvalues are squared.

Let K be a psd similarity matrix, for which we can write its decomposition as

$$\begin{aligned}\tilde{K} &= K_{N,m}K_{m,m}^{-1}K_{m,N} \\ &= K_{N,m}U\Lambda^{-1}U^\top K_{N,m}^\top \\ &= BB^\top,\end{aligned}$$

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

$$\begin{aligned}\tilde{K}^2 &= BB^\top BB^\top \\ &= BVAV^\top B^\top,\end{aligned}$$

where V and A are the eigenvectors and eigenvalues of $B^\top B$, respectively. The corresponding eigenequation can be written as $B^\top Bv = av$. Multiplying it with B from left we get the eigenequation for $\tilde{K}$

$$\underbrace{BB^\top}_{\tilde{K}}\underbrace{(Bv)}_{u} \quad = \quad a\underbrace{(Bv)}_{u}$$

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

$$\begin{aligned}\tilde{K} &= BVV^\top B^\top \\ &= BVA^{-1/2}AA^{-1/2}V^\top B^\top \\ &= CAC^\top,\end{aligned}$$

where we defined $C = BVA^{-1/2}$ as the matrix of orthonormal eigenvectors of K. The eigenvalues of $\hat{K}$ can be obtained using $A = C^\top \hat{K} C$.

Using the above introduce Nyström based SVD and EVD we can represent an asymmetric matrix K by a Nyström approximated SVD.

An alternative derivation is based on analyzing the submatrices involved in the standard Nyström approximation leading to the generalized Nyström approximation [11], by taking analogies into account [5] One finally obtains a formulation similar to the original Nyström approximation such that an asymmetric kernel matrix K is approximated as

$$\tilde{K} = K_{N,m}K_{m,m}^{-1}K_{m,N}$$

with $K_{N,m} \neq K_{m,N}^\top$. Regardless of the chosen approach we have three different matrices which for the SVD approach are subsequently denoted as $K1 = U, \quad K2 = S, \quad K3 = V^\top$ or in case of the generalized Nyström will be denoted as $K1 = K_{N,m}, \quad K2 = K_{m,m}^{-1}, \quad K3 = K_{m,N}$, all having a low rank m. It should be noted that the presented approach can be also used for the approximation of (asymmetric) squared and non-squared input matrices.

F. EPCVM for large scale proximity data

The EPCVM parameters are optimized using the laplacian approximation to iteratively adapt the weight vector $\mathbf{w}$ during learning by keeping only those basis function which are relevant for the model. We will now show multiple modifications of EPCVM to integrate the Nyström approximation and to ensure that the memory and runtime complexity remains linear at all time. We refer to our method as Ny-EPCVM. First we consider the normalization of the large quadratic input design matrix[6] ϕ. This matrix leads to a complexity of $\mathcal{O}(N^2)$ and therefore we would like to approximate it by a lower rank representation.

To ensures that all basis function are equally treated in the relevance learning scheme used in EPCVM, the matrix ϕ is normalized leading to an asymmetric matrix ϕ'. This normalization is in fact common also for other similar methods, e.g. the original RVM and hence the approach is more generic and not restricted to EPCVM. The matrix ϕ' can not any longer be approximated by the classical Nyström approximation but one of the schemes as discussed before has to be used.

In the preprocessing the matrix ϕ is normalized to unit length

$$\rho_j \quad = \quad \sqrt{\sum_i \phi_{i,j}^2} \qquad \phi'_{\cdot,j} = \frac{\phi_{\cdot,j}}{\rho_j}$$

The calculation of the scaling parameters ρ_j requires an operation on the full matrix ϕ this can be avoided by using a Nyström approximation of the input matrix ϕ (with squared entries) and by calculating the approximated scaling coefficients ρ^* in matrix notation as:

$$\rho^* = \sqrt{K3^\top \cdot (\sum_i (K1 \cdot K2)_{[i,\cdot]})^\top}$$

A corresponding approximated and normalized matrix $\phi'^{,*}$ is obtained by deriving new Nyström matrices as shown before but with scaled landmark columns based on the vector ρ^*. It should be noted that the obtained matrix is asymmetric.

Another source of squared complexity in EPCVM is the parameter initialization. In EPCVM the basis function is set

[3] Within the Nyström framework.

[4] The approximation is indeed in linear time given the matrix has low rank, which is the standard assumption for the Nyström approximation.

[5] The work in [11] however lacks a specific proof.

[6] In general the original design matrix ϕ is just a symmetric kernel matrix of the input data

to the largest projection with the target values. This operation reads as:

$$\pi = \phi^\top \cdot t$$

and is approximated by

$$\pi \approx K3^\top \cdot (\sum_i ((K1 \otimes t) \cdot K2)_{[i,\cdot]})^\top;$$

where $\otimes$ indicates the scaling of all columns of $K1$ by t and the matrices $K1, K2$ and $K3$ are obtained from $\phi'^{,*}$.

Also in the calculation of the quality factor $\mathcal{Q}$ and the sparsity factor $\mathcal{S}$ see [3] the full basis function matrix ϕ is used. The original $\mathcal{S}$ is calculated (in matrix notation) as:

$$\begin{aligned} \beta_{pp} &= ((\phi_c \otimes \beta)^\top \cdot \phi')^\top \\ \mathcal{S} &= (\beta^\top \cdot (\phi.^2))^\top - \sum_j (\beta_{pp} \cdot U^{-1}).^2 \end{aligned}$$

with ϕ_c as the current set of included basis functions and $.^2$ indicating element wise square and U is the upper triangle matrix of the Hessian matrix Q as defined before. The quality factor $\mathcal{Q}$ is calculated mainly by:

$$\mathcal{Q} = (t - y)^\top \cdot \phi'$$

$\mathcal{Q}$ is related to how well the basis function contributes to reducing the error. $\mathcal{S}$ is related to how orthogonal a given basis function is to the currently used set of basis functions. Again the Nyström approximation can be employed to calculate the approximated terms with linear costs.

The approximation for $\mathcal{S}$ can be written as:

$$\begin{aligned} \beta_{pp} &= ((\phi_c \otimes \beta)^\top \cdot K1 \cdot (K2 \cdot (K3^\top)^\top))^\top \\ \mathcal{S} &= (K3^{2\top} \cdot (\sum_i \beta \otimes K1^{2\top}) \cdot K2^{-2})^\top - \sum_j (\beta_{pp} \cdot U^{-1}).^2 \end{aligned}$$

the matrices $K1, K2$ and $K3$ are obtained from $\phi'^{,*}$ as well as the matrices $K1^2, K2^2$ and $K3^2$ but with element-wise squared entries of $\phi'^{,*}$. The approximation of the quality factor $\mathcal{Q}$ can be obtained as:

$$\mathcal{Q} \approx K3^\top \cdot (\sum_i ((t - y) \otimes K1) \cdot K2)^\top$$

The operations used to calculate $\mathcal{Q}$ and $\mathcal{S}$ as shown before can now be done without any $N \times N$ matrices. It should be noted that these operations can also be used in other algorithms, like the RVM, which are based on the same sparse relevance learning scheme.

III. Complexity analysis

Classical Support Vector Machine (SVM) algorithms [12] and PCVM have a time complexity of $\mathcal{O}(N^3)$, where N is the number of training points, but the computational complexity of SVMs can be reduced to approximately $\mathcal{O}(N^{2.1})$ for sequential minimal optimization (SMO) like algorithms [13], which breaks the large quadratic programming (QP) problem into a series of smallest possible QP problems. In [14] an approximated solution for SVM was proposed using core sets which is called Core Vector Machine (CVM)[7]. This approach scales to million of points modeling the SVM problem by means of a minimum enclosing ball optimization problem. The only extra parameter is the error tolerance which we have set to $\epsilon = 0.01$. An efficient, Nyström based, SVM for psd kernels was proposed recently in [15] (LLSVM), scaling quadratic in the number of landmarks. But our objective is on *generic*, *probabilistic* output models as detailed later on which can not be obtained by LLSVM or CVM. The experiments to CVM are given to show that EPCVM and Ny-EPCVM perform competitive in the prediction accuracy for standard psd data. Further we show that the runtime of Ny-EPCVM is substantially better for larger data sets in comparison to EPCVM and scales reasonable compared to optimized SVM approaches.

The original EPCVM update rules of the relevance parameters involve matrix vector operations on the whole kernel matrix which has a computational complexity $\mathcal{O}(N^2)$ and memory storage $\mathcal{O}(N^2)$, where N is the number of samples. In the former derivation of Ny-EPCVM we have replaced all quadratic operations of EPCVM by a corresponding Nyström approximation of these operations. As the Nyström approximation is linear if the number of landmarks $M \ll N$ also the Ny-EPCVM has now a complexity of $\mathcal{O}(N)$.

IV. Experiments

First, we present experimental results for different medium size standard datasets using the Nyström approximated EPCVM (Ny-EPCVM) and compare to EPCVM and CVM. These experiments will link the new approach to classical techniques to show that the approximation does not significantly reduce the prediction accuracy. Then we show results for the application to Ny-EPCVM on different relatively large data sets which in general can not be processed by the original EPCVM formulation.

A. Benchmark data sets

In order to evaluate the performance of the Ny-EPCVM we compare different standard algorithms on seven benchmark datasets[8] which are sufficiently large to motivate the Ny-EPCVM approach but are still small enough to be analyzed with the considered alternative approaches. We consider the banana (5300pts,2dims), diabetes (768pts,8dims), image (2086pts,18dims), ringnorm (7400pts,20dims), splice (2991pts,60dims), twonorm (7400pts,20dims) and waveform (5000pts,21dims) dataset given in two classes. All data are normalized to have 0 mean and unit variance, neglecting constant input dimensions and without cases of missing values. To simplify the evaluation process and to ease the reproduction of the experiments we use a defacto parameter free extreme learning machine kernel (ELM) as suggested in [16] to represent the data in the kernel space. For Ny-EPCVM we use 100 randomly sampled landmarks (new sampled in the repeats) for each dataset. For all experiments we report mean and standard errors and runtimes as obtained by a 10 fold crossvalidation with 10 repeats.

As can be seen from Table I and II the Ny-EPCVM does not sacrifice accuracy for speed[9]. It is also not very sensitive to

[7] We use the code as provided at http://www.c2i.ntu.edu.sg/ivor/cvm.html

[8] Taken from http://www.raetschlab.org/Members/raetsch/benchmark

[9] Best results are highlighted in bold, significant better ones are marked with a $\star$.

TABLE I. TEST SET ACCURACY (% ± STD) OF UCI BENCHMARKS FOR TWO CLASSES PROBLEMS FOR NY-PCVM

two-class UCI data	Ny-EPCVM	EPCVM	CVM
banana	86.62 ± 2.46	83.25 ± 4.18	$\mathbf{90.26 \pm 0.97}*$
diabetes	70.44 ± 3.30	71.35 ± 2.21	$\mathbf{76.84 \pm 5.16}$
image	88.06 ± 1.45	$\mathbf{96.55 \pm 0.71}*$	94.50 ± 1.36
ringnorms	95.95 ± 0.73	$\mathbf{98.55 \pm 0.45}$	98.54 ± 0.36
splice	83.98 ± 1.72	83.85 ± 1.76	$\mathbf{85.69 \pm 1.45}$
twonorm	97.62 ± 0.60	$\mathbf{97.84 \pm 0.24}$	97.82 ± 0.56
waveform	81.68 ± 2.66	90.68 ± 0.84	$\mathbf{91.32 \pm 1.06}$

TABLE II. RUNTIME (% ± STD) OF UCI BENCHMARKS FOR TWO CLASSES PROBLEMS FOR NY-PCVM

two-class UCI data	Ny-EPCVM	EPCVM	CVM
banana	13.56 ± 3.46	39.89 ± 3.58	4.3 ± 0.12
diabetes	0.78 ± 0.58	1.93 ± 0.34	0.14 ± 0.02
image	19.29 ± 2.62	30.93 ± 3.29	0.65 ± 0.02
ringnorms	42.74 ± 1.20	232.21 ± 28.09	6.26 ± 0.05
splice	43.85 ± 1.22	17.60 ± 3.60	2.11 ± 0.02
twonorm	77.87 ± 12.06	99.00 ± 22.52	3.09 ± 0.00
waveform	46.24 ± 5.24	131.20 ± 3.31	3.47 ± 0.01

initialization effects as reflected by the in general low standard deviations which are similar to those of CVM. The prediction accuracies on the test data are comparable to those of EPCVM. For the image and waveform dataset the accuracy dropped down in comparison to EPCVM. For the image data we can explain this by the large number of potential clusters in this segmentation dataset which where not sufficiently represented by the rather small number of landmarks. The comparison with CVM clearly shows competitive behaviour for both methods. The runtime (given in seconds for a single model) of the Ny-EPCVM is in general much better than for EPCVM. The CVM has a runtime complexity independent of N, if probabilistic sampling is used [14], but is constrained to metric inputs.

A runtime analysis of EPCVM, CVM and Ny-EPCVM is given in Figure 1 at logarithmic scale.

In a further experiment we consider non-vectorial data given by means of indefinite kernels which have not yet been considered for the EPCVM but are of wide interest [17]. See [18] for a recent survey on this topic. Indefinite kernels are often obtained from domain specific measures like alignment functions, shape measures or other score-functions [19], [20]. In contrast to many standard kernel approaches like

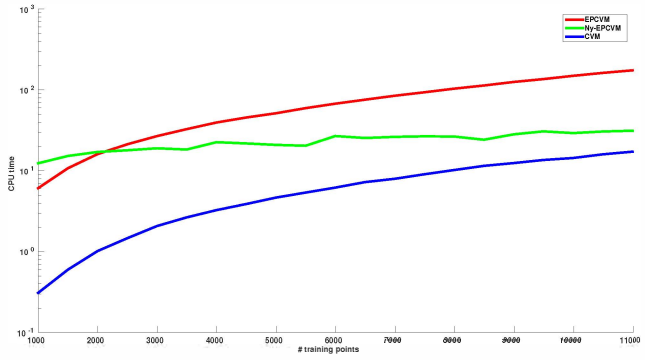

Fig. 1. CPU time at logarithmic scale for a larger dataset for EPCVM, CVM and Ny-EPCVM. For Ny-EPCVM

the CVM, for EPCVM, the indefinite kernel matrices need not to be corrected by costly eigenvalue correction [21][10] Further the EPCVM provides direct access to probabilistic classification decisions. We compare to the indefinite kernel fisher discriminant (iKFD) [24] as the state of the art method in this field [11]. However the iKFD is a (in general) non-sparse approach which is not only costly during the model generation with $\mathcal{O}(N^3)$ complexity, but also in the out-of-sample extension. The indefinite datasets have a size which can still be processed using iKFD. EPCVM on the other hand is a very sparse method as already outlined in [2] and for new samples only the similarities to the very few basis functions used in the model have to be calculated.

The data sets are: *Sonatas* (1068pts, 5 classes) taken from [25]. It is comprised of pairwise dissimilarities between 1,068 sonatas from the classical period (by Beethoven, Mozart and Haydn) and the baroque era (by Scarlatti and Bach). The musical pieces were given in the MIDI file format, taken from the online MIDI collection *Kunst der Fuge*[12]. Their mutual dissimilarities were measured with the normalized compression distance (NCD), see [26].

Gesture (1500pts, 20 classes), taken from [27] is a set of dissimilarities generated from a sign-language interpretation problem. The gestures are measured by two video cameras observing the positions of the two hands in 75 repetitions of creating 20 different signs. The dissimilarities are computed using a dynamic time warping procedure on the sequence of positions [28].

Zongker (2000pts, 10 classes) digit dissimilarity data (2000 points in 10 classes) from [27] is based on deformable template matching. The dissimilarity measure was computed between 2000 handwritten NIST digits in 10 classes, with 200 entries each, as a result of an iterative optimization of the non-linear deformation of the grid [29].

Proteom (2604pts, 53 classes) which contains a comprehensive set of protein families and appeared first in the work of [30]. The pairwise structural alignments are computed by [30]. Each sequence belongs to a group labeled by experts, here we use the data as provided in [27].

Chromosom (4200pt, 21 classes) from [31] constitute a benchmark from cytogenetics. 4,200 human chromosomes from 21 classes are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. The string indicates the thickness of the gray levels of the image. These strings can be directly compared using the edit distance based on the differences of the numbers and insertion/deletion costs 4.5 [31]. The classification problem is to label the data according to the chromosome type.

All data are processed as indefinite kernels with 100 landmarks if not stated otherwise[13]. For all experiments we report mean and standard errors as obtained by a 10 fold crossvalidation. The probabilistic outputs can be directly used

[10]More recently the authors have proposed a fast eigenvalue correction for low rank matrices [22], [23], but these corrections are not always appropriate.

[11]The few SVM based kernel methods available for indefinite kernels use a proxy approach not scaling to larger data and are not probabilistic.

[12]http://www.kunstderfuge.com

[13]Dissimilarities have been converted to similarities by double centering using the approach given in [32]

TABLE III. ACCURACIES - INDEFINITE KERNELS

	Ny-EPCVM	EPCVM	iKFD
Sonatas	83.90 ± 2.0	$\mathbf{84.58 \pm 2.8}*$	80.54 ± 0.03
gesture	92.53 ± 1.2	91.20 ± 1.8	$\mathbf{98.07 \pm 0.7}*$
zongker	87.65 ± 0.9	$\mathbf{97.70 \pm 1.11}$	96.95 ± 0.1
proteom	94.70 ± 1.64	96.24 ± 0.98	$\mathbf{99.35 \pm 0.8}$
chromosom	92.00 ± 1.40	95.86 ± 0.44	$\mathbf{97.29 \pm 0.7}$

TABLE IV. RUNTIMES - INDEFINITE KERNELS

	Ny-EPCVM	EPCVM	iKFD
Sonatas	12.46 ± 1.0	13.69 ± 1.78	40.11 ± 0.5
gesture	12.32 ± 0.3	12.65 ± 0.75	69.38 ± 7.5
zongker	13.09 ± 1.3	14.12 ± 3.00	74.22 ± 6.9
proteom	14.26 ± 0.93	30.30 ± 3.30	758.90 ± 28.4
chromosom	29.97 ± 0.7	37.81 ± 0.9	1073.9 ± 26.2

to allow for a reject region but can also be used to provide alternative classification decisions e.g. in a ranking framework

In Table III and Table IV we show the results for different non-metric proximity datasets using Ny-EPCVM, EPCVM and iKFD. We observe that the prediction accuracy of iKFD is better compared to Ny-EPCVM on the non-metric proximity data. The main reason for this effect can be found if we consider the model complexity. For iKFD basically all training points are used in the model $\geq 97\%$ whereas for Ny-EPCVM only less than 0.3% are kept. In practice it is often costly to calculate the non-metric proximity measures like sequence alignments and accordingly sparse models are very desirable. Considering the runtime Ny-EPCVM is again faster although not so pronounced as before since the number of points per class are smaller then for the benchmarks. The speed-up compared to iKFD is obvious. For non-psd data Ny-EPCVM is obviously substantially better in runtime and sparsity compared to the state of the art approach while showing good prediction accuracy.

V. CONCLUSIONS

In this paper we presented an alternative formulation of the probabilistic classification vector machine employing the Nyström approximation. Here we also provided an alternative derivation of a Nyström approximation for asymmetric matrices based on a Nyström approximated SVD. We found that Ny-EPCVM is competitive in the prediction accuracy with EPCVM and alternative approaches while taking substantially less memory and runtime. For a variety of benchmark data the Ny-EPCVM performed competitive to EPCVM and CVM. Additionally Ny-EPCVM and EPCVM can also be applied to non-mercer kernels which make them available for moderate to larger scale indefinite proximity data. Ny-EPCVM and EPCVM showed quite good results with respect to iKFD but with a much sparser model. The complexity of iKFD is cubic whereas for EPCVM we have squared and for Ny-EPCVM linear complexity. Further, the EPCVM and Ny-EPCVM are full probabilistic classifiers with obviously good performance on a broad spectrum of problems. Now the effective EPCVM technique is also available for even larger data sets. Although Ny-EPCVM shows a large number of benefits there are also some shortcomings. The major underlying assumption of Ny-EPCVM is an intrinsically low dimensional feature space, scaling with the rank of the Nyström approximation. If proximity matrices have a high intrinsic dimension the number of landmarks has to be large and the benefits of Ny-EPCVM are reduced. Also if the datasets are rather small with some hundred points one better takes the original EPCVM or even PCVM since the approximation effects of the Ny-EPCVM may have a negative effect on the results. The Ny-EPCVM provides now an effective way to obtain a *probabilistic* classification model for medium to large datasets with *linear* runtime and memory complexity.

ACKNOWLEDGMENT

A Marie Curie Intra-European Fellowship (IEF): FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) is gratefully acknowledged. PT was supported by the EPSRC grant EP/L000296/1, "Personalized Health Care through Learning in the Model Space". We would like to thank Andrej Gisbrecht for support in former work related to the paper.

REFERENCES

[1] H. Chen, P. Tino, and X. Yao, "Probabilistic classification vector machines," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 901–914, 2009.

[2] ——, "Efficient probabilistic classification vector machine with incremental basis function selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 356–369, 2014.

[3] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001. [Online]. Available: http://dx.doi.org/10.1162/15324430152748236

[4] C. K. I. Williams and M. 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*, 2000, pp. 682–688.

[5] X. Nguyen, L. Huang, and A. D. Joseph, "Support vector machines, data reduction, and approximate kernel matrices," in *ECML/PKDD (2)*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds., vol. 5212. Springer, 2008, pp. 137–153.

[6] M. Vladymyrov and M. Á. Carreira-Perpiñán, "Locally linear landmarks for large-scale manifold learning," in *ECML/PKDD (3)*, ser. Lecture Notes in Computer Science, H. Blockeel, K. Kersting, S. Nijssen, and F. Železný, Eds., vol. 8190. Springer, 2013, pp. 256–271.

[7] A. Faul and M. Tipping, "Analysis of sparse bayesian learning," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 383–389.

[8] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved Nystrom low-rank approximation and error analysis," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1232–1239.

[9] A. Gittens and M. W. Mahoney, "Revisiting the Nyström method for improved large-scale machine learning," *CoRR*, vol. abs/1303.1849, 2013.

[10] J. Wang, Y. Dong, X. Tong, Z. Lin, and B. Guo, "Kernel Nyström method for light transport," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 29:1–29:10, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1531326.1531335

[11] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, "A theory of pseudoskeleton approximations," *Linear Algebra and its Applications*, vol. 261, no. 13, pp. 1 – 21, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379596003011

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

[13] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization." Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[14] I. W.-H. Tsang, J. T.-Y. Kwok, and J. M. Zurada, "Generalized core vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 5, pp. 1126–1140, 2006.

[15] K. Zhang, L. Lan, Z. Wang, and F. Moerchen, "Scaling up kernel svm on limited resources: A low-rank linearization approach," *JMLR - Proceedings Track*, vol. 22, pp. 1425–1434, 2012.

[16] B. Frénay and M. Verleysen, "Parameter-insensitive kernel in extreme learning for non-linear support vector regression," *Neurocomputing*, vol. 74, no. 16, pp. 2526–2531, 2011.

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

[18] F.-M. Schleif and P. Tino, "Indefinite proximity learning - a review," *Neural Computation*, vol. to appear, 2015.

[19] M. Strickert, K. Bunte, F. Schleif, and E. Hüllermeier, "Correlation-based embedding of pairwise score data," *Neurocomputing*, vol. 141, pp. 97–109, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2014.01.049

[20] H. Ling and D. W. Jacobs, "Using the inner-distance for classification of articulated shapes," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 2005, pp. 719–726. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.362

[21] Y. Chen, E. K. Garcia, M. R. Gupta, A. Rahimi, and L. Cazzanti, "Similarity-based classification: Concepts and algorithms," *JMLR*, vol. 10, pp. 747–776, 2009.

[22] A. Gisbrecht and F. Schleif, "Metric and non-metric proximity transformations at linear costs," *CoRR*, vol. abs/1411.1646, 2014. [Online]. Available: http://arxiv.org/abs/1411.1646

[23] A. Gisbrecht and F.-M. Schleif, "Metric and non-metric proximity transformations at linear costs," *Neurocomputing*, vol. to appear, 2015.

[24] E. Pekalska and B. Haasdonk, "Kernel discriminant analysis for positive definite and indefinite kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1017–1032, 2009.

[25] B. Mokbel, A. Hasenfuss, and B. Hammer, "Graph-based representation of symbolic musical data," in *Graph-Based Representations in Pattern Recognition, 7th IAPR-TC-15 International Workshop, GbRPR 2009, Venice, Italy, May 26-28, 2009. Proceedings*, ser. Lecture Notes in Computer Science, A. Torsello, F. Escolano, and L. Brun, Eds., vol. 5534. Springer, 2009, pp. 42–51. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02124-4_5

[26] R. Cilibrasi and P. M. B. Vitányi, "Clustering by compression," *IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1523–1545, 2005.

[27] R. P. Duin, "PRTools," march 2012. [Online]. Available: http://www.prtools.org

[28] J. Lichtenauer, E. Hendriks, and M. Reinders, "Sign language recognition by combining statistical dtw and independent classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 2040–2046, 2008.

[29] A. K. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1386–1391, Dec. 1997.

[30] V. Roth, J. Laub, J. M. Buhmann, and K.-R. Müller, "Going metric: Denoising pairwise data," in *NIPS*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 817–824.

[31] M. Neuhaus and H. Bunke, "Edit distance based kernel functions for structural pattern classification," *Pattern Recognition*, vol. 39, no. 10, pp. 1852–1863, 2006.

[32] F. Schleif and A. Gisbrecht, "Data analysis of (non-)metric proximities at linear costs," in *Similarity-Based Pattern Recognition - Second International Workshop, SIMBAD 2013, York, UK, July 3-5, 2013. Proceedings*, ser. Lecture Notes in Computer Science, E. R. Hancock and M. Pelillo, Eds., vol. 7953. Springer, 2013, pp. 59–74. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-39140-8_4