ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Probabilistic Classification Vector Machine at large scale

Frank-Michael Schleif[1], Andrej Gisbrecht[2], Peter Tino[1]

1 - The University of Birmingham, School of Computer Science,
Edgbaston Birmingham B15 2TT, United Kingdom.
2 - University of Bielefeld, Theoretical Computer Science,
33619, Bielefeld, Germany.

Abstract. Probabilistic kernel classifiers are effective approaches to solve classification problems but only few of them can be applied to indefinite kernels as typically observed in life science problems and are often limited to rather small scale problems. We provide a novel batch formulation of the Probabilistic Classification Vector Machine for large scale metric and non-metric data.

1 Introduction

The Probabilistic Classification Vector Machine (PCVM) [2, 3] is a sparse probabilistic kernel classifier pruning unused basis functions during training and a full probabilistic classifiers which can be used for arbitrary positive definite *and* indefinite symmetric kernel matrices. Here we propose a runtime and memory efficient formulation with linear complexity using the Nyström matrix approximation [11], which is exact if the rank of the matrix equals the number of independent landmarks points. We also consider for the first time PCVM with indefinite kernel matrices which are common e.g. in the life science. First we review PCVM, then present Ny-PCVM, ensuring linear memory and runtime complexity at good generalization ability, shown on various data sets.

2 Probabilistic Classification Vector Learning for large scale

PCVM uses a kernel regression model $\sum_{i=1}^{N} w_i \phi_{i,\theta}(\mathbf{x}) + b$ with a link function, with w_i being the weights of the basis functions $\phi_{i,\theta}(\mathbf{x})$ and b as a bias term. The basis functions corresponds 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 EM implementation of PCVM [3] 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 $\mathcal{N}(0,1)$. We get:$l(\mathbf{x};\mathbf{w},b) = \Psi\left(\sum_{i=1}^{N} w_i \phi_{i,\theta}(\mathbf{x}) + b\right) = \Psi\left(\Phi_\theta(\mathbf{x})\mathbf{w} + b\right)$ Where $\Phi_\theta(\mathbf{x})$ is a vector of basis function evaluations for data item $\mathbf{x}$.

In the PCVM formulation [2], a truncated Gaussian prior N_t with support on $[0, \infty)$ and mode at 0 is introduced for each weight w_i and a zero-mean Gaussian prior is adopted for the bias b. The priors are assumed to be mutually independent. $p(\mathbf{w}|\alpha) = \prod_{i=1}^{N} p(w_i|\alpha_i) = \prod_{i=1}^{N} N_t(w_i|0, \alpha_i^{-1}), \quad p(b|\beta) = \mathcal{N}(b|0, \beta^{-1}), \delta(\cdot) = \mathbf{1}_{x>0}(x).$

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

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

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, we have:

$$h_\theta(\mathbf{x}) = \Phi_\theta(\mathbf{x})\mathbf{w} + b + \epsilon \geq 0, y = 1, \quad h_\theta(\mathbf{x}) = \Phi_\theta(\mathbf{x})\mathbf{w} + b + \epsilon < 0, y = -1 \quad (1)$$

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

$$\mathbf{w}^{\text{new}} = M(M\Phi_\theta^\top(\mathbf{x})\Phi_\theta(\mathbf{x})M + I_N)^{-1}M(\Phi_\theta^\top(\mathbf{x})\bar{\mathbf{H}}_\theta - b\Phi_\theta^\top(\mathbf{x})\mathbf{I}) \quad (2)$$

$$\mathbf{b}^{\text{new}} = t(1 + tNt)^{-1}t(\mathbf{I}^\top\bar{\mathbf{H}}_\theta - \mathbf{I}^\top\Phi_\theta(\mathbf{x})\mathbf{w}) \quad (3)$$

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

$$m_i = (\bar{\alpha}_i)^{-1/2} = \begin{cases} \sqrt{2}w_i & \text{if} \quad y_i w_i \geq 0 \\ 0 & \text{else} \end{cases} \quad (4)$$

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

2.1 Nyström approximation

The Nyström approximation for kernel methods (details in [11]) gives:

$$\tilde{K} = K_{N,m}K_{m,m}^{-1}K_{m,N}. \quad (5)$$

Thereby m (columns/rows) of the original kernel matrix have been selected as so called landmarks and $K_{m,m}^{-1}$ denotes the Moore-Penrose pseudoinverse of this landmark matrix. The approximation is exact, if $K_{m,m}$ has the same rank as K.

2.2 PCVM for large scale proximity data

The PCVM parameters are optimized using the EM algorithm to prune the weight vector $\mathbf{w}$ during learning and hence the considered basis functions representing the model. We will now show multiple modifications of PCVM 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-PCVM. Initially the Ny-PCVM algorithm makes use of the matrices $K_1 = K_{N,m}$ and $K_2 = K_{m,m}^{-1} \cdot K_1^\top$ obtained from the original kernel matrix using the Nyström landmark technique described above. Given a matrix$X$, we denote by $\hat{X}$ the matrix formed from X containing elements at indices that have not yet been pruned out of the weight vector $\mathbf{w}$. As an example, the matrices $\hat{K_1} = K_1^{\mathbf{w}\neq 0,\cdot}$, $\hat{K_2} = K_2^{\cdot,\mathbf{w}\neq 0}$ hold only those columns/rows of K_1 or K_2 not yet pruned out from the weight vector. We will use the same notation also for other variables. We denote the set of indices of m randomly selected landmarks by $[m]$. Finally, in contrast to the

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

original PCVM formulation [2], in our notation we explicitly use the data labels - for example, instead of vector $\Phi_\theta(\mathbf{x})$ we write $\Xi_\theta(\mathbf{x}) \circ \mathbf{y}$, where $\Xi_\theta(\mathbf{x})$ is the kernel vector of $\mathbf{x}$ without any label information, $\mathbf{y}$ is the label vector and $\circ$ is the element-wise multiplication.

We now adapt multiple equations of the original PCVM to integrate the Nyström approximated matrix. Beginning with the elements of vector (for a *single* training vector i) $\mathbf{z}_\theta$:

$$z_{i,\theta} = \Xi_\theta(\mathbf{x}_i)(\mathbf{y} \circ \mathbf{w}) + b, \tag{6}$$

we rewrite Eq.(6) in matrix notation for all training points:

$$\hat{\mathbf{z}} = (((\hat{\mathbf{y}} \circ \hat{\mathbf{w}})^\top \hat{K_1}) \cdot K_2)^\top + b \tag{7}$$

and further obtain column vectors $\bar{\mathbf{H}}_\theta$ and the reduced form $\bar{\bar{\mathbf{H}}}_\theta$, by using only the non-vanishing basis functions and the Nyström approximated matrices in Eq. (1). In the maximization step of the original PCVM the $\mathbf{w}$ are updated as (see Eq. (2)):

$$\mathbf{w}^{\text{new}} = M\underbrace{(M\Phi_\theta(\mathbf{x})^\top \Phi_\theta(\mathbf{x})M + I_N)}_{\Upsilon}{}^{-1} M(\Phi_\theta(\mathbf{x})^\top \bar{\mathbf{H}}_\theta - b\Phi_\theta(\mathbf{x})^\top \mathbf{I}) \tag{8}$$

To account for the now excluded labels we reformulate Equation (2) as:

$$\mathbf{w}^{\text{new}} = M\underbrace{(M(\Xi_\theta(\mathbf{x})^\top \Xi_\theta(\mathbf{x})\hat{\mathbf{y}}^\top \hat{\mathbf{y}})M + I_N)}_{\Upsilon}{}^{-1} M(\hat{\mathbf{y}}^\top(\Xi_\theta(\mathbf{x})^\top \bar{\mathbf{H}}_\theta) - b\hat{\mathbf{y}}^\top(\Xi_\theta(\mathbf{x})^\top \mathbf{I}))$$

The update equations of the weight vector include the calculation of a matrix inverse of Υ which was originally calculated using the Cholesky decomposition. To keep our objective of small matrices we will instead calculate the pseudo-inverse of this matrix using a Nyström approximation of Υ. It should be noted at this point that the matrix Υ is psd by construction. We approximate Υ by selecting another set of m^* landmarks from the indices of the not yet pruned weights and calculate the matrix $\tilde{\Upsilon} = C_{Nm^*} W^{-1}_{m^*,m^*} C^\top_{Nm^*}$ in analogy to Eq (5) with submatrices: [1]

$$\begin{aligned} C_{Nm^*} &= E_{N[m]} + ((\hat{K_1} \cdot (K_2 \cdot (K_1 \cdot \hat{K}_{2\cdot,[m^*]}))(\hat{\mathbf{y}}^\top \hat{\mathbf{y}}_{[m^*]})) \\ &\quad \circ\sqrt{2}\hat{\mathbf{w}}) \circ \sqrt{2}\hat{\mathbf{w}}^\top_{[m^*]} \\ W_{m^*,m^*} &= C^{-1}_{m^*,\cdot} \end{aligned}$$

Where $\circ$ indicates (in analogy to its previous meaning) that each row of the left matrix is elementwise multiplied by the right vector and $E_{N[m]}$ is the matrix consisting of the m landmark columns of the $N \times N$ identity matrix. The terms $\sqrt{2}\hat{\mathbf{w}}$ and $\sqrt{2}\hat{\mathbf{w}}^\top_{[m^*]}$ are the entries of the diagonal matrix M as defined in Eq. (4) but now given in vector form.

These two matrices serve as the input of a Nyström approximation based pseudo-inverse (as discussed in sub section 2.3) and we obtain matrices $V \in \mathbb{R}^{N\times r}, U \in \mathbb{R}^{r\times N}$ and $S \in \mathbb{R}^{r\times r}$, where $r \leq m^*$ is the rank of the pseudo inverse. Further we define two

[1] The number of landmarks m^* is fixed to be 1% of $|w|$ but not more then 500 landmarks. If the length of $\mathbf{w}$ drops below 100 points we use the original PCVM formulations.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

vectors $\mathbf{v}_1 = \bar{\hat{\mathbf{H}}}_\theta^{\top} \cdot K_1$ and $\mathbf{v}_2 = \mathbf{I}^\top \cdot K_1$. We obtain the approximated weight update $\mathbf{w}^{\text{new}} = V \cdot (S \cdot U^\top \cdot (\sqrt{2}\hat{\mathbf{w}}(\hat{\mathbf{y}}(\mathbf{v}_1 \cdot \hat{K_2})^\top - b \cdot \hat{\mathbf{y}}(\mathbf{v}_2 \cdot \hat{K_2})^\top)))\sqrt{2}\hat{\mathbf{w}}$. The update of the bias is originally done as

$$\mathbf{b} = t(1 + tNt)^{-1}t(\mathbf{I}^\top \bar{\mathbf{H}}_\theta - \mathbf{I}^\top \Phi_\theta(\hat{\mathbf{y}}\hat{\mathbf{w}})) \tag{9}$$

which is replaced to: $\mathbf{b} = t(1 + tNt)^{-1}t(\mathbf{I}^\top \bar{\hat{\mathbf{H}}}_\theta - \mathbf{I}^\top((((\hat{\mathbf{y}}\hat{\mathbf{w}})^\top \hat{K_1}) \cdot K_2)^\top))$ Subsequently the entries in $\hat{\mathbf{w}}$ which are close to zero are pruned out and the matrices $\hat{K_1}$ and $\hat{K_2}$ are modified accordingly.

2.3 Pseudo Inverse, SVD and EVD of a Nyström approximated matrix

The pseudo inverse of a Nyström approximated matrix can be calculated by a modified singular value decomposition (SVD) with a rank limited by $r^* = \min\{r, m\}$ where r is the rank of the pseudo inverse and m the number of landmark points. The output is given by the rank reduced left and right singular vectors and the reciprocal of the singular values. The singular value decomposition based on a nyström approximated similarity matrix $\tilde{K} = C_{Nm}W_{m,m}^{-1}C_{Nm}^\top$ with m landmarks, calculates the left 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 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 Nyström approximation of the matrix $\tilde{K}\tilde{K}^\top$ using e.g. the same landmark points as for the input matrix $\tilde{K}$. Subsequently an eigenvalue decomposition (EVD) is calculated on the approximated matrix $\zeta = \tilde{K}\tilde{K}^\top$.For a matrix approximated by Eq. (5) it is possible to compute its exact eigenvalue decomposition in linear time. To compute the eigenvectors and eigenvalues of an *indefinite* matrix we first compute its squared form. Let K be a psd similarity matrix, for which we can write its decomposition as $\tilde{K} = K_{N,m}K_{m,m}^{-1}K_{m,N} = K_{N,m}U\Lambda^{-1}U^\top K_{N,m}^\top = BB^\top$, 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* $\tilde{K}$: $\tilde{K}^2 = BB^\top BB^\top = BVAV^\top B^\top$, 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 with B from left we get: $\underbrace{BB^\top}_{\tilde{K}}\underbrace{(Bv)}_{u} = a\underbrace{(Bv)}_{u}$. 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: $\tilde{K} = BVV^\top B^\top = BVA^{-1/2}AA^{-1/2}V^\top B^\top = CAC^\top$, where we defined $C = BVA^{-1/2}$ as the matrix of orthonormal eigenvectors of K. The eigenvalues of $\hat{K}$ can be obtained using $A = C^\top \hat{K} C$.

3 Complexity analysis

The original PCVM update rules have costs of $\mathcal{O}(M^3)$ and memory storage $\mathcal{O}(M^2)$, where M is the number of non-zero basis functions and $M \leq N$ calculating the kernel matrix costs $\mathcal{O}(N^2)$. Accordingly the runtime complexity PCVM is $\mathcal{O}(N^2 + M^3)$ and the memory complexity is $\mathcal{O}(N^2)$, which is reduced in the final model, due to

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

the sparseness constraint. The Ny-PCVM involves the extra Nyström approximation of the kernel matrix to obtain $K_{N,m}$ and $K_{m,m}^{-1}$. If we have m landmarks, $m \ll N$, this gives costs of $\mathcal{O}(mN)$ for the first matrix and $\mathcal{O}(m^3)$ for the second, due to the matrix inversion. Further both matrices are multiplied within the optimization so we get $\mathcal{O}(m^2N)$. Similarly, the matrix inversion of the original PCVM with $\mathcal{O}(M^3)$ is reduced to $\mathcal{O}(m^{*2}M) + \mathcal{O}(m^{*3})$ due to the Nyström approximation of the matrix Υ. Since we choose $m^* = \min\{\max\{1, N \cdot 0.01\}, 500\}$, the complexity of the inversion of Υ is small. If we assume $m^* < m \ll N$ we get $\mathcal{O}(m^2N)$ as the overall runtime complexity. The memory complexity of Ny-PCVM is $\mathcal{O}(mN)$.

4 Experiments

We compare Ny-PCVM to PCVM on various $\mathcal{N}(0, 1)$ normalized larger datasets[2], The *spam* data (4601pts, two classes,57dims), *satellite* (6435pts, six classes,36dims), *usps*(11000pts, 10 classes, 256dims)[3] and the *adult* data (30162pts, two classes,14dims) we use the ELM kernel [7] and for MNIST (70.000pts, ten classes,784dims)[4] a polynomial kernel (Details see [5]) all with 500 landmarks each. *Adult*, *Spam* and *Satellite* are taken from the UCI database.We report mean, standard errors and runtimes as obtained by a 10 fold crossvalidation. Considering the results in Table 1 and Table 2 we observe that Ny-PCVM achieves similar accuracies compared to the other approaches while being substantially faster then PCVM and competitive to CVM. For *adult* and *mnist* the runs took too long with PCVM. On can clearly see that Ny-PCVM scales linear in the number of samples in contrast to the cubic complexity of PCVM Ny-PCVM is a magnitude slower than CVM but can also be used for non-psd datasets. Non-vectorial data

	Ny-PCVM	PCVM	CVM
spam	92.63 ± 1.0	92.63 ± 1.0	93.50 ± 1.0
satellite	83.53 ± 1.3	71.39 ± 2.3	89.26 ± 1.2
usps	90.43 ± 0.6	87.53 ± 1.1	96.21 ± 0.7
adult	79.25 ± 0.7	$--$	80.90 ± 0.5
mnist	83.24 ± 0.7	$--$	89.03 ± 0.7

Table 1: Accuracies - vectorial data

	Ny-PCVM	PCVM	CVM
spam	5.69 ± 0.9	62.38 ± 2.0	0.28 ± 0.2
satellite	6.10 ± 1.0	64.37 ± 1.9	0.42 ± 0.2
usps	13.42 ± 1.5	153.40 ± 15.6	1.14 ± 0.1
adult	19.62 ± 1.1	$--$	0.95 ± 0.1
mnist	53.23 ± 1.5	$--$	2.4 ± 0.1

Table 2: Runtimes - vectorial data

given by means of indefinite kernels have not yet been considered for the PCVM but are of wide interest [9]. In contrast to many standard kernel approaches, for PCVM, the indefinite kernel matrices need not to be corrected by costly eigenvalue correction [4]. Further the PCVM provides direct access to probabilistic classification decisions. We compare to the indefinite kernel fisher discriminant (iKFD) [10] [5].

The data sets are *gesture* (1500pts, 20 classes), *Zongker* (2000pts, 10 classes) and *Proteom* (2604pts, 53 classes) all from [6]; *Chromo* (4200pt, 21 classes) from [8] and *Swiss* (82525 pts, 46 classes) from [1], database 10/2010, reduced to prosite labeled classes with at least 1000 entries (1000 randomly chosen landmarks). All data are processed as indefinite kernels with 100 landmarks if not stated otherwise. For all experiments we report mean and standard errors as obtained by a 10 fold crossvalidation. The probabilistic outputs can be directly used to allow for a reject region but can also be used to provide alternative classification decisions e.g. in a ranking framework In Table

[2]Comparison to standard benchmarks skipped due to lack of space - results e.g. to SVM are competitive.
[3]Taken from `http://www.cs.nyu.edu/~roweis/data.html`
[4]`http://yann.lecun.com/exdb/mnist/`
[5]SVM for indefinite kernels use a proxy approach not scaling to larger data and are not probabilistic.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

	Ny-PCVM	PCVM	iKFD
gesture	93.00 ± 1.8	95.80 ± 1.4	98.07 ± 0.7
zongker	91.35 ± 1.9	91.65 ± 2.3	96.95 ± 0.1
proteom	88.06 ± 2.3	88.56 ± 2.4	99.35 ± 0.8
chromo	93.10 ± 1.1	95.07 ± 1.0	97.29 ± 0.7
swiss	67.70 ± 4.4	–	–

Table 3: Accuracies - indefinite kernels

	Ny-PCVM	PCVM	iKFD
gesture	1.62 ± 0.2	16.72 ± 14.0	69.38 ± 7.5
zongker	2.00 ± 0.4	17.13 ± 13.6	74.22 ± 6.9
proteom	1.98 ± 0.4	5.33 ± 0.7	758.90 ± 28.4
chromo	3.00 ± 0.3	10.36 ± 6.4	1073.9 ± 26.2
swiss	29.37 ± 6.9	–	–

Table 4: Runtimes - indefinite kernels

3 and Table 4 we show the results for different non-metric proximity datasets using Ny-PCVM, PCVM and iKFD. We observe that the prediction accuracy of iKFD is better compared to Ny-PCVM 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-PCVM 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-PCVM is faster than PCVM by a magnitude and by 1-3 magnitudes compared to iKFD in the training. For non-psd data Ny-PCVM is substantially better in runtime and sparsity compared to the state of the art with good prediction accuracy.

5 Conclusions

We presented an alternative formulation of the PCVM employing the Nyström approximation. We found that Ny-PCVM is competitive in the prediction accuracy with PCVM and alternative approaches, while taking substantially less memory and runtime. In this work we also have shown how the Nyström approximation can be used to calculate an eigenvalue decomposition, a singular value decomposition and the pseudo-inverse of a Nyström approximation in an efficient way. The Ny-PCVM provides now an effective way to obtain a *probabilistic* classification model for medium to large psd *and* non-psd datasets, in batch mode with *linear* runtime and memory complexity. To our best knowledge it is the only approach which scales to larger non-psd data. [6]

References

[1] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. 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.

[2] H. Chen, P. Tino, and X. Yao. Probabilistic classification vector machines. *IEEE TNN*, 20(6):901–914, 2009.

[3] H. Chen, P. Tino, and X. Yao. Efficient probabilistic classification vector machine with incremental basis function selection. *IEEE TNN-LS*, 25(2):356–369, 2014.

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

[5] R. Chitta, R. Jin, and A. K. Jain. Efficient kernel clustering using random fourier features. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 161–170, 2012.

[6] R. P. Duin. PRTools, march 2012.

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

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

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

[10] E. Pekalska and B. Haasdonk. Kernel discriminant analysis for positive definite and indefinite kernels. *IEEE TPAMI*, 31(6):1017–1032, 2009.

[11] C. K. I. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *NIPS 2000*, pages 682–688, 2000.

[6] **Acknowledgment:** A Marie Curie Intra-European Fellowship (IEF): FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) and support from the Cluster of Excellence 277 Cognitive Interaction Technology funded by the German Excellence Initiative 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 R. Duin, Delft University for various support with distools and prtools.