Discriminative Fast Soft Competitive Learning

Frank-Michael Schleif

University of Birmingham, School of Computer Science
Edgbaston, B15 2TT Birmingham, UK

Abstract. Proximity matrices like kernels or dissimilarity matrices provide non-standard data representations common in the life science domain. Here we extend fast soft competitive learning to a discriminative and vector labeled learning algorithm for proximity data. It provides a more stable and consistent integration of label information in the cost function solely based on a give proximity matrix without the need of an explicite vector space. The algorithm has linear computational and memory requirements and performs favorable to traditional techniques.

1 Introduction

The amount of digital data doubles roughly every 20 months often given by non-vectorial data formats such as XML, graph structures, sequence data or others. Such data is getting more and more frequent, leading to large proximity data sets. Classical cluster methods, like k-means process Euclidean data only. Also kernel approaches like kernel-k-means or kernel soft competitive learning (KSCL) [1] are often limited, due to the lack of a valid kernel. More recently indefinite kernel techniques were proposed [2] but often with high complexity or by optimizing an aligned kernel matrix.

A further efficient alternative is given by dissimilarity learners like the relational soft competitive learning [3] (R-SCL), a clustering algorithm for arbitrary dissimilarity data. R-SCL is an extension of soft-competitive learning (SCL) [4]. It replaces the Euclidean distance function of a data point $\mathbf{v}$ to a cluster representant or prototype $\mathbf{w}$ by an implicit representation which refers to the dissimilarity matrix $D \in \mathbb{R}^{N \times N}$ only with N as the number of samples and $d_{ij} = |\mathbf{v}_i - \mathbf{v}_j|^2$ denotes the underlying dissimilarities induced by an arbitrary symmetric bilinear form. While standard R-SCL has squared complexity a linear cost algorithm can be obtained by using the Nyström approximation [5].

In [6] the author has shown that arbitrary proximity matrices can be used by the R-SCL algorithm by integrating simple transformation rules in the original formulation. The obtained fast soft competitive learning algorithm (FSCL) is an effective approach to analyze large proximity datasets for *unsupervised* problems. Often the given data may also contain partial or full label information and especially in the life science domain those labels may be probabilistic in contrast to crisp labels. Considering such data sets, standard supervised or even semi-supervised learning algorithm are not applicable in general. To apply a standard support vector classifier (SVM) we would need to defuzzify the labels making them crisp which in general can degenerate accuracy.

Here we extend batch FSCL to take this label information into account, leading to a new formulation of FSCL called supervised FSCL (S-FSCL) which is a discriminative

clustering approach. In contrast to the former approaches also partial labeling is permitted and even more important the labels need not to be crisp such that probabilistic or fuzzy labeled data can be analyzed.

In section 2 we briefly review related work on proximity learning focusing on discriminative clustering approaches. We present the supervised fast soft competitive learning in section 3 and summarize the results of our empirical studies in section 4.

2 Related work

Clustering analysis has found a wide range of application [7] and with the advent of large proximity data sets also proximity clustering has been studied by different authors, e.g. in the line of large scale kernel clustering [8] and relational or dissimilarity clustering [3]. The availability of label information may help to improve current cluster approaches by guiding the optimization process. This is also interesting if data are only partially labeled or fuzzy labeled and fully supervised approaches are inaccessible or semi-supervised techniques are limited e.g. to two class scenarios [9].

In the last years different clustering approaches were proposed using partial label information, some of them also supported fuzzy-labeled data. In [10] an online SCL clustering approach was coupled with an additive label error term in the cost function to allow for fuzzy-labeled data [11], but this approach is sensitive to the balancing parameter in the cost function. In [12] this *online* approach was changed to a product based label-error leading to a more stable behavior. Both former approaches were found to be efficient but do not scale well to larger problems and consider vectorial datasets only. A batch SCL clustering method using the additive label error was proposed in [13] motivating also a relevance learning strategy for vectorial data, but also this approach is sensitive to the parameters. More recently also an *online* supervised Learning Vector Quantizer for multivariate class labels[1] was proposed in [14] which was found to be very efficient and which we will use as our baseline method. Other recent methods to incorporate label information for kernel clustering were proposed e.g. in [15] and for vectorial data in [16]. All these approaches focus either on vectorial data or do not scale to larger problems. Although some of the methods are online, and theoretically of linear complexity, the repetitive calculation of distances for high dimensional data and the used gradient descent learning makes them slow in practice whereas batch methods are known for quick convergence. Our proposal is a *batch* approach for fuzzy labeled data which is efficient for larger scale proximity matrices. Hence it keeps a lot of flexibility regarding the data encoding, e.g. by a dedicated kernel or distance function.

To address large scale problem in proximity clustering a multitude of contributions have been made in the last years e.g. by means of core set clustering [17], the Nyström approximation [8, 6] or patch learning approaches [18]. We will use a Nyström strategy as given in [6]. Subsequently, we will briefly introduce soft competitive learning as the basic method for multiple related approaches [13, 10–12, 6] mentioned above followed by the derivation of a (semi-)supervised extension of FSCL.

[1] Not to mix-up with multi labels, where an object can fully belong to multiple classes

2.1 Soft Competitive Learning

In contrast to regular k-means, soft competitive learning (SCL) [4] extends the quantization error to incorporate data induced neighborhood cooperation: $E_{\mathrm{SCL}} := \sum_{ij} h_\sigma(r_{ij}) d(\mathbf{v}_i, \mathbf{w}_j)$ where $h_\sigma(t) = \exp(-t/\sigma)$ exponentially scales the neighborhood range, and r_{ij} denotes the rank of prototype $\mathbf{w}_j$ with respect to $\mathbf{v}_i$, i.e. the number of prototypes $\mathbf{w}_k$ with $k \neq j$ which are closer to $\mathbf{v}_i$ as measured by the Euclidean distance d. SCL optimizes the prior cost function E_{SCL} by means of a stochastic gradient descent, annealing the neighborhood range σ during training such that, in the limit, the standard quantization error is approximated [4]. The iterative adaptation rule is $\mathbf{w}_j := \mathbf{w_j} + \eta \cdot h_\sigma(r_{ij})(\mathbf{v}_i - \mathbf{w}_j)$ where η denotes the learning rate. There exists a faster (euclidean) batch optimization scheme as introduced in [19] which optimizes prototype locations and assignments as:

$$\mathbf{w_j} := \sum_i h_\sigma(r_{ij}) \mathbf{v}_i / \sum_i h_\sigma(r_{ij}) \tag{1}$$

with r_{ij} based on $d(\mathbf{v}_i, \mathbf{w}_j)$. The online kernelized SCL was proposed in [1], replacing the original distance calculation by a kernel expansion and a batch version was implicitly proposed in the FSCL [6].

2.2 Relational Soft Competitive Learning

Relational soft competitive learning (R-SCL) as introduced in [3] assumes that a symmetric dissimilarity matrix D with entries d_{ij} describing pairwise dissimilarities of data is available. In principle, it is very similar to KSCL. There are two differences: R-SCL is based on dissimilarities rather than similarities, and it solves the resulting cost function using a *batch* optimization with quadratic convergence as compared to a stochastic gradient descent.

As shown in [20], there always exists a so-called pseudo-Euclidean embedding of a given set of points characterized by pairwise symmetric dissimilarities by means of a mapping Φ, i.e. a real vector space and a symmetric bilinear form (with probably negative eigenvalues) such that the dissimilarities are obtained by means of this bilinear form. As before, prototypes are restricted to convex combinations $\mathbf{w}_j = \sum_l \alpha_{jl} \Phi(\mathbf{v}_l)$ with $\sum_l \alpha_{jl} = 1$ Dissimilarities are computed as:

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

where $[\cdot]_i$ refers to component i of the vector. This allows a direct transfer of batch SCL to general dissimilarities by the following iterations derived from (1)

$$\alpha_{jl} := h_\sigma(r_{jl}) / \sum_l h_\sigma(r_{jl}) \tag{3}$$

with r_{jl} based on $d(\Phi(\mathbf{v}_j), \mathbf{w}_l)$. This algorithm is soft competitive learning in pseudo Euclidean space for every symmetric dissimilarity matrix D. If negative eigenvalues are present convergence is not always guaranteed, but observed in general[3].

3 Supervised fast soft competitive learning

Let Y be the label matrix of the training points with entries $\mathbf{y}_i \in \mathbb{R}^C$ and C the number of classes. For each $\mathbf{y}_i$ we expect entries $y_{ic} \in [0,1]$ and $\sum_{c=1}^{C} y_{ic} = 1$. Further we introduce a vector label $\mathbf{l}_j$ for each prototype $\mathbf{w}_j$ following the same constraints.

To integrate supervised information in the FSCL approach we extend the original distance calculation[2] by a multiplicative error term similar as suggested for online, vectorial SCL in [12].

Former approaches using an additive or multiplicative label error were found to be sensitive to the used weighting or offset parameter to rescale the different error contributions. This problem is addressed subsequently by application of the softmax function on the individual distance errors and the label error of each data point such that both error functions provide values in a range $[0,1]$. In this way we avoid additional control parameters and obtain a stable learning behavior [3].

Accordingly the original distance function Eq. (2) is adapted to:

$$d((\Phi(\mathbf{v}_i), \mathbf{y}_i), (\mathbf{w}_j, \mathbf{l}_j)) = 1 - \underbrace{(f(d(\Phi(\mathbf{v}_i), \mathbf{w}_j)) \cdot f(d^*(\mathbf{y}_i, \mathbf{l}_j)))}_{(s)} \tag{4}$$

with f being a softmax function and d^* the squared Euclidean distance. If only partial label information is given we can just ignore the label part in Eq 4. The *initial* label $\mathbf{l}_j$ of the prototypes are determined by post labeling and are updated as:

$$\mathbf{l}_j^* = \alpha_{j\cdot} \cdot Y \tag{5}$$

being the mean label of all data points weighted by the contribution of each point to this prototype. It should be noted that the distance in Eq. (4) is always non-negative and symmetric but may be non-metric as easily shown by counter examples, this however is not a severe problem for the underlying R-SCL as discussed in [3]. For test data points the distance calculation remains unchanged following the winner takes all scheme. The formulation given in Eq. (4) can be interpreted as the probability that a data point was generated by a Gaussian distribution centered on the prototype under the condition of similar labeling of the data point and the prototype. The obtained similarity (s) is subsequently mapped back into a dissimilarity by $1-(s)$ to keep the distance interpretation of the remaining optimization function.

To address the problem of large input proximity matrices we use the Nyström approximation [21, 22]. The practical idea is to select m landmark indices and the corresponding rows and columns from the matrix S to obtain the landmark matrix $S_{m,m}$. The original gram matrix S can then be approximated as $\tilde{S} = S_{N,m} S_{m,m}^{-1} S_{m,N}$ which is of complexity $\mathcal{O}(m^3 N)$ instead of $\mathcal{O}(N^2)$, i.e. it is linear if the approximation quality m is fixed. In [22] it was shown that the same strategy can also be applied to symmetric dissimilarity matrices. For R-SCL, this yields the approximation of the distance

[2] Either based on a kernel expansion or on a dissimilarity expansion, as shown e.g in Eq. (2)

[3] The softmax parameter σ is fixed to $\sigma = 1$ and is insensitive with respect to the data, assuming that the data representation is reasonable expressive. It can be subsumed by the given distance if we assume σ to be equal for each prototype.

computation (2)

$$d(\mathbf{v_i}, \mathbf{w_j})^2 \approx [D_{N,m}(D_{m,m}^{-1}(D_{m,N}\alpha_j))]_i - \frac{1}{2} \cdot (\alpha_j^t D_{N,m}) \cdot (D_{m,m}^{-1}(D_{m,N}\alpha_j))$$

which is $\mathcal{O}(m^3 N)$. Again, the approximation is exact if the number of samples m is chosen according to the rank of D. For similarity data we transform the similarity matrix S to a dissimilarity matrix D using Equations from [20].

$$d(\mathbf{v_i}, \mathbf{v_j})^2 = s(\mathbf{v_i}, \mathbf{v_i}) + s(\mathbf{v_j}, \mathbf{v_j}) - 2s(\mathbf{v_i}, \mathbf{v_j}) \tag{6}$$

which can be coupled with the Nyström approximation, avoiding the full calculation of the matrix S [22]. S-FSCL is a wrapper around a modified R-SCL using the distance function Eq. (4) followed by a subsequent update of the prototype labels using Eq. (5) details on the implementation of the original R-SCL are given in [3]. The runtime-complexity of S-FSCL is dominated by the distance calculations with $\mathcal{O}(m^3 N)$. R-SCL and hence S-FSCL shows fast convergence (see [3]) due to the batch approach. The memory complexity is dominated by the $m \times N$ dis-/similarity matrix and is $\mathcal{O}(mN)$.

4 Experiments

We compare the efficiency of supervised fast SCL (S-FSCL) with its unsupervised fast SCL (FSCL) and the online Robust Soft-LVQ for multivariate (MRSLVQ) labels as proposed in [14], for very large data sets we use the core vector machine (CVM) [23]. Initially we show the usefulness of the introduced supervision concept and the effectiveness of the batch approach by use of the classical checkerboard data set. The data form a 5×5 checkerboard with each cluster consisting of 200 Gaussian distributed points, so we have $N = 5000$ points. To represent the data we use an rbf kernel which is approximated in the FSCL method by a Nyström approximation with $N/10$ landmarks. We use 2 prototypes which is theoretically sufficient to represent the checkerboard data in a supervised learning task. We apply the supervised FSCL and the unsupervised FSCL and obtain prototype assignments as shown in Figure 1. For both methods the prototypes are finally located in the center of the data. The supervised FSCL achieves an accuracy of

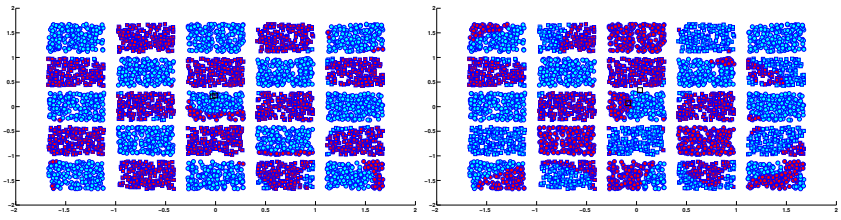

Fig. 1: Checkerboard data (left) supervised FSCL, (right) unsupervised FSCL. The predicted label is given by color (gray shade) and the true label by the shape of the objects.

95.12%, whereas the unsupervised FSCL got 52.96%. For this test data the supervised

	FSCL	S-FSCL	MRSLVQ
Checkerboard (rbf)	44.14 ± 3.51	0.52 ± 3.77	$\mathbf{0.40 \pm 0.05}$
Plant-Tissue (rbf)	42.76 ± 1.19	$\mathbf{32.23 \pm 1.00}$	37.62 ± 2.13
Remote-Sensing (euclidean)	40.86 ± 0.94	$\mathbf{39.95 \pm 1.09}$	44.27 ± 1.61

Table 1: Results of the fuzzy labeled data with mean and standard deviation of the test error.

information is clearly needed to achieve a good classification result since the data distribution is not sufficient to estimate the prototype labels. To get a fair comparison the *only* difference between the supervised and unsupervised FSCL is the distance function as discussed before with respect to Eq. (4).

In the further experiments we used fuzzy labeled data sets, all represented by means of dissimilarity matrices using either the Euclidean distance or an underlying rbf kernel so all data are metric but as already shown in [6] more generic data formats can be used. All data matrices have been approximated with a constant number of 100 randomly chosen landmarks. (1) *Plant tissue data*: The data are 4418 points of 22 dimensional image features in 11 classes of a serial transverse section of barley grains taken from [24]. The different tissue regions are hard to discriminate such that for a substantial part of this datasets items are labeled by fuzzy labels. (2) *Remote sensing data*: is a multi-spectral LANDSAT TM satellite image of the Colorado area taken from [25] with 6 different spectral bands. There are 14 labels describing different vegetation types and geological formations. The size of the original image is 1907×1784 pixels[4]. Fuzzy labels were obtained by a downsampling of this data set to 12650 points where the original image was cut to 1840×1760 pixel. The fuzzy labels were derived from the histograms of the averaged pixel areas.

For comparison to other alternative methods we also analyze different medium scale standard datasets using either a linear kernel or a defacto parameter-free extreme learning machine (elm) kernel [26] compared with a core vector machine classifier [23]. The MNIST data [5] contains $70000, 719$-dimensional binary images from 10 digit classes. We used a neural kernel $k(\mathbf{v_i}, \mathbf{v_j}) = \tanh(av_i^\top v_j + b)$ with $a = 0.0045$ and $b = 0.11$ acc. to [8]. The USPS [6] contains $11000, 256$-dimensional character feature vectors from 10 classes analyzed by a parameter free elm kernel. The SPAM database [7], contains $4601, 57$-dimensional feature vectors, processed by a linear kernel. All matrices have been Nyström approximated and converted to a distance matrix at linear costs[22].

The model complexity for (supervised) FSCL and MRSLVQ has to be determined in advance, although one prototype per class if often sufficient it is beneficial to spend some extra prototypes to address potential sub-populations. Unused prototypes are removed either during learning (FSCL) or in the final model (MRSLVQ). For the Plant-Tissue data we use 2 prototypes per class, for the Checkerboard data 1 per class and for the remaining data sets we used 10 prototypes per class. All data are analyzed in a 5-fold crossvalidation. The results are shown in Table 1 and Table 2. For Table 1 we observe that the additional supervised information is in general helpful to improve the

[4] Thereby 9 pixel have an unknown label and have been removed.

[5] http://yann.lecun.com/exdb/mnist/

[6] http://www.cs.nyu.edu/ roweis/data.html

[7] http://archive.ics.uci.edu/ml/datasets

	FSCL	S-FSCL	CVM
MNIS T	$\mathbf{20.83 \pm 1.05}$	22.16 ± 7.61	40.04 ± 3.54
USPS	15.62 ± 1.01	$\mathbf{14.33 \pm 1.10}$	18.77 ± 1.01
SPAM	17.26 ± 1.79	$\mathbf{12.06 \pm 1.20}$	27.67 ± 1.08

Table 2: Test set error (mean/std) of medium to large scale standard data set.

model with respect to the classification task[8]. The proposed approach is in parts better or competitive to the MRSLVQ but substantially faster under practical settings due to the batch strategy, avoiding repetitive calculations of distance or gradients as needed in MRSLVQ. The general runtime for S-FSCL for a single model is in the range of seconds whereas the MRSLVQ is most often slower by two magnitudes. In Table 2 we observe again that the supervision is in general helpful although often the effect is not so substantial. The results in Table 2 are again quite good compared to a CVM result. Due to the pre-calculation of the approximated kernel the actual model calculation can be done within seconds to minutes. Note that e.g. for the SPAM database fuzzy labels are not given but likely to observe in practical applications because multiple users will consider almost identical emails as spam or non-spam. Hence, the S-FSCL model would be more appropriate in these cases than the crisp CVM approach.

5 Conclusions

Here we proposed a *supervised* version of the batch FSCL algorithm. The algorithm permits the usage of fuzzy labeled input data by means of a dissimilarity matrix representation. The given dissimilarity data can be of large scale due to the underlying Nyström approximation such that data with multiple 1000 points can be handled easily. The obtained supervised clustering approach provides probabilistic class assignments and was found to be quite robust and achieved better or competitive results to alternative approaches. Using the suggested transformation and Eq. (6) also kernel representation are available. In this way S-FSCL can be used for a wide range of problem settings. Considering the very limited and restricted amount of classifiers for fuzzy labeled data the FSCL is an effective solver for medium to large scale problems in this line. In future work we will focus on further improvements for very large scale problems using e.g. random approximation strategies as suggested in [27] and analyze the efficiency for practical problems with unsafe label information in the life sciences. **Acknowledgment:** Marie Curie Intra-European Fellowship (IEF): FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS) is greatly acknowledged.

References

[1] A. Kai Qin and Ponnuthurai N. Suganthan. Kernel neural gas algorithms with application to cluster analysis. In *ICPR (4)*, pages 617–620, 2004.

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

[8] The effect is obviously less severe if the data distribution follows the labeling very closely as e.g. for the remote sensing data, where also an unsupervised clustering gives similar results

[3] B. Hammer and A. Hasenfuss. Topographic mapping of large dissimilarity datasets. *Neural Computation*, 22(9):2229–2284, 2010.

[4] T. Martinetz, S. Berkovich, and K. Schulten. Neural Gas Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE TNN*, 4(4):558–569, 1993.

[5] 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.

[6] F.-M. Schleif, X. Zhu, A. Gisbrecht, and B. Hammer. Fast approximated relational and kernel clustering. In *Proc. of ICPR 2012*, pages 1229 – 1232. IEEE, 2012.

[7] A. K. Jain. Data clustering: 50 years beyond K-means. *Pat. Rec. Let.*, 31:651–666, 2010.

[8] Radha Chitta et al. Approximate kernel k-means: solution to large scale kernel clustering. In Chid Apté, editor, *KDD*, pages 895–903. ACM, 2011.

[9] Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artif. Intell. and Machine Learning*, 3(1):1–130, 2009.

[10] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, and W. Herrmann. Fuzzy classification by fuzzy labeled neural gas. *Neural Netw.*, 19(6-7):772–779, 2006.

[11] T. Villmann, F.-M. Schleif, B. Hammer, and M. Kostrzewa. Exploration of mass-spectrometric data in clinical proteomics using learning vector quantization methods. *Briefings in Bioinformatics*, 9(2):129–143, 2008.

[12] M. Kästner and T. Villmann. Fuzzy supervised self-organizing map for semi-supervised vector quantization. In Leszek Rutkowski, editor, *ICAISC (1)*, LNCS, pages 256–265, 2012.

[13] B. Hammer, A. Hasenfuss, F.-M. Schleif, and T. Villmann. Supervised batch neural gas. In *Proc. of ANNPR 2006*, pages 33–45, 2006.

[14] P. Schneider, T. Geweniger, F.-M. Schleif, M. Biehl, and T. Villmann. Multivariate class labeling in robust soft LVQ. In *Proc. of ESANN 2011*, pages 17–22, 2011.

[15] T. Finley and T. Joachims. Supervised clustering with support vector machines. In Luc De Raedt, editor, *ICML*, volume 119, pages 217–224. ACM, 2005.

[16] O. Arandjelovic. Discriminative k-means clustering. In *IJCNN*, pages 1–7, 2013.

[17] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *STOC*, pages 250–257, 2002.

[18] Nikolai Alex, Alexander Hasenfuss, and Barbara Hammer. Patch clustering for massive data sets. *Neurocomputing*, 72(7-9):1455–1469, 2009.

[19] M. Cottrell, B. Hammer, A. Hasenfuss, and T. Villmann. Batch and median neural gas. *Neural Networks*, 19:762–771, 2006.

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

[21] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In Todd K. Leen, editor, *NIPS*, pages 682–688. MIT Press, 2000.

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

[23] I. Tsang, A. Kocsor, and J. Kwok. Simpler core vector machines with enclosing balls. In *Proc. of the 24th Int. Conf. on Machine Learning (ICML 2007), 2007*, pages 911–918, 2007.

[24] C. Brüß, F. Bollenbeck, and F.-M. Schleif et al. Fuzzy image segmentation with fuzzy labelled neural gas. In *Proc. of ESANN 2006*, pages 563–569, 2006.

[25] F.-M. Schleif, F.-M. Ongyerth, and T. Villmann. Supervised data analysis and reliability estimation for spectral data. *Neuro Comp.*, 72(16-18):3590–3601, 2009.

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

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