Kernel Robust Soft Learning Vector Quantization

Daniela Hofmann and Barbara Hammer

CITEC Center of Excellence, Bielefeld University, Germany
{dhofmann,bhammer}@techfak.uni-bielefeld.de

Abstract. Prototype-based classification schemes offer very intuitive and flexible classifiers with the benefit of easy interpretability of the results and scalability of the model complexity. Recent prototype-based models such as robust soft learning vector quantization (RSLVQ) have the benefit of a solid mathematical foundation of the learning rule and decision boundaries in terms of probabilistic models and corresponding likelihood optimization. In its original form, they can be used for standard Euclidean vectors only. In this contribution, we extend RSLVQ towards a kernelized version which can be used for any positive semidefinite data matrix. We demonstrate the superior performance of the technique, kernel RSLVQ, in a variety of benchmarks where results competitive or even superior to state-of-the-art support vector machines are obtained.

1 Introduction

A variety of powerful classification, regression, and inference techniques being available, machine learning has revolutionized the possibility to deal with large electronic data sets and to infer models for complex settings where standard statistical models are no longer sufficient. Because of its high flexibility and its usually excellent classification and generalization performance, the support vector machine (SVM) constitutes one of the current flagships of supervised machine learning. With machine learning techniques becoming more and more popular in diverse application domains, there is an increasing need for models which can easily be used by applicants outside the field of machine learning or computer science. Moreover, due to more and more complex data and settings, the tasks become more and more complex and, often, applicants do not only have to apply a machine learning technique but also to inspect and interpret the result. Based on insight gained this way, an improvement or focus of the model can be done [23]. In this setting, a severe drawback of many state-of-the-art machine learning tools occurs: they act as black-boxes. In consequence, applicants cannot interpret the results and it is hardly possible to substantiate a machine classification by a semantic explanation, or to change the functionality of the model based on this insight.

Prototype-based methods enjoy a wide popularity in various application domains due to their very intuitive and simple behavior: they represent their decisions in terms of typical representatives contained in the input space and a

N. Mana, F. Schwenker, and E. Trentin (Eds.): ANNPR 2012, LNAI 7477, pp. 14–23, 2012.

classification is based on the distance of data as compared to these prototypes [12]. Thus, models can be directly inspected by experts since prototypes can be treated in the same way as data. Popular techniques in this context include simple learning vector quantization (LVQ) schemes and extensions to more powerful settings such as variants based on cost functions or metric learners [18,21]. Robust soft LVQ (RSLVQ) as proposed in [21] constitutes one particularly interesting approach since it is based on a generic probabilistic modeling of data in terms of mixture models and it derives a learning rule based on this model by optimizing the likelihood ratio. A behavior which closely resembles standard LVQ2.1 results if modes are represented as Gaussians and the limit case of small bandwidth is considered. While the limit case as well as standard LVQ2.1 do not achieve optimum behavior already in simple model situations, as investigated in the context of the theory of online learning in the approach [1] for example, RSLVQ displays excellent generalization ability in the standard intermediate case, see e.g. the approach [20] for an extensive comparison of the technique.

With data sets becoming more and more complex, input data are often no longer given as simple Euclidean feature vectors, rather structured data or dedicated formats can be observed such as bioinformatics sequences, graphs, or tree structures as they occur in linguistics, time series data, functional data arising in mass spectrometry, relational data stored in relational databases, etc. In consequence, a variety of techniques has been developed to extend powerful statistical machine learning tools towards non-vectorial data such as kernel methods using structure kernels, recursive and graph networks, functional methods, relational approaches, and similar [6,19,8,17,10]. Recently, popular prototype-based algorithms have also been extended to deal with more general data. Diverse techniques rely on a characterization of the data by means of a matrix of pairwise similarities or dissimilarities only rather than explicit feature vectors. In this setting, median clustering as provided by median self-organizing maps, median neural gas, or affinity propagation characterizes clusters in terms of typical exemplars [7,13,5]. More general smooth adaptation is offered by relational extensions such as relational neural gas or relational learning vector quantization [9]. A further possibility is offered by kernelization such as proposed for neural gas, self-organizing maps, or different variants of learning vector quantization [15,3,16]. By formalizing the interface to the data as a general similarity or dissimilarity matrix, complex structures can be easily dealt with: structure kernels for graphs, trees, alignment distances, string distances, etc. open the way towards these general data structures [14,8].

In this contribution, we propose an extension of RSLVQ towards a kernel variant. This way, a statistically well motivated model is obtained which achieves excellent results as we will show in several benchmarks. Interestingly, albeit the method, strictly speaking, requires a semi positive definite kernel, it also yields good results if applied to arbitrary dissimilarity matrices. Corrections which turn the latter towards valid kernels can further improve the results. Now we first shortly review RSLVQ and we explain how this technique can be extended to a kernelized version. We evaluate the behavior for several benchmarks and

also show first visualizations which emphasize the interpretability of the resulting models in terms of prototypes. We conclude with a discussion.

2 Robust Soft Learning Vector Quantization

Learning vector quantization (LVQ) constitutes a very popular class of intuitive prototype based learning algorithms with successful applications ranging from telecommunications to robotics [12]. Basic algorithms as proposed by Kohonen include LVQ1 which is directly based on Hebbian learning, and improvements such as LVQ2.1, LVQ3, or OLVQ which aim at a higher convergence speed or better approximation of the Bayesian borders. These types of LVQ schemes have in common that their learning rule is essentially heuristically motivated and a valid cost function does not exist [2]. One of the first proposals which derives LVQ from a cost function can be found in [18] with an exact computation of the validity at class boundaries in [11]. One very elegant LVQ scheme which is based on a probabilistic model and which can be seen as a more robust probabilistic extension of LVQ2.1 has been proposed in [21]. This method, robust soft LVQ (RSLVQ) models data by means of a mixture of Gaussians and derives learning rules thereof by means of a maximization of the log likelihood ratio of the given data. In the limit of small bandwidth, a learning rule which is similar to LVQ2.1 but which performs adaptation in case of misclassification only, is obtained.

Assume data $\xi_k \in \mathbb{R}^n$ are labeled y_k where labels stem from a finite number of different classes. A RSLVQ network models data by means of a mixture distribution characterized by m prototypes $w_j \in \mathbb{R}^n$ with priorly fixed labels $c(w_j)$ and bandwidths σ_j. Mixture component j defines the probability

$$p(\xi|j) = K(j) \cdot \exp(f(\xi, w_j, \sigma_j^2))$$

with normalization constant $K(j)$ and function f chosen e.g. as follows

$$f(\xi, w_j, \sigma_j^2) = -\|\xi - w_j\|^2/\sigma_j^2$$

based on the Euclidean distance or a generalization thereof. This induces the probability of an unlabeled data point

$$p(\xi|W) = \sum_j P(j) \cdot p(\xi|j)$$

with prior $P(j)$ and parameters W of the model. The probability of a labeled data point is

$$p(\xi, y|W) = \sum_{c(w_j)=y} P(j) \cdot p(\xi|j)\,.$$

Learning aims at an optimization of the log likelihood ratio

$$L = \sum_k \log \frac{p(\xi_k, y_k|W)}{p(\xi_k|W)}\,.$$

A stochastic gradient ascent yields the following update rules, given data point (ξ_k, y_k)

$$\Delta w_j = \alpha \cdot \begin{cases} (P_y(j|\xi_k) - P(j|\xi_k)) \cdot K(j) \cdot \partial f(\xi_k, w_j, \sigma_j^2)/\partial w_j & \text{if } c(w_j) = y_k \\ -P(j|\xi_k) \cdot K(j) \cdot \partial f(\xi_k, w_j, \sigma_j^2)/\partial w_j & \text{if } c(w_j) \neq y_k \end{cases}$$

with learning rate $\alpha > 0$ and the probabilities

$$P_y(j|\xi_k) = \frac{P(j) \exp(f(\xi_k, w_j, \sigma_j^2))}{\sum_{c(w_j)=y_j} P(j) \exp(f(\xi_k, w_j, \sigma_j^2))}$$

and

$$P(j|\xi_k) = \frac{P(j) \exp(f(\xi_k, w_j, \sigma_j^2))}{\sum_j P(j) \exp(f(\xi_k, w_j, \sigma_j^2))}$$

With the standard Euclidean distance, equal class priors, and small bandwidth, a learning rule similar to LVQ2.1, learning from mistakes, results thereof.

Given a novel data point ξ, its class label can be determined by means of the most likely label y corresponding to a maximum value $p(y|\xi, W) \sim p(\xi, y|W)$. For typical settings, bandwidths are chosen of equal size $\sigma_j^2 = \sigma^2$, and priors are equal $P(j) = \text{const}$. Further, the simple Euclidean distance is used. Then, this rule can usually be approximated by a simple winner takes all rule, i.e. ξ is mapped to the label $c(w_j)$ of the closest prototype w_j. It has been shown in [21], for example, that RSLVQ often yields excellent results while preserving interpretability of the model due to prototypical representatives of the classes in terms of the parameters w_j.

3 Kernel Robust Soft Learning Vector Quantization

RSLVQ, albeit offering a very powerful learning algorithm, is restricted to Euclidean data only. Here we propose a kernelization of the method such that the technique becomes applicable for more general data sets which are implicitly characterized in terms of a Gram matrix only. We assume that a kernel k is fixed corresponding to a feature map Φ, hence

$$k_{kl} := k(\xi_k, \xi_l) = \Phi(\xi_k)^t \Phi(\xi_l)$$

holds for all data points ξ_k, ξ_l. We assume that prototypes are represented as linear combinations of data in the feature space

$$w_j = \sum_m \gamma_{jm} \Phi(\xi_m) \,.$$

It is reasonable to assume that they are contained in the convex hull of the data, i.e. coefficients γ_{jm} are non-negative and sum up to 1. The cost function of RSLVQ becomes

$$L = \sum_k \log \frac{\sum_{c(w_j)=y_k} P(j) p(\Phi(\xi_k)|j)}{\sum_j P(j) p(\Phi(\xi_k)|j)} \,.$$

We assume equal bandwidth $\sigma^2 = \sigma_j^2$, constant prior $P(j)$ and mixture components induced by normalized Gaussians. These can be computed in the data space based on the Gram matrix because of the identity

$$\|\Phi(\xi_i) - w_j\|^2 = \|\Phi(\xi_i) - \sum_m \gamma_{jm}\Phi(\xi_m)\|^2 = k_{ii} - 2 \cdot \sum_m \gamma_{jm} k_{im} + \sum_{s,t} \gamma_{js}\gamma_{jt} k_{st}$$

where the distance in the feature space is referred to by $\|\cdot\|^2$. Thus the update rules become $\Delta w_j = \sum_m \Delta\gamma_{jm}\Phi(\xi_m) =$

$$\alpha \cdot K(j) \cdot \begin{cases} (P_y(j|\Phi(\xi_k)) - P(j|\Phi(\xi_k)))\,(\Phi(\xi_k) - \sum_m \gamma_{jm}\Phi(\xi_m)) & \text{if } c(w_j) = y_k \\ -P(j|\Phi(\xi_k))\,(\Phi(\xi_k) - \sum_m \gamma_{jm}\Phi(\xi_m)) & \text{if } c(w_j) \neq y_k \end{cases}$$

Hence a gradient technique yields the following adaptation rules for the coefficients γ_{jm}:

$$\Delta\gamma_{jm} = \alpha \cdot K(j) \cdot \begin{cases} -(P_y(j|\Phi(\xi_k)) - P(j|\Phi(\xi_k)))\gamma_{jm} & \text{if } \xi_m \neq \xi_k, c(w_j) = y_k \\ (P_y(j|\Phi(\xi_k)) - P(j|\Phi(\xi_k)))(1 - \gamma_{jm}) & \text{if } \xi_m = \xi_k, c(w_j) = y_k \\ P(j|\Phi(\xi_k))\gamma_{jm} & \text{if } \xi_m \neq \xi_k, c(w_j) \neq y_k \\ -P(j|\Phi(\xi_k))(1 - \gamma_{jm}) & \text{if } \xi_m = \xi_k, c(w_j) \neq y_k \end{cases}$$

Note that this adaptation performs exactly the same updates as RSLVQ in the feature space provided that the prototypes can be expressed as linear combinations of data points in the feature space. To guarantee non-negative and normalized coefficients, simple normalization takes place after every adaptation step. This restriction to the convex hull of the feature space is reasonable: it has been demonstrated e.g. in [20] that RSLVQ, by learning from mistakes, does not necessarily place prototypes at typical positions of the data space if this does not further improve the classification accuracy, rather orthogonal transformations are accepted in this case, leading to unintuitive representations of the data. These ambiguities of the solution are avoided by referring to the convex hull.

4 Experiments

We compare the method to the support vector machine (SVM) and a k-nearest neighbor classifier (k-NN) on a variety of benchmarks as introduced in [4]. The data sets represent a variety of similarity matrices which are, in general, non-Euclidean. It is standard to symmetrize the matrices by taking the average of the matrix and its transposed. Further, the substitution of a given similarity by its normalized variant constitutes a standard preprocessing step, arriving at diagonal entries 1. Even in symmetrized and normalized form, the matrices do not necessarily provide a valid kernel. Hence k-NN is directly applicable, while SVM and, strictly speaking, kernel RSLVQ, are not. We observe, however, that, unlike SVM, kernel RSLVQ can deal with these data directly without any correction due to its direct optimization of the cost function by means of a gradient descent method.

There exist different standard preprocessing tools which transfer a given similarity matrix into a valid kernel, as presented e.g. in [4,14]. In general, the similarity matrix can posses negative eigenvalues which yield to an invalid kernel. Corrections are:

- *Spectrum clip:* simply set negative eigenvalues of the matrix to 0. Since this can be realized as a linear projection, it directly transfers to out-of-sample extensions.
- *Spectrum flip:* negative eigenvalues are substituted by their positive values. Again, this can be realized by means of a linear transformation.
- *Spectrum shift:* the absolute value of the smallest negative eigenvalue is added to all eigenvalues. For spectrum shift there does not exist an according linear transform. Since the transform only affects self-similarities, a possible out-of-sample extension is to let the new similarities unchanged.

These transforms are tested for kernel RSLVQ in comparison to SVM with according preprocessing and a k-nearest neighbor approach with kernel ridge regression weights. For the latter, we report results taken from [4]. We use training data in analogy to [4]. For all data sets, we also report the signature, i.e. the number of positive and negative eigenvalues of the Gram matrix, indicating the degree of non-Euclideanity of the data.

- *Amazon47*: This data set consists of 204 books written by four different authors. The similarity is determined as the percentage of customers who purchase book j after looking at book i. This matrix is fairly sparse and mildly non-Euclidean with signature $(191, 13, 0)$. Class labeling of a book is given by the author.
- *Aural Sonar*: This data set consists of 100 wide band solar signals corresponding to two classes, observations of interest versus clutter. Similarities are determined based on human perception, averaging over 5 random probands for each signal pair. The signature is $(62, 38, 0)$. Class labeling is given by the two classes: target of interest versus clutter.
- *Face Rec*: 945 images of faces of 139 different persons are recorded. Images are compared using the cosine-distance of integral invariant signatures based on surface curves of the 3D faces. The signature is given by $(794, 151, 0)$. The labeling corresponds to the 139 different persons.
- *Patrol*: 241 samples representing persons in seven different patrol units are contained in this data set. Similarities are based on responses of persons in the units about other members of their groups. The signature is $(116, 125, 0)$. Class labeling corresponds to the seven patrol units.
- *Protein*: 213 proteins are compared based on evolutionary distances comprising four different classes according to different globin families. The signature is $(171, 72, 0)$. Labeling is given by four classes corresponding to globin families.
- *Voting*: Voting contains 435 samples with categorical data compared by means of the value difference metric. Class labeling into two classes is present. The signature is $(225, 210, 0)$.

For these data sets, results for the SVM and a weighted k-NN classifier have been reported in [4]. Thereby, data are preprocessed using shift, clip, or flip to guarantee positive definiteness for SVM. The latter is used with the RBF kernel and optimized meta-parameters in [4]. For multi-class classification, the one versus one scheme has been used.

In comparison, we train a kernel RSLVQ network using the real data or its clip, flip, or shift, respectively. Results of a ten-fold cross-validation with the same partitioning as proposed in [4] are reported. Prototypes are initialized by means of normalized random coefficients γ_{jm} where the prior class label $c(w_j)$ determines the non-zero elements. Further, while training, we guarantee that prototypes are contained in the convex hull of the data by enforcing non-negative coefficients and normalized vectors after every adaptation step. The number of prototypes is taken as a small multiple of the number of classes, exact values being displayed in Tab. 1. Other meta-parameters are optimized on the data sets using cross-validation.

The results obtained on these data sets are reported in Tab. 1, whereby results for k-NN and SVM are taken from [4]. Since SVM requires a positive semidefinite matrix, only results for the corrected data are reported for SVM. For kernel RSLVQ, albeit it is defined for valid kernels only in the strict sense, a direct application for the original data leads to (often very competitive) results which are reported in Tab. 1. For every data set, the best achieved result is shown in boldface. Interestingly, in half the cases, kernel RSLVQ achieves the best result. For four out of six cases, already the performance for the original data beats the SVM result for preprocessed data. Only in two cases (Protein and Voting), kernel RSLVQ is substantially worse as compared to SVM, albeit the result still stays in the same order of magnitude. Overall, it can be inferred that kernel RSLVQ constitutes a very competitive algorithm with excellent classification results overall.

Since prototypes are represented only implicitly by means of coefficient vectors, a direct inspection of a kernel RSLVQ classifier in the same way as a standard LVQ network by inspecting the prototype vectors is not possible. There are two possibilities which still allow an intuitive inspection of the result: since prototypes are contained in the convex hull of the data, it is possible to approximate prototypes by means of the closest data point without too much loss of information. This approximation by exemplars enables its inspection in the same way as data points. As an alternative, pairwise dissimilarities of data and prototypes are given for both, prototypes in its original form as well as exemplar based approximations. Thus it is possible to display data and prototypes in two dimensions by means of a standard non-linear dimensionality reduction technique such as t-SNE which relies on dissimilarities only [22].

To illustrate this possibility, we visualize the Aural Sonar and the Voting data set by means of t-SNE. For both cases, a kernel RSLVQ model is trained using only one prototype per class. The respective closest exemplar is marked in the projection. Fig. 1 displays the results. Obviously, representative discriminative positions are chosen as prototypes which have the potential to offer

Table 1. Results of kernel RSLVQ in comparison to SVM and k-NN on different benchmark data. The test error is reported, the standard deviation is given in parenthesis and best results are shown in boldface.

	k-NN	SVM	kernel RSLVQ		prototypes
Amazon47	16.95 (4.85)	75.98 (7.33)	**15.00**	(0.33)	94
clip	17.68 (4.75)	81.34 (4.77)	**14.63**	(0.26)	
flip	17.56 (4.91)	84.27 (4.33)	**16.70**	(0.33)	
shift	17.68 (4.75)	77.68 (6.14)	**13.78**	(0.23)	
Aural Sonar	17.00 (7.65)	14.25 (7.46)	**12.50**	(0.48)	10
clip	14.00 (6.82)	13.00 (5.34)	**12.50**	(0.48)	
flip	12.75 (6.42)	13.25 (5.31)	**12.00**	(0.35)	
shift	13.50 (6.73)	14.00 (5.61)	**13.00**	(0.43)	
Face Rec	4.23 (1.43)	3.92 (1.29)	**3.67**	(0.02)	139
clip	4.15 (1.32)	4.18 (1.25)	**3.67**	(0.02)	
flip	4.15 (1.32)	4.18 (1.32)	**3.65**	(0.02)	
shift	4.07 (1.33)	4.15 (1.33)	**3.88**	(0.01)	
Patrol	**11.88** (4.42)	40.73 (5.95)	17.29	(0.36)	24
clip	**11.56** (4.54)	38.75 (4.81)	17.91	(0.18)	
flip	**11.67** (4.24)	47.29 (5.90)	18.43	(0.24)	
shift	**13.23** (4.48)	40.83 (5.37)	23.33	(0.30)	
Protein	29.88 (9.96)	**2.67** (2.97)	29.06	(0.27)	20
clip	30.35 (9.71)	**5.35** (4.60)	10.00	(0.26)	
flip	31.28 (9.63)	**1.51** (2.36)	3.13	(0.10)	
shift	30.35 (9.71)	**23.49** (7.31)	34.65	(0.31)	
Voting	5.80 (1.83)	**5.52** (1.77)	9.42	(0.05)	20
clip	5.29 (1.80)	**4.89** (2.05)	9.42	(0.05)	
flip	5.23 (1.80)	**4.94** (2.03)	9.42	(0.05)	
shift	5.29 (1.80)	**5.17** (1.87)	9.42	(0.05)	

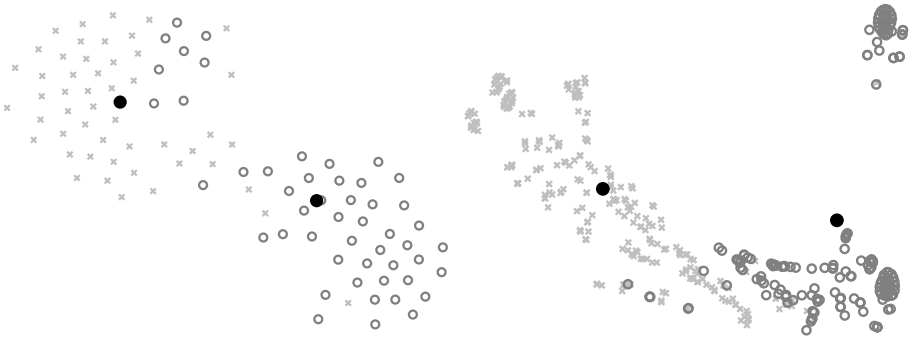

Fig. 1. Visualizing the Aural Sonar (left) and Voting data (right) sets together with representative exemplars approximating the prototypes of a kernel RSLVQ classifier using t-SNE

interpretability of the results. Thereby it is vital that, unlike support vectors in SVM, representative positions are chosen as prototypes and its number is fixed a priori.

5 Discussion

We have proposed an extension of RSLVQ to a kernel variant and we have shown that this technique yields excellent results on a variety of benchmarks, reaching the classification accuracy of the SVM in all cases. Thereby, unlike SVM, a representation of the data in terms of representative prototypes is given, and the model can be interpreted as a probabilistic mixture model induced by the prototypes, provided the considered similarity measure is a valid kernel. The latter can be achieved by using e.g. flip or clip. In most cases, also the raw similarity matrix can be used albeit it does not constitute a valid kernel. Since data and classification is based on similarities, standard visualization tools such as t-SNE allow to non-linearly project data onto the plane and to inspect the obtained result. We have demonstrated this opportunity for two simple cases, the visualization of more advanced settings being the subject of ongoing work.

While kernelization greatly enhances the applicability of RSLVQ to complex settings, it has the drawback that it trades linear complexity by quadratic one caused by the quadratic size of the similarity matrix. This makes the technique unsuited if large data sets are dealt with. Popular approximation algorithms include e.g. the Nyström approximation to substitute the full Gram matrix by a low-rank counterpart, or patch processing which processes streaming data consecutively in patches relying on a linear subpart of the full Gram matrix only. See e.g. the publications [24,25] for these techniques and [25] for first successful applications in the context of prototype based methods and LVQ schemes. These techniques can directly be integrated into kernel RSLVQ inducing linear time approximation schemes. It is the subject of future work to evaluate the performance of these approximation schemes.

Acknowledgement. This work has been supported by the DFG under grant number HA2719/7-1 and by the CITEC center of excellence.

References

1. Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of LVQ algorithms. Journal of Machine Learning Research 8, 323–360 (2007)
2. Biehl, M., Hammer, B., Verleysen, M., Villmann, T. (eds.): Similarity-Based Clustering. LNCS (LNAI), vol. 5400. Springer, Heidelberg (2009)
3. Boulet, R., Jouve, B., Rossi, F., Villa, N.: Batch kernel SOM and related Laplacian methods for social network analysis. Neurocomputing 71(7-9), 1257–1273 (2008)
4. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. JMLR 10, 747–776 (2009)

5. Cottrell, M., Hammer, B., Hasenfuss, A., Villmann, T.: Batch and median neural gas. Neural Networks 19, 762–771 (2006)
6. Frasconi, P., Gori, M., Sperduti, A.: A general framework for adaptive processing of data structures. IEEE TNN 9(5), 768–786 (1998)
7. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
8. Gärtner, T.: Kernels for Structured Data. PhD thesis, Univ. Bonn (2005)
9. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity datasets. Neural Computation 22(9), 2229–2284 (2010)
10. Hammer, B., Micheli, A., Sperduti, A.: Universal approximation capability of cascade correlation for structures. Neural Computation 17, 1109–1159 (2005)
11. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15(8-9), 1059–1068 (2002)
12. Kohonen, T.: Self-Oganizing Maps, 3rd edn. Springer (2000)
13. Kohonen, T., Somervuo, P.: How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9), 945–952 (2002)
14. Pekalska, E., Duin, R.P.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific (2005)
15. Qin, A.K., Suganthan, P.N.: Kernel neural gas algorithms with application to cluster analysis. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), vol. 4, pp. 617–620. IEEE Computer Society, Washington, DC (2004)
16. Qin, A.K., Suganthan, P.N.: A novel kernel prototype-based learning algorithm. In: Proc. of the 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK (August 2004)
17. Rossi, F., Villa-Vialaneix, N.: Consistency of functional learning methods based on derivatives. Pat. Rec. Letters 32(8), 1197–1209 (2011)
18. Sato, A., Yamada, K.: Generalized Learning Vector Quantization. In: NIPS (1995)
19. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: Computational capabilities of graph neural networks. IEEE TNN 20(1), 81–102 (2009)
20. Schneider, P., Biehl, M., Hammer, B.: Distance learning in discriminative vector quantization. Neural Computation 21, 2942–2969 (2009)
21. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Comput. 15, 1589–1604 (2003)
22. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-sne. JMLR 9, 2579–2605 (2008)
23. Vellido, A., Martin-Guerroro, J.D., Lisboa, P.: Making machine learning models interpretable. In: ESANN 2012 (2012)
24. Williams, C., Seeger, M.: Using the nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13, pp. 682–688. MIT Press (2001)
25. Zhu, X., Gisbrecht, A., Schleif, F.-M., Hammer, B.: Approximation techniques for clustering dissimilarity data. Neurocomputing 90, 72–84 (2012)