ESANN'2005 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 27-29 April 2005, d-side publi., ISBN 2-930307-05-6.

Classification using non-standard metrics

Barbara Hammer[1] and Thomas Villmann[2]

1- Clausthal University of Technology, Institute of Computer Science,
Clausthal-Zellerfeld, Germany, hammer@in.tu-clausthal.de

2- University of Leipzig - Clinic for Psychotherapy and Psychsomatic
Medicine, Leipzig, Germany, villmann@informatik.uni-leipzig.de

Abstract. A large variety of supervised or unsupervised learning algorithms is based on a metric or similarity measure of the patterns in input space. Often, the standard euclidean metric is not sufficient and much more efficient and powerful approximators can be constructed based on more complex similarity calculations such as kernels or learning metrics. This procedure is benefitial for data in euclidean space and it is crucial for more complex data structures such as occur in bioinformatics or natural language processing. In this article, we review similarity based methods and its combination with similarity measures which go beyond the standard Euclidian metric. Thereby, we focus on general unifying principles of learning using non-standard metrics and metric adaptation.

1 Introduction

The key ingredient of popular machine learning methods such as support vector machines (SVM), learning vector quantization (LVQ), self-organizing maps (SOM), or k-nearest neighbor classification is a similarity measure of the data in the input space. It allows to judge the similarity of the data to given representative points which determine the classification, i.e. prototypes, support vectors, or stored training patterns. Similarity based classification models have the advantage that their behavior is often sparse and simple, since the output of a classifier is determined by the similarity of a given data point to (usually few) prototypical cases. Typical learning rules (e.g. SOM, LVQ, or kernel-Adatron) can be motivated by intuitive principles such as Hebbian learning, possibly extended by a winner-takes-all dynamics and neighborhood cooperativeness.

However, the choice of the metric for these methods is directly connected to the representation of data and it crucially influences the efficiency, accuracy, and generalization ability of the results. Depending on the respective application, different aims need to be fulfilled: the similarity measure should possess the flexibility to capture the complexity inherent in the learning task; a powerful metric allows for sparse models and a natural representation of data since complex decision borders and data preprocessing are part of the similarity measure. At the same time, the similarity measure should be simple enough such that an efficient computation and good generalization ability can be guaranteed; sparse metrics which focus on relevant information allow to reduce the influence of noise and uncertainty in the data set.

Naturally, the choice of an appropriate metric depends on the given learning task and it is often difficult and time consuming to find the right similarity measure for a concrete problem. However, several factors influence the spectrum: learning models put constraints on the metric choice such as specific requirements, e.g. positive definiteness. The training data account for further design criteria, e.g. its amount of noise, dimensionality, inherent invariances, or its specific possibly non-Euclidian format. Thus, it is worth considering general demands and possibilities to combine models, data types, and metric choices since this triple determines the general framework in which concrete realizations can take place. We will focus on this aspect in the first part of the article.

In the second part of this article, we will emphasize possibilities which allow an automatic adaptation of the metric based on additional information given within the concrete learning task. This idea of learning metrics is particularly interesting since it allows an automatic choice of metric parameters which fit the specific situation based on data dependent information. Several realizations of this general approach have been proposed in the literature, starting from simple pruning mechanisms up to metrics such as the Fisher kernel which incorporate a statistical model of the given data. From an abstract point of view, the different models of learning metrics automatically include metric sparsity and flexibility within an exact mathematical learning objective for the given task. This constitutes a step towards automatic model optimization for metric-based learning.

2 Models

We consider supervised and unsupervised classification and clustering models which compute an output based on some form of similarity measure. Thereby, the term similarity measure is understood in a broad sense and covers metrics and kernels as specific cases.

2.1 Unsupervised models

Unsupervised similarity based learning models aim at data clustering, representation, and visualization. Thereby, the objectives might be diverse and depend on the problem at hand. There exist a lot of different models like EM approaches for standard or fuzzy k-means [4], generative topographic mapping (GTM) [5], kernelized topographic mapping [81], ISOMAP, ISODATA, multi-dimensional scaling, neural gas [57] or self-organizing maps [45], to name just a few. In our context, SOM can serve as an excellent example since a variety of typical techniques and extensions to more complex metrics exists. Therefore, first, we focus on the SOM as proposed by Kohonen which constitutes a very popular approach with successful applications in different areas [45]. In the last paragraph of this section we shortly review the alternative models.

2.1.1 Standard SOM

For standard SOM, codebook vectors or prototypes $\vec{w}_r$ in data space are arranged on a regular lattice, often given by a two-dimensional rectangular topology. A new data point $\vec{x}$ is mapped to the winning prototype $\vec{w}_r$ for which $d(\vec{w}_r, \vec{x})$ is minimum, whereby d denotes the standard Euclidian metric. Learning often takes place in Hebb style by adapting the weights of the winner and its neighborhood into the direction of the given training data. Thus, for standard SOM, the relevant aspects concerning the metric and the corresponding data space are threefold:

1. the Euclidian **metric** d is used to determine the winner and, consequently, the output of the map,
2. representative **prototypes** are elements in the vector space spanned by the input patterns,
3. **adaptation** of prototypes takes place into the direction of input patterns; this direction can be seen as the derivative of the squared Euclidian metric with respect to the prototypes.

Thus, several points are to be considered if SOM is extended to non-standard metrics.

2.1.2 Choice of the metric in SOMs

The Euclidian metric d can be extended to a more complex similarity measure which is better adapted to the considered problem. This can be performed explicitly, substituting d e.g. by a similarity measure for more complex structures (e.g. a metric for graphs [23] or microarray profiles [46]). Although there is in principle no restriction on the function d in this procedure, d is usually chosen as a symmetric, positive semidefinite measure in these cases. Then, the winner can be directly computed using d.

Alternatively, the metric can be changed implicitly by an extension of the dynamics of winner computation. This method has been proposed for temporal, sequential, or more general recursive data such as trees and graph structures: the winner computation takes place in the context set by previous computation steps, thus substituting d by a recursive distance computation. The temporal Kohonen map and the recursive SOM constitute early proposals for time series, where the distance in time step t is given by the average of the current distance and the distance in the previous time step, leading to a leaky integration over the whole sequence [8, 48]. More powerful models have recently been proposed which use a richer representation of the context [25, 26, 73, 87]. First approaches to explicitly characterize the similarity measure which arises from this recursive dynamics can be found in [29].

2.1.3 Choice of the prototypes in SOMs

Prototype representation and adaptation are often related and can be based on different training schemes for SOM. The original SOM learning rule has been proposed as a biologically plausible heuristic and an exact mathematical investigation is difficult [10]. An elegant general view of SOM training can be derived for a variant of the original SOM for which an intuitive energy function exists also in the continuous case [36]. Standard SOM training can be seen as an (approximate) stochastic gradient descent of this cost function. Naturally, alternative optimization schemes can be derived. A very popular one is realized by the batch learning rule for SOM, which iteratively determines the assignment of data points to closest prototypes and the location of prototypes as the centre of gravity of the data points weighted by the neighborhood degree of their respective winner. As shown in [37], this procedure results as limit case of an EM minimization scheme. Prototype representation and adaptation can be transferred to more general metrics based on these principles: from an abstract point of view, SOM training for general metrics can be seen as optimization of the underlying cost function where the standard Euclidian metric d is substituted by an (either directly or recursively computed) non-standard distance measure. Taking the derivative of this term yields online SOM training for non-standard metrics, an EM approach allows to derive batch versions.

However, in practice, this principle often faces severe problems such that specific modifications are necessary. Online training can be derived in this way if the corresponding similarity measure is differentiable and data are embedded in a real-vector space (see e.g. [2]). Still, the distance computation might be costly depending on the form of d in particular for high-dimensional data. In addition, an optimization by means of a simple gradient descent might face numerical problems for more complex similarity measures such as e.g. recursively computed distances. For this case, the problem of long-term dependencies which is well known from supervised recurrent networks occurs and, usually, approximations are considered instead, as pointed out in [28].

If data are discrete such as graphs, trees, or sequences which are compared by an appropriate distance measure such as the edit distance, the derivatives do not exist. In addition, data are not embedded in a real-vector space such that a smooth adaptation of prototypes is not possible. In such cases, it is necessary to substitute the derivatives of the metric and prototype adaptation. One possibility is to approximate derivatives and smooth adaptations for online learning by small discrete steps, such as proposed for graph structures based on the edit distance in [23]. Alternatively, one can restrict to the discrete and usually rough space given by the training points itself. Batch learning can be performed in this space substituting the mean value by the median, i.e. setting the prototypes as the training patterns which minimize the respective function of the M-step, as proposed in [46]. This way, the SOM can be used for domains where only pairwise distances are computed such as protein sequences which are compared by their homology [46] or web sites which are clustered based on usage information such as log files [64]. The procedure has the additional benefit that only pairwise distances of the given data points are necessary. Thus, distances need to be computed only once and also computationally complex metrics can be considered. Moreover, the method can also be used if no closed formula for d, but only the distance matrix is available.

Note that the specific choices how to realize cost function minimization in the concrete scenarios have consequences on the potential application area: the realizations differ in the fact whether the similarity measure needs to fulfill additional criteria such as differentiability or not, whether prototypes are represented in terms of data points or a surrounding continuous space must exist for prototype representation, whether a distance matrix is sufficient or the similarity measure is computed afresh after each adaptation step by means of a closed form. These aspects have immediate consequences on the efficiency, flexibility, and applicability of the algorithm for concrete settings.

2.1.4 Alternative models

In analogy to SOM, alternative unsupervised learning schemes incorporating non-standard metrics have been proposed. Naturally, extensions of SOM to problem adapted topologies such as growing topologies, tree structures, or non-

Euclidian lattices can be readily combined with non-standard metrics [12, 17, 65, 83]. These extensions are particularly relevant for non-standard metrics since the data topology is usually not Euclidian in these cases.

However, most qualitatively different models of self-organization using non standard metrics arise from a different objective of training. The quantization error is minimized by EM approaches for standard k-means, or, allowing soft assignments, classical fuzzy-k-means [4]. These algorithms have been generalized to various more general metrics to achieve a better representation of data or an alternative shape of clusters, see e.g. [38]. Alternative very powerful formulations model the SOM by probabilistic mixture models, e.g. the generative topographic map [5] or the approach proposed in [37]. These formulations are very flexible due to the generic formalization and they allow to incorporate more general metrics by substituting the statistical components for data or noise which are classically Gaussian components by more complex models, e.g. a member of the exponential family or a hidden Markov model [37, 61, 76]. The form of the components thereby depends on the form of the data which might be discrete, continuous, structured, or even of mixed form [50]. Further vector quantization approaches aim at optimum information presentation or information transfer, such as the (also kernelized) model proposed in [81] or the approach [55] which has been extended to general proximity data in [22]. Finally, a variety of algorithms is based on the objective to find an optimum visualization of given data in the plane, such as ISOMAP, ISODATA, multidimensional scaling, or variations thereof [6, 52, 89]. Here, specific emphasize is laid on metrics which mirror the intrinsic Riemannian metric of the considered data manifold which is usually a low-dimensional subset of the surrounding Euclidian space.

For these models, similar design choices as for the classical SOM can be considered, including principled model considerations such as the representation and adaptation of prototypes, and concrete issues caused by the available training data such as the form of the metric and its computational complexity.

2.2 Supervised models

Unlike unsupervised learning, the objective of supervised classification models is error minimization. Thus, a natural cost function, the number of misclassifications, exists. Nevertheless, several metric-based classification models do not explicitly optimize this cost function, but they are based on intuitive heuristics.

2.2.1 k-nearest neighbor

k-nearest neighbor or instance based learning simply stores the available training data and determines its output for a new data point based on the k closest training pattern. Thus, a transfer to general metrics can easily be achieved by a substitution of the metric by a non-standard version. However, depending on the number of patterns, this approach can be very inefficient and its generalization ability is not optimum [13].

2.2.2 Learning vector quantization

LVQ offers an alternative which adapts few prototypes based on a given set of training data [45]. Basic LVQ is given by a set of prototypes $\vec{w}_r$ together with class information. An input $\vec{x}$ is mapped to the class of the winner, i.e. the prototype $\vec{w}_r$ with smallest distance $d(\vec{w}_r, \vec{x})$. Standard LVQ learning moves the respective winner into the direction of the considered pattern $\vec{x}$ or into the opposite direction, depending on the fact whether the classification is correct. Thus, LVQ shares the aspects of SOM which are relevant for a more general metric: besides a choice of the metric, the representation and adaptation of the prototypes is to be defined.

LVQ itself does not possess a cost function in the continuous case, thus adaptations of original LVQ to more general metrics are often based on heuristics as proposed e.g. in [27] for recursive data. Usually, the metric d is substituted by a problem specific version, but adaptation of the prototypes takes place as in standard LVQ using Hebbian learning. Various modifications of LVQ based on a cost function have been proposed in the literature [66, 68]. Interestingly, though error minimization is related to the latter model, the objective is large margin optimization, i.e. structural risk minimization comparable to SVM [30]. For these cost functions, an integration of more general differentiable metrics is possible and training can take place as a stochastic gradient descent as shown in [32]. If the metrics can be interpreted as a kernelized version of the Euclidian metric, e.g. $-d$ is symmetric and conditionally positive definite, dimensionality independent large margin bounds which have been derived for LVQ also hold for the generalized version and good generalization ability can be guaranteed [32, 30]. This fact is particularly important for non-standard metrics since often, a high dimensional and only sparsely covered space is considered. However, a batch approach to minimize a given cost function of LVQ with explicit solution or a formulation of LVQ type algorithms for discrete data structures have not yet been proposed in the literature.

2.2.3 SVM

SVM is usually not introduced as metric based approach; however, its dual formulation in terms of kernel values and adaptation schemes such as kernel adaptron share important aspects of distance based learning: (usually sparse) solutions can be obtained based on a given kernel matrix of the data points. Therefore, the design of appropriate similarity measures, i.e. kernels for given data structures is of particular importance for SVM and ideas of kernel and metric design are closely related [78]. Moreover, the SVM principle gives rise to various learning algorithms which tackle problems different from crisp classification such as classification given fuzzy memberships [40] or the unsupervised task of approximating the support of an unknown data distribution [14].

3 Metrics

A variety of different metrics which go beyond the standard Euclidian metric have been proposed in the literature. Thereby, the approaches differ with respect to the considered data types – real vectors or more complex structures, e.g. graphs or sequences –, the intended semantic meaning – e.g. incorporating invariances, additional information, or statistical properties of data –, and the efficiency of computation – ranging from simple summation up to methods which require dynamic programming or optimization. Naturally, the metric has to be chosen in such a way, that it fits the given data structure, but also the regularities within the data and the noise model [16]. Here, we propose a taxonomy of metric designs based on the given data structures.

3.1 Discrete values

Often, only single values, e.g. elements of a distance matrix, are available and the classification takes place based on the given distance matrix. Usually, the difficulty to deal with such situations does not lie in the choice of the metric (since the matrix is available) but the design of appropriate algorithms which can work with this limited information. Several approaches have been proposed based on different optimization principles in the unsupervised scenario [21, 69]. Problems might arise for partial information where missing entries have to be restored, see e.g. [79], or if mixed data which contain discrete as well as continuous values are considered [50]. A further interesting line of research considers discrete data given as contingency tables, e.g. survey data [11]. Here, similarity measures compare appropriately scaled rows or columns from the complete disjunctive table or summations thereof such as Burt tables. For the supervised case an adaptive scheme is proposed for nominal data using the value difference metric [9].

3.2 Real vectors

Several proposed metrics deal with real vectors in the standard Euclidian space, but they try to incorporate appropriate data characteristics or invariances into the design to achieve a better accuracy. This idea includes l_p (Minkowski) norms with $p \neq 2$ [16], reduction of the dimensionality [14], or feature selection as summarized e.g. in [24]. Further popular methods rely on an appropriate weighting of dimensionalities or a full (possibly local) matrix, such as a variation of the Mahalanobis distance which takes the correlation of data dimensions into account [13, 20, 33, 38, 74]. For two-dimensional settings, elements can possibly be interpreted as a complex number, e.g. amplitude and phase of spectra. For such data, complex valued networks can be used [35]. More complex invariances can be integrated into the setting by requiring invariance e.g. with respect to certain transformations such as realized by the tangent distance, for example [88]. Finally, classical kernels such as the RBF kernel, or ANOVA kernels provide similarity measures for standard vectors which are related to Gaussian shaped contours or higher order correlations [67].

3.3 Manifolds

Often, real vectors are elements of a (low dimensional) submanifold of the surrounding space. In such cases, the Euclidian metric or another global metric does not fit the local structure of the data manifold, but the inherent Riemannian metric should be used [58]. This problem occurs, e.g. when projecting high dimensional data onto lower dimensions. Several different metrics have been proposed in this context which are based on the assumption that the local distances of neighbors can be computed in the standard way, however, the global distances need to be computed along the graph which is spanned by the data points. Several different possibilities to compute an appropriate metric, based e.g. on the shortest path or an average short connection have been proposed [3, 51, 53].

The situation becomes more complex, if only specific directions within the manifold are relevant, e.g. those directions which influence the output classification. In this case, the Fisher information matrix of the class distribution given input x can be used to model the local similarity; an extension to the whole manifold can be done in an exact way using path integrals, or by efficient approximations thereof [62].

3.4 Sequences

Sequences occur naturally in several domains such as speech, text, DNA- and protein sequences, temporal data, or spikes [7]. They might be given as vectors of fixed dimensionality in a real-vector space. However, due to the spatial structure, correlations of neighbored points and the principled shape are of particular importance when assessing similarity. Correspondingly, specific similarity measures which take this fact into account have been proposed: the locality improved kernel weights the similarity of local correlations of given sequences [31, 72]. Correlation coefficients focus on the shape of the sequences rather than the amplitude [59, 60]. For spike sequences, the distance of the spike times between the two candidates can be extended to a similarity measure [7]. The general form of the data is also emphasized by a treatment of the sequence as a function as proposed e.g. in [63]. In particular, a functional interpretation of a given sequence can also deal with sequences which are sampled in a different way. Note that sequences are embedded in a real-vector space in these cases and smooth online adaptation of prototypes is possible.

For general sequences with different length, several mostly discrete similarity measures have been proposed. Often, the similarity is based on an alignment of the sequences and a corresponding cost function, e.g. the edit distance or some weighted version thereof [34, 46]. This setting is usually tackled in a batch way, referring to only the distance matrix of the training pattern. However, a smooth (so computationally demanding) adaptation of prototypes is also possible as demonstrated in [71]: the similarity of sequences is determined by dynamic programming, and adaptation is performed on the respective warping path in the standard Hebb way. An alternative similarity measure for sequences relies on common substructures of two inputs, i.e. the number of contiguous or non-contiguos substrings which two inputs have in common. Different realizations thereof have been proposed together with efficient computation schemes based on dynamic programming or suffix trees [18, 54, 56]. Still, the methods are quite demanding and usually only applied to compute the relevant part of the distance matrix in batch algorithms.

A quite general alternative to the comparison of subparts of sequences is offered by similarity measures derived from a statistical model by means of the Fisher information, as introduced in [39]. This way, hidden Markov models or alternative statistical generators yield natural similarities. Note that these two principles are not disjoint but substring methods can be seen as a special case of metrics based on a probabilistic model as shown in [85].

3.5 Trees and graphs

For more general structures such as trees or graph structures, the computational burden to compute discriminative metrics increases [19], however the principled ideas of metric design transfer from the more simple case of vectors or sequences. Trees are often rooted and possess a natural processing order, such that specific kernels e.g. stemming from natural language parsing or focussing on subtrees can readily be defined [75, 82, 86]. Graph kernels can be based on their similarity if restricting to contained paths [42, 47]. This similarity measure can be computed efficiently by means of matrix exponentiation and the Laplacian. Alternatively, discrete comparisons of given graphs using the edit distance or the generalized median have been proposed [23, 41].

4 Metric adaptation

Methods which automatically adapt the metric based on the given learning task are particularly interesting since they allow to automate the model selection process. Thereby, a flexible metric should be tuned in such a way that the parameters fit the regularities of the data set without adapting to noise and outliers.

4.1 Learning metric parameters

Several approaches deal with metrics the form of which is fixed, e.g. Mahalanobis distance, but the metric includes real valued parameters which are adapted based on general optimization criteria. This principle is used e.g. in fuzzy clustering, where metric parameters are chosen as optima of the quantization error [38]. For supervised LVQ, relevance learning has been introduced in [33]. Thereby, relevance parameters which weight the data dimensions are included in the metric and adapted based on the objective function of generalized LVQ. Interestingly, this method allows a much better accuracy on the training set without decreasing the generalization ability of the approach since it remains a large margin optimization scheme [30]. It should be mentioned that an appropriate metric which achieves a good separation of classes allows to train subsequent nearest neighbor classifiers based on very few examples as demonstrated in [15].

Alternative objectives of metric optimization are due to information theoretic learning [77] or regression models [90]. In the first case the metric is adapted according to the increase of the mutual information or information energy between data and class information [1, 84] wheras in the latter one the optimization of the regression model determines the metric adaptation [90].

4.2 Learning the metric form

Naturally, there exists a smooth transition between learning metric parameters and learning the metric form. Relevance factors, for example, allow to select features and, consequently, to change the representation of data, i.e. the metrics in a principled form. Alternatively, a variety of direct feature selection and feature creation schemes have been proposed, see e.g. [24, 80].

Several proposals fit a statistical model to the data and derive a metric based on the generative model. Thereby, the form of the metric is determined by the form of the statistical model. The Fisher kernel as proposed in [39] constitutes a popular example of this approach. A statistical model, often a hidden Markov model, is fit to the given training data and adapts to statistically relevant aspects of the data. These aspects serve for the differentiation of data points for classification when comparing two data points by means of the Fisher kernel. An alternative approach has been proposed for unsupervised learning in [44, 43, 70]. Here, auxiliary information guides the similarity measure and only those aspects of data are monitored which have an influence on given auxiliary information. From a technical point of view, this learning metrics principle is realized by using the Fisher information of a generative model which describes the dependency of the auxiliary information on the input data.

These principles allow a very flexible adaptation of the metric based on general principles. The opposite point of view is taken in the approach [49], where only the kernel matrix for given data is adapted. This approach is interesting when training as well as (unlabeled) test data are available and a similarity measure which mimics a potential labeling should be inferred for the test data. In [49], the problem is solved by determining the matrix such that the classification error of the training set is minimum and, to avoid overfitting, the matrix is regularized to achieve good generalization performance.

5 Discussion

Classification performance for a given task strongly depends on the chosen model, the used metric type and related parameters. We reviewed in the contribution approaches dealing with adequate, possibly non-standard, metrics for data representation and classifcication. We emphasized that the choice of the metric must fit the used model. However, finding an apropriate metric for the given task is still a difficult problem. A very interesting possibility in this line is an automatic learning of the metric within the given task. In this tutorial we have summarized recent approaches to optimize metrics within a given frame work (metric type) based on different objectives such as the classification error, information theoretic measures etc.

References

[1] R. Andonie and A. Cataron. An information energy LVQ approach for feature ranking. In M. Verleysen, editor, *European Symposium on Artificial Neural Networks 2004*, pages 471–476. d-side publications, 2004.

[2] P. Andras. Kernel-Kohonen networks. *International Journal of Neural Systems*, 12:117–135, 2002.

[3] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectr al clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[4] J. C. Bezdek. Self-organization and clustering algorithms. In *Proc. 2nd Joint Technology Workshop on Neural Networks and Fuzzy Logic*, volume I, pages 143–158, 1991.

[5] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: A principled alternative to the self-organizing map. Technical Report NCRG/96/015, Neural Computing Research Group, Aston University, 1996.

[6] E. Bonabeau. Graph multidimensional scaling with self-organizing maps. *Information Sciences*, 143(1–4):159–180, June 2002.

[7] A. Carnell and R. Daniel. Linear algebra for time series of spikes. In *this volume*.

[8] G. J. Chappell and J. G. Taylor. The temporal Kohonen map. *Neural Networks*, 6:441–445, 1993.

[9] V. Cheng, C.-H. Li, J. Kwok, and C.-K. Li. Dissimilarity learning for nominal data. *Pattern Recognition*, 37(7):1471–1477, 2004.

[10] M. Cottrell, J. C. Fort, and G. Pagès. Two or three things that we know about the Kohonen algorithm. In M. Verleysen, editor, *Proc. ESANN'94, European Symp. on Artificial Neural Networks*, pages 235–244, Brussels, Belgium, 1994. D facto conference services.

[11] M. Cottrell, S. Ibbou, and P. Letrémy. SOM-based algorithms for qualitative variables. *Neural Networks*, 17:1149–1168, 2004.

[12] J. Dopazo and J. Carazo. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of Molecular Evolution*, 44(2):226–233, 1997.

[13] R. Duda, P. Hart, and D. Storck. *Pattern classification*. Wiley, 2000.

[14] P. Evangelista, P. Bonissone, M. Embrechts, and B. Szymanski. Fuzzy ROC curves for the one class SVM: application to intrusion detection. In *this volume*.

[15] M. Fink. Object classification from a single example utilizing class relevance pseudo-metrics. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[16] D. Francois, V. Wertz, and M. Verleysen. Non-euclidean metrics for similarity search in noisy datasets. In *this volume*.

[17] R. Freeman and H. Yin. Adaptive topological tree structure (ATTS) for document organisation and visualisation. *Neural Networks*, in press.

[18] T. Gärtner. A survey of kernels for structured data. *SIGKDD explorations*, 2003.

[19] T. Gärtner, P. A. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Sixteenth Annual Conference on Computational Learning Theory and Seventh Kernel Workshop (COLT-2003)*. 2003.

[20] J. Goldberger and S. Roweis. Hierarchical clustering of a mixture model. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[21] T. Graepel, M. Burger, and K. Obermayer. Self-organizing maps: generalizations and new optimization techniques. *Neurocomputing*, 21(1–3):173–90, 1998.

[22] T. Graepel and K. Obermayer. A stochastic self organizing map for proximity data. *NeuralComputation*, 11:139–155, 1999.

[23] S. Günter and H. Bunke. Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, 23:401–417, 2002.

[24] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

[25] M. Hagenbuchner, A. Sperduti, and A. Tsoi. Contextual processing of graphs using self-organizing maps. In *this volume*.

[26] M. Hagenbuchner, A. Sperduti, and A. Tsoi. A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14:191–505, 2003.

[27] M. Hagenbuchner, A. C. Tsoi, and A. Sperduti. A suprevised self-organising map for structured data. In *Advances in Self-Organising Maps*.

[28] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57:3–35, 2004.

[29] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. Recursive self-organizing network models. *Neural Networks*, to appear.

[30] B. Hammer, M. Strickert, and T. Villmann. On the generalization ability of GRLVQ networks. *Neural Processing Letters*, page in press, 2005.

[31] B. Hammer, M. Strickert, and T. Villmann. Prototype based recognition of splice sites. In U. Seiffert, L. Jain, and P. Schweitzer, editors, *Bioinformatic using Computational Intelligence Paradigms*, pages 25–56. Springer-Verlag, 2005.

[32] B. Hammer, M. Strickert, and T. Villmann. Supervised neural gas with general similarity measure. *Neural Processing Letters*, 21(1):21–44, 2005.

[33] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[34] J. Hanke and J. Reich. Kohonen map as a visualization tool for the analysis of protein sequences: multiple alignments, domains and segments of secondary structures. *Computer Applications in the Biosciences*, 12(6):447–454, 1996.

[35] T. Hara and A. Hirose. Plastic mine detecting radar system using complex-valued self-organizing map that deals with multiple-frequency interferometric images. *Neural Networks*, 17:1201–1210.

[36] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.

[37] T. Heskes. Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks*, 12(6):1299–1305, November 2001.

[38] F. Höppner, F. Klawonn, and R. Kruse. *Fuzzy Cluster Analysis.* Wiley, 1999.

[39] T. Jaakkola, M. Diekhans, and D. Haussler. *Journal of Computational Biology*, 7, 2000.

[40] J. Jayadeva, R. Khemchandani, and S. Chandra. Fuzzy proximal support vector classification via generalized eigenvalues. In *this volume.*

[41] X. Jiang, A. Münger, and H. Bunke. On median graphs: properties, algorithms, and applicationsa. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1144–1151, 2001.

[42] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th International Conference on Machine Learning*, pages 321–328. AAAI Press, 2003.

[43] S. Kaski, J. Nikkilä, M. Oja, J. Venna, P. Törönen, and E. Castren. Trustworthiness and metrics in visualizing similarity of gene expression. *BMC Bioinformatics*, 4:48, 2003.

[44] S. Kaski, J. Sinkkonen, and J. Peltonen. Bankruptcy analysis with self-organizing maps in learning metrics. *IEEE Transactions on Neural Networks*, 12:936–947, 2001.

[45] T. Kohonen. *Self-Organizing Maps.* Springer, 1995.

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

[47] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*. 2002.

[48] T. Koskela, M. Varsta, J. Heikkonen, and K. Kaski. Time series prediction using RSOM with local linear modesl. Technical Report B15, Helsinki University of Techology, Laboratory of Computational Engineering, Espoo, Finland, 1997.

[49] G. Lansckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–77, 2004.

[50] M. Lebbah, A. Chazottes, F. Badran, and S. Thiria. Mixed topological map. In *this volume.*

[51] J. Lee, A. Lendasse, and M. Verleysen. Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. *Neurocomputing*, 57:49–67, 2004.

[52] J. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, to appear.

[53] J. Lee and M. Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, to appear.

[54] C. Leslie, E. Eskin, J. Weston, and W. Noble. Mismatch string kernels for svm protein classification. In *NIPS*. 2002.

[55] R. Linsker. How to generate maps by maximizing the mutual information between input and output signals. *Neural Computation*, 1:402–411, 1989.

[56] H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins. *Journal of Machine Learnin Research*, 2:419–444, 2002.

[57] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[58] N. Mayer, J. M. Herrmann, and T. Geisel. Curved feature metrics in models of visual cortex. *Neurocomputing*, 44–46:533–539, 2000.

[59] C. Moller-Levet and H. Yin. Modelling and clusterig of gene expressions using RBFs and and a shape similarity metric. In *Lecture Notes in Computer Science*, volume 3177, pages 1–10. 2004.

[60] C. Moller-Levet, H. Yin, K.-H. Cho, and O. Wolkenhauer. Modelling gene expression time-series with radial basis function neural networks. In *IJCNN'04*, pages 1191–1196. 2004.

[61] I. Nabney, Y. Sun, P. Tino, and A. Kaban. Semisupervised learning of hierarchical latent trait models for data visualization. *IEEE Transactions on Knowledge and Data Engineering*, 17, 2005.

[62] J. Peltonen, A. Klami, and S. Kaski. Improved learning odf Riemannian metrics for exploratory data analysis. *Neural Networks*, 17:1087–1100, 2004.

[63] F. Rossi, B. Conan-Guez, and A. El Golli. Clustering functional data with the SOM algorithm. In M. Verleysen, editor, *Proc. ESANN'04*, pages 305–312. de side publications, 2004.

[64] F. Rossi, A. ElGolli, and T. Lechevallier. Usage guided clustering of web pages with the median self organizing map. In *this volume*.

[65] A. Saalbach, T. Twellmann, T. Nattkemper, A. Wismüller, J. Ontrup, and H. Ritter. A hyperbolic topographic mapping for proximity data. In *Artificial Intelligence and Applications*, 2005.

[66] A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.

[67] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT, 2001.

[68] S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15:1589–1604, 2003.

[69] S. Seo and K. Obermayer. Self-organizing maps and clustering methods for matrix data. *Neural Networks*, 17:1211–1230, 2004.

[70] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, 14:217–239, 2002.

[71] P. J. Somervuo. Online algorithm for the self-organizing map of symbol strings. *Neural Networks*, 17:1231–1240, 2004.

[72] S. Sonnenburg, G. Rätsch, A. Jagota, and K. Müller. New methods for splice site recognition. In *Proceedings of the International Conference on Artifical Neural Networks*. 2002.

[73] M. Strickert and B. Hammer. Neural gas for sequences. In *Proc. International Workshop on Self-Organizing Maps (WSOM'2003)*, pages 53–58, Kitakyushu, 2003.

[74] M. Strickert, N. Sreenivasulu, W. Weschke, U. Seiffert, and T. Villmann. Generalized relevance LVQ with correlation measures for biological data. In *this volume*.

[75] J. Suzuki, Y. Sasaki, and E. Maeda. Kernels for structured natural language data. In *NIPS*. 2003.

[76] P. Tino, A. Kaban, and Y. Sun. A generative probabilistic approach to visualizing sets of symbolic sequences. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–706. ACM Press, 2004.

[77] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415–1438, 2003.

[78] I. Tsang and J. Kwok. Distance metric learning with kernels. In O. Kaynak, editor, *Proc. International Conference on Artificial Neural Networks (ICANN'2003)*, pages 126–129, Istanbul, 2003.

[79] K. Tsuda, S. Akaho, and K. Asai. The EM algorithm for kernel matrix completion with auxiliary data. *Journal of Machine LEarning Research*, 4:67–81, 2003.

[80] S. Ullmann and E. Bart. Recognition invariance obtained by extended and invariant features. *Neural Networks*, 17:833–848, 2004.

[81] M. M. van Hulle. *Faithful Representations and Topographic Maps From Distortion- to Information-based Self-organization*. J. Wiley & Sons, Inc., 2000.

[82] J.-P. Vert. Kernel methods in computational biology, 2004.

[83] T. Villmann and H.-U. Bauer. Applications of the growing self-organizing map. *Neurocomputing*, 21(1-3):91–100, 1998.

[84] T. Villmann, F. M. Schleif, and B. Hammer. Comparison of relevance learning vector quantization with other metric adaptive classification methods. *Neural Networks*, submitted.

[85] A. Vinokurov, A. Soklakov, and C. Saunders. A probabilistic framework for mismatch and profile string kernels. In *this volume*.

[86] S. Vishwanathan and A. Smola. Fast kernels for string and tree matching. In *NIPS*. 2002.

[87] T. Voegtlin and P. F. Dominey. Recursive self-organising maps. In N. Allinson, H. Yin, L. Allinson, and J. Slack, editors, *Advances in Self-Organising Maps*, pages 210–5. Springer, 2001.

[88] J. Wood. Invariant pattern recognition: A review. *Pattern Recognition*, 29:1–17, 1996.

[89] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens. Clustering using a random walk based distance measure. In *this volume*.

[90] Z. Zhang, J. Kwok, and D.-Y. Yeung. Parametric distance metric learning with label information. In O. Kaynak, editor, *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1450–1452, Acapulco, Mexico, 2003.