Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Locally adaptive nearest neighbours

Jan Philip Goepfert, Heiko Wersing, Barbara Hammer

2021

Preprint:

Copyright Elsevier 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

Interpretable Locally Adaptive Nearest Neighbors

Jan Philip Göpfert[a], Heiko Wersing[b], Barbara Hammer[a]

[a] *Bielefeld University, Germany*
[b] *Honda Research Institute Europe GmbH, Offenbach, Germany*

Abstract

When training automated systems, it has been shown to be beneficial to adapt the representation of data by learning a problem-specific metric. This metric is global. We extend this idea and, for the widely used family of k nearest neighbors algorithms, develop a method that allows learning *locally* adaptive metrics. These local metrics not only improve performance, but are naturally interpretable. To demonstrate important aspects of how our approach works, we conduct a number of experiments on synthetic data sets, and we show its usefulness on real-world benchmark data sets.

Keywords: interpretable machine learning, metric learning, nearest neighbors

1. Introduction

Machine learning models increasingly pervade our daily lives in the form of recommendation systems, computer vision, driver assistance, etc., challenging us to realize seamless cooperation between human and algorithmic agents. One desirable property of predictions made by machine learning models is their transparency, expressed in such a way as a statement about which factors of a given setting have the greatest influence on the decision at hand – in particular, this requirement aligns with the EU General Data Protection Regulations, which include a "right to explanation" [1]. The native transparency of machine learning models varies considerably based on the form and complexity of the models, ranging from intuitive prototype-based classifiers, which allow a substantiation of a decision in the form of a typical class representative [2], to mostly opaque

black-box models found in deep learning, for which additional posterior explanation technologies are required [3]. Interestingly, several popular interpretation technologies for black-box models rely on local feature weighting schemes [4]. Moreover, machine learning models that are intrinsically based on a feature relevance weighting [5], enjoy a wide popularity in particular in medical domains to uncover relevant insight, such as the discovery of potential biomarkers [6].

Intuitive indications of which features are most or least relevant for a given model's decision can be provided by metric-learning approaches, such as GRLVQ [5], which adapts a diagonal matrix, scaling the relevance of the input features. Generalizations that use a full matrix, such as GMLVQ [7], exist, but a *single global* quadratic matrix remains the most common choice [8]. Large margin nearest neighbor learning (LMNN) implements this idea for a k-nearest neighbor (kNN) classification scheme [9]. A few approaches extend this setting to non-global matrices, such as LGRLVQ [10] and LGMLVQ [7], which can be accompanied by learning-theoretical guarantees, but they allow only one matrix per prototype, which corresponds to one metric per Voronoi cell in the input space. An extension of LMNN [11] requires an explicit partitioning of the training data and learns one metric per subset. The partitioning, commonly based on the respective class labels, is set before training and remains unchanged, which makes the extension straight-forward but inflexible. Parametric Local Metric Learning (PLML) [12] learns a smooth metric matrix function over the data manifold, but again, its specific metric matrices are based on so-called anchor points, such as the means of clusters according to some supervised algorithm. Noh et al. [13] take a different approach with Generative Local Metric Learning (GLML), where they learn an optimal local metric for a learned generative model. Fitting class-wise Gaussians, they inherit the inflexibilties that come with this assumption-heavy approach and fail to learn well-performing metrics [12]. GLML does show promise for the special case where the number of training samples is further constrained. Its original analysis is therefore focused on that particular setting.

In this work, we formulate and explore an extension of kNN to local relevance

matrices, which are *specific to a given point* and indicate the *local relevance* of the features in its region, i. e. the factors most relevant for a *specific decision* rather than the global model. Further, unlike LMNN, PLML, and GLML, we implement an online adaptation technique, which can be integrated into incremental models or models for streaming data, such as the one proposed by Losing et al. [14]. In the following, we will propose a cost function based on a differentiable approximation of the output label distribution of a kNN classifier, and we will demonstrate how to derive an intuitive local relevance learning scheme based thereon. To investigate the resulting learned, local feature relevances, and to demonstrate how they aid in interpreting data, we compute according low-dimensional embeddings.

This extension of the Locally Adaptive Nearest Neighbors [15] contains an expanded evaluation with further datasets and an embedding-based demonstration of the usefulness of the learned metrics.

2. Local metric learning for kNN classifiers

Assume data $X = \{\vec{x}^1, \ldots, \vec{x}^m\} \subset \mathbb{R}^n$ are given, with label y_i for data point $\vec{x}^i$, where labels are element of a finite number of L different labels. Assume a number $k > 0$ is fixed. A kNN classifier crucially depends on a distance measure $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Given a data point $\vec{x} \in \mathbb{R}^n$, the *neighborhood* $N(\vec{x})$ of $\vec{x}$ in X is defined as the set of k points $\vec{x}^i$ in X where $d(\vec{x}^i, \vec{x})$ is smallest. A weighted kNN classifier computes the *support* S for label y given input $\vec{x}$

$$S(y \mid \vec{x}) = \sum_{\substack{\vec{x}^i \in N(\vec{x}) \\ y_i = y}} \frac{1}{d(\vec{x}^i, \vec{x})}$$

and outputs the label y with maximum support. This definition relies on a global distance measure d such as the squared Euclidean distance measure $d(\vec{x}^i, \vec{x}) = (\vec{x}^i - \vec{x})^\mathsf{T}(\vec{x}^i - \vec{x})$. Metric learning such as LMNN [9] substitutes the Euclidean distance by a parameterized quadratic form

$$d_\Lambda(\vec{x}^i, \vec{x}) = (\vec{x}^i - \vec{x})^\mathsf{T} \Lambda \, (\vec{x}^i - \vec{x})$$

with positive semi-definite (p. s. d.) matrix Λ, which is determined based on given data. LMNN relies on the objective to change the distance such that intruders, i.e. points $\vec{x}^i$ in $N(\vec{x})$ which do not have the same label as $\vec{x}$, are moved outside $N(\vec{x})$ with a margin. This problem can be phrased as a semi-convex constraint optimization problem for the metric parameters Λ [9]. LMNN uses a global distance measure, which does not necessarily resemble the relevance of input features for the local decision $f(\vec{x})$.

In the following, we want to ask and answer, whether it is possible to (i) learn local metrics without a fixed prior decomposition of the space, and (ii) develop an online learning scheme, which carries the potential of an integration into streaming and incremental scenarios such as the self-adjusting-memory kNN [14]. We assume a local distance measure

$$d_{\Lambda_i}(\vec{x}^i, \vec{x}) = (\vec{x}^i - \vec{x})^\mathsf{T} \Lambda_i (\vec{x}^i - \vec{x})$$

where d_{Λ_i} is attached to the data point $\vec{x}^i$ and it is used whenever the distance measure from $\vec{x}^i$ to another data point is computed. Here, Λ_i is an adaptive p. s. d. matrix, which can be parameterized as $\Lambda_i = (\Omega^i)(\Omega^i)^\mathsf{T}$ with possibly low-rank matrix $\Omega^i \in \mathbb{R}^{n \times n'}$ for some $n' \leqslant n$ or even diagonal form $\Lambda_i = \operatorname{diag}\big((\lambda_1^i)^2, \ldots, (\lambda_n^i)^2\big)$.

Given an input $\vec{x}$ with desired output y, we can derive a stochastic gradient scheme to adapt these metric parameters online as follows: We approximate the output of a weighted kNN using the *softmax* function with parameter $\beta > 0$, which yields a probability distribution over all possible output labels $1, \ldots, L$:

$$P(y \mid \vec{x}) := \left(\frac{\exp(S(y \mid \vec{x})/\beta)}{\sum_{y'} \exp(S(y' \mid \vec{x})/\beta)} \right)_{y=1,\ldots,L} \in [0,1]^L$$

where local metrics d_{Λ_i} are used to evaluate the support $S(y \mid \vec{x})$, which indicates the vector of probabilities of the L output labels. Assume a desired output $y = l$ is given, this induces a probability distribution over the labels by its one-hot encoding in $\{0,1\}^L$, which we denote by $P(y \mid l)$.

Then, a suitable loss function is offered by the Kullback-Leibler divergence,

resulting in the overall error

$$
\begin{aligned}
E &= \sum_{i=1}^{m} E(\vec{x}^i, y_i) = \sum_{i=1}^{m} \mathrm{KL}\big(P(y \mid y_i) \,\big\|\, P(y \mid \vec{x}^i)\big) \\
&= -\sum_{i=1}^{m} \sum_{l=1}^{L} P(y = l \mid y_i) \cdot \log \frac{P(y = l \mid \vec{x}^i)}{P(y = l \mid y_i)} \\
&= -\sum_{i=1}^{m} \log P(y = y_i \mid \vec{x}^i) = -\sum_{i=1}^{m} \log \left(\frac{\exp(S(y_i \mid \vec{x}^i)/\beta)}{\sum_{y'} \exp(S(y' \mid \vec{x}^i)/\beta)} \right)
\end{aligned}
$$

since $P(y = l \mid y_i) = \delta_{l,y_i}$ (the Kronecker delta), where we use the identity $0 \cdot \log 0 = 0$. For stochastic gradient descent, we consider the derivative of a term w. r. t. metric parameters Ω^j_{kl} for a matrix $\Lambda_j = (\Omega^j)(\Omega^j)^\mathsf{T}$. This yields

$$
\begin{aligned}
&\frac{\partial}{\partial \Omega^j_{kl}} \left(-\log\big(\exp(S(y_i \mid \vec{x}^i)/\beta)\big) + \log\left(\sum_{y'} \exp(S(y' \mid \vec{x}^i)/\beta) \right) \right) \\
&= -\frac{1}{\beta} \cdot \frac{\partial S(y_i \mid \vec{x}^i)}{\partial \Omega^j_{kl}} \\
&\quad + \frac{1}{\beta} \cdot \frac{1}{\sum_{y'} \exp(S(y' \mid \vec{x}^i)/\beta)} \cdot \sum_{y'} \left(\frac{1}{\beta} \cdot \exp(S(y' \mid \vec{x}^i)/\beta) \cdot \frac{\partial S(y' \mid \vec{x}^i)}{\partial \Omega^j_{kl}} \right)
\end{aligned}
$$

Further,

$$
\begin{aligned}
\frac{\partial S(y' \mid \vec{x}^i)}{\partial \Omega^j_{kl}} &= \frac{\partial}{\partial \Omega^j_{kl}} \sum_{\vec{x}^o \in N(\vec{x}^i), y_o = y'} \frac{1}{d_{\Lambda_o}(\vec{x}^o, \vec{x}^i)} \\
&= \begin{cases} -1/d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)^2 \cdot \frac{\partial d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)}{\partial \Omega^j_{kl}} & \text{if } \vec{x}^j \in N(\vec{x}^i), y_j = y' \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

yields the derivative 0 for all Λ_j where $\vec{x}^j \notin N(\vec{x}^i)$. For neighbors $\vec{x}^j \in N(\vec{x}^i)$ we obtain

$$
\frac{\partial E(\vec{x}^i, y_i)}{\partial \Omega^j_{kl}} = \begin{cases} \frac{1}{\beta \cdot d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)^2} \cdot \left(1 - \frac{\exp(S(y_i|\vec{x}^i)/\beta)}{\sum_{y'} \exp(S(y'|\vec{x}^i)/\beta)}\right) \cdot \frac{\partial d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)}{\partial \Omega^j_{kl}} & \text{if } y_j = y_i \\ -\frac{1}{\beta \cdot d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)^2} \cdot \left(\frac{\exp(S(y_j|\vec{x}^i)/\beta)}{\sum_{y'} \exp(S(y'|\vec{x}^i)/\beta)}\right) \cdot \frac{\partial d_{\Lambda_j}(\vec{x}^j, \vec{x}^i)}{\partial \Omega^j_{kl}} & \text{if } y_j \neq y_i \end{cases}
$$

It is necessary to add a regularization step to prevent divergence of the parameters, e.g. a soft or hard constraint for $\det \Lambda_j$ or a restriction of the norm of the

diagonal of the matrices. If we chose the metrics in the form of diagonal matrices $\Lambda = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_n^2)$, the derivative yields $\partial d_\Lambda(\vec{x}, \vec{x}')/\partial \lambda_l = 2\lambda_l \cdot (x_l - x_l')^2$. In this case, a stochastic gradient descent directly corresponds to a Hebbian scheme: for $y_j = y_i$, diagonal terms for those dimensions l are enhanced (after normalization) which correspond to small values $(x_l^j - x_l^i)^2$; for $y_j \neq y_i$, we find the opposite. This behavior resembles popular metric learning schemes as proposed in the context of prototype-based classifiers [10, 7]. Yet, while these technologies restrict metric forms to receptive fields of prototypes, we are able to learn an individual weighting scheme for every data point of the kNN classifier. Apart from the different objective, this fact – a local weighting scheme – is the most distinguishing feature of the proposed method when compared to alternatives such as LMNN.

3. Explaining predictions using local metrics

The metrics learned by our proposed method have two important characteristics:

Diagonality Each learned weight directly corresponds to a feature in the input space. If the input space itself is interpretable, so are our learned metrics.

Locality For each point in our training set we find a local metric. If the learned metrics are interpretable, they tell us about how individual samples contribute to a prediction.

Consider a point x with a local metric that has a very small weight for some feature a and a very large weight for another feature b. A second point y can be extremely different from x with respect to feature a and still be close to x in terms of the local metric, if the two points are similar with respect to feature b.

When we use LANN to predict a label for an input, the local metrics at the k nearest neighbors responsible for the prediction give us a distribution over the feature relevances involved in the prediction. These distributions can be used

directly for any given prediction or aggregated to gain insight on different levels of detail; the specifics depend on the down-stream task.

To demonstrate the usefulness and applicability of our learned local metrics in this sense, we analyze the quality of embeddings induced by them.

4. Interpreting learned metrics via low-dimensional embeddings

Reducing the dimensionality of data, as preprocessing or for visualization, has a long history. Pearson [16] introduced the linear *Principal Component Analysis* in 1901, to determine – and project onto – the most important directions in the original feature space; in 1969, Sammon [17] proposed the non-linear *Sammon Mapping* to find low-dimensional representations with locally faithful differences. More recently, *Uniform Manifold Approximation and Projection* (UMAP) [18] has emerged as a versatile technique to preserve topology during dimensionality reduction, arguably replacing *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [19] as state of the art in dimensionality reduction.

The above-mentioned GMLVQ and LMNN, each learning a global metric, also function to reduce dimensionality, when the learned metric is used to project the data. Both algorithms learn metrics to aid classification, and so their embeddings are naturally discriminative, meaning that even with their reduced dimensionality they allow to discriminate between classes.

In addition to embedding data in (low-dimensional) space, when a dimensionality reduction algorithm (such as UMAP) uses differences between points, we can use learned metrics during embedding to not only observe the data, but also the respective metrics. Therefore, we can directly use our local metrics learned as described in Section 2 to obtain an embedding that, if the metrics behave properly, should be discriminative, similarly to LDA and GMLVQ.

Because LGMLVQ learns one metric per prototype, it is not immediately clear which metric to use when determining the distance between any two given points. We overcome this by first mapping each point $\vec{x}$ onto its distance to every one of the n prototypes $\vec{p}_1, \ldots, \vec{p}_n$:

Table 1: Accuracies (cross validation averages) for all algorithms and datasets considered in our experiments. All but the first constitute real-world data.

Dataset	kNN	LMNN	LGMLVQ	LANN
Art. Classification	0.95 ± 0.0029	0.97 ± 0.0042	0.99 ± 0.0017	0.99 ± 0.0018
Adrenal	0.82 ± 0.0293	0.81 ± 0.0550	0.77 ± 0.0391	0.88 ± 0.0171
Breast Cancer	0.95 ± 0.0079	0.95 ± 0.0113	0.92 ± 0.0155	0.94 ± 0.0077
Digits	0.94 ± 0.0075	0.96 ± 0.0050	0.87 ± 0.0168	0.96 ± 0.0054
Gamma Telescope	0.82 ± 0.0024	0.82 ± 0.0029	0.84 ± 0.0036	0.83 ± 0.0027
Image Segmentation	0.93 ± 0.0039	0.95 ± 0.0064	0.94 ± 0.0051	0.95 ± 0.0041
Ionosphere	0.77 ± 0.0468	0.79 ± 0.0353	0.76 ± 0.0375	0.90 ± 0.0306
Iris	0.93 ± 0.0340	0.95 ± 0.0152	0.93 ± 0.0298	0.96 ± 0.0120
Letter Recognition	0.88 ± 0.0019	0.91 ± 0.0027	0.87 ± 0.0072	0.91 ± 0.0025
Outdoor Objects	0.80 ± 0.0070	0.83 ± 0.0084	0.83 ± 0.0117	0.87 ± 0.0085
Pen Digits	0.98 ± 0.0012	0.99 ± 0.0008	0.99 ± 0.0016	0.99 ± 0.0011
Robot Navigation	0.79 ± 0.0094	0.80 ± 0.0081	0.79 ± 0.0067	0.83 ± 0.0088
USPS	0.94 ± 0.0025	0.95 ± 0.0023	0.95 ± 0.0027	0.95 ± 0.0024
Wine	0.95 ± 0.0160	0.96 ± 0.0164	0.63 ± 0.0880	0.96 ± 0.0153

$$\vec{x} \mapsto (d_i(\vec{x}, \vec{p}_i))^{\mathsf{T}}_{i=1,\ldots,n} \in \mathbb{R}^n, \tag{1}$$

where d_i is the learned metric of prototype $\vec{p}_i$. We subsequently find a low-dimensional embedding of this n-dimensional space using UMAP.

5. Experiments

5.1. Classification

We have implemented our proposed algorithm (henceforth referred to as LANN) in Python 3.7 within the scikit-learn[1] [20] framework, restricting the metrics to diagonal matrices as discussed above. We compare its performance against a standard kNN classifier (as provided by scikit-learn), against LMNN

[1] https://scikit-learn.org/

with a global adaptive metric (via the implementation PyLMNN[2] by John Chiotellis) – we keep $k = 5$ fixed for all three algorithms to facilitate comparability – and against Localized Generalized Matrix Learning Vector Quantization (LGMLVQ – using the open implementation[3] for scikit-learn). Each algorithm is fitted and evaluated on a number of datasets:

Artificial Classification An artificial dataset provided by scikit-learn that contains strongly relevant features, weakly relevant features, as well as redundant features. We sample 2000 data points according to the default parameters, which results in 2 classes, 20 features, of which 2 are strongly relevant, and 2 are weakly relevant.

Adrenal [21] Results from an analysis of adrenal gland metabolomics. The dataset contains 147 data points in 2 classes (adrenocortical carcinoma and adenoma), described by 32 features that relate to the underlying metabolic processes.

Wisconsin Breast Cancer Classic dataset of 569 data points in 2 classes (benign and malignant) described by 30 features that relate to the properties of cells visible under a microscope.

Digits [22] 1797 images of 8 by 8 pixels that contain handwritten digits (10 classes).

Gamma Telescope [23] Registration of high-energy gamma particles in a telescope. The dataset contains 19 020 samples with 11 features in two classes (signal and background).

Image Segmentation [24] In this dataset, 2306 data points fall into 7 classes and are described by 16 features that encode several attributes of image regions. We leave out three near-constant features, as suggested by Schneider et al. [7].

Ionosphere [25] Electrons in the ionosphere recorded by a high-frequency radio antenna array. The binary dataset contains 351 samples with 34 features.

[2] https://github.com/johny-c/pylmnn

[3] https://github.com/MrNuggelz/sklearn-lvq

Iris [26] Classic dataset of 150 samples in 3 classes that are three different types of the plant Iris. The 4 features are sepal and petal length and width, respectively.

Letter Recognition [27] Based on black-and-white images of capital letters (corresponding to 26 classes), this dataset consists of 20 000 samples of 16 hand-crafted features.

Outdoor Objects [28] Here, 4000 data points correspond to images that belong to one of 40 classes, depending on objects visible in the images. Its 21 features constitute normalized color histograms.

Pen Digits [29] Recognition of handwritten digits, based on readings from a stylus and a pressure-sensitive tablet. The dataset consists of 10 992 samples of 16 features in 10 classes.

Robot Navigation [30] Ultrasound sensor readings obtained by a robot during navigation. The 5456 samples are represented by 24 features and the 4 classes correspond to directional movement instructions.

USPS [31] Another dataset for handwritten digit recognition that contains 16 by 16 pixel images. The data was originally obtained in cooperation with the US Postal Service.

Wine Classic dataset with 3 classes (types of wine), 13 features, and 178 samples.

For each algorithm and dataset we perform a 10-fold, stratified, randomly shuffled cross validation and include a z-score transformation as the only preprocessing step. We report the averaged accuracies together with their standard deviations in Table 1. LANN obtains an improvement as compared to LMNN in four out of five cases, yielding a smaller variation in all cases. Interestingly, local metric learning seams particularly profitable for the outdoor objects data, a setting with a large number of classes and comparably high degree of noise.

LANN yields an indication of relevance for each feature with respect to each individual data point. We can use these to develop a *local* understanding of feature relevance. For the Wisconsin Breast Cancer dataset, we aggregate these relevances class-wise. Our findings, presented in Figure 1, align with those previously discovered and discussed in the literature [32]. In particular, it

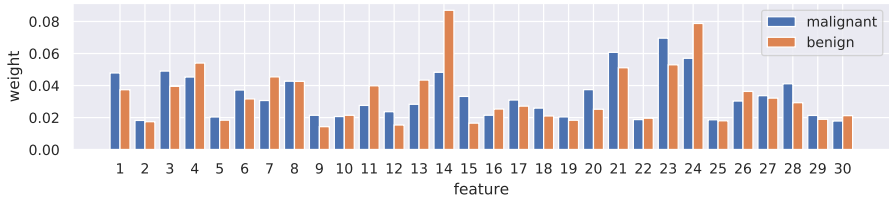


Figure 1: Aggregated per-class relevances for the Wisconsin Breast Cancer dataset, as determined by our proposed algorithm. The two different colors indicate the two classes *benign* and *malignant*.

becomes apparent that different averages result for the two classes.

5.2. Embeddings

To asses and visualize the nature of and relation between the local metrics LANN finds, we compute embeddings for two data sets: one artificial data set where we know that locality is crucial, and *Image Segmentation* (see Section 5.1) as a real-world data set. We present the results in Figures 2 and 3. The artificial data set, which we dub *Licorice*, consists of several distinct cylinders of varying orientation that in turn contain points labeled according to whether they are located inside the cylinder or on its outside.

For each data set, we first compute an embedding via UMAP using the Euclidean metric. We then train a set of algorithms on the entire data set; the global low-rank metrics learned by GMLVQ and LMNN directly lead to embeddings; LANN yields a local metric for each point, such that we can use pairwise distances as input for UMAP; and for LGMLVQ we compute a proxy embedding as described in Section 4, which is in turn embedded by UMAP.

To quantify the quality of the produced embeddings with respect to their discriminative power, we reclassify each point by its nearest neighbors in said embedding, and indicate the proportion of correctly classified points below the respective plots. Note that these numbers are not to be taken as deciding scores with regards to the efficacy of the respective algorithms; they merely aid in interpreting the validity of the embeddings. Because UMAP preserves local

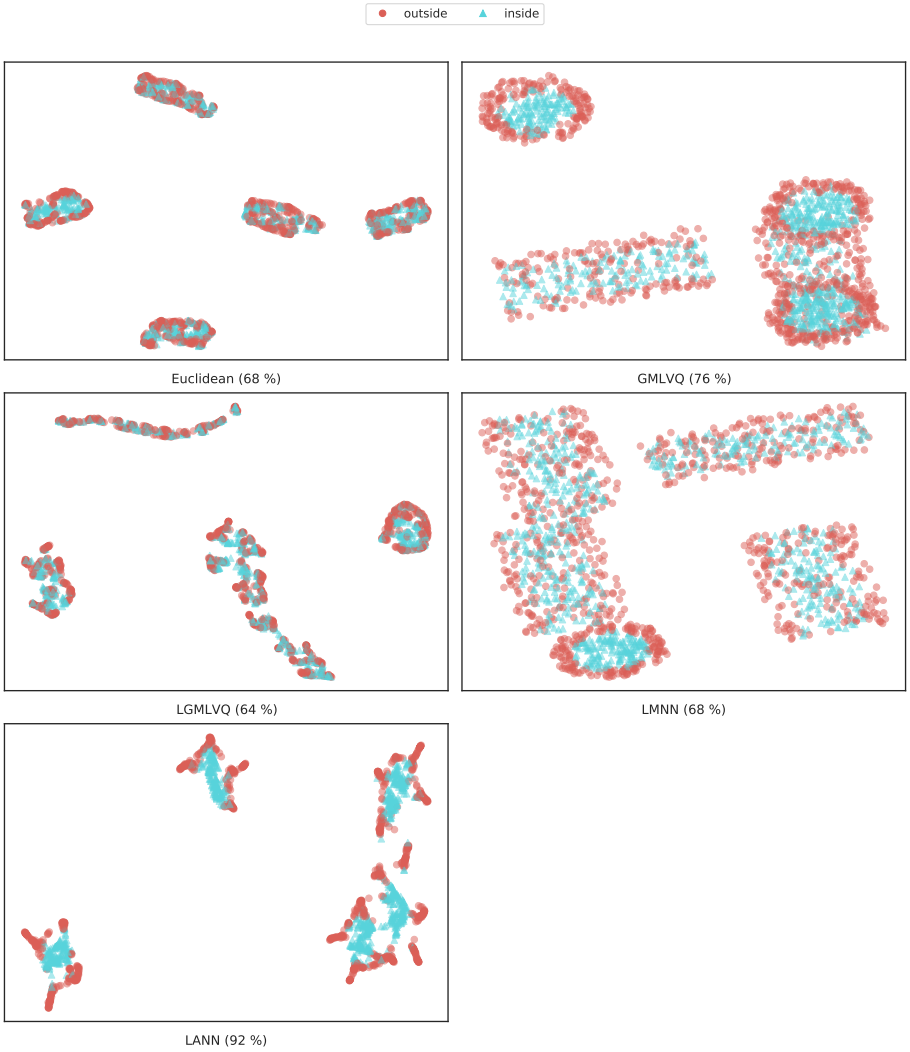


Figure 2: Embeddings of an artificial data set (Licorice), which consists of five distinct cylinders that in turn contain points labeled according to whether they are inside the cylinder or on its outside. Below each plot, we indicate the metric or the algorithm (that produces it) used to obtain the embedding, as well as the accuracy when classifying based on the embedding. Class labels are indicated by color and shape. Notably, the local metrics of LANN ensure that "inside" points are placed close to one another with "outside" points spread around them, irrespective of the underlying cylinder's orientation – at the same time, the global metrics of GMLVQ and LMNN struggle to cope with these different orientations.

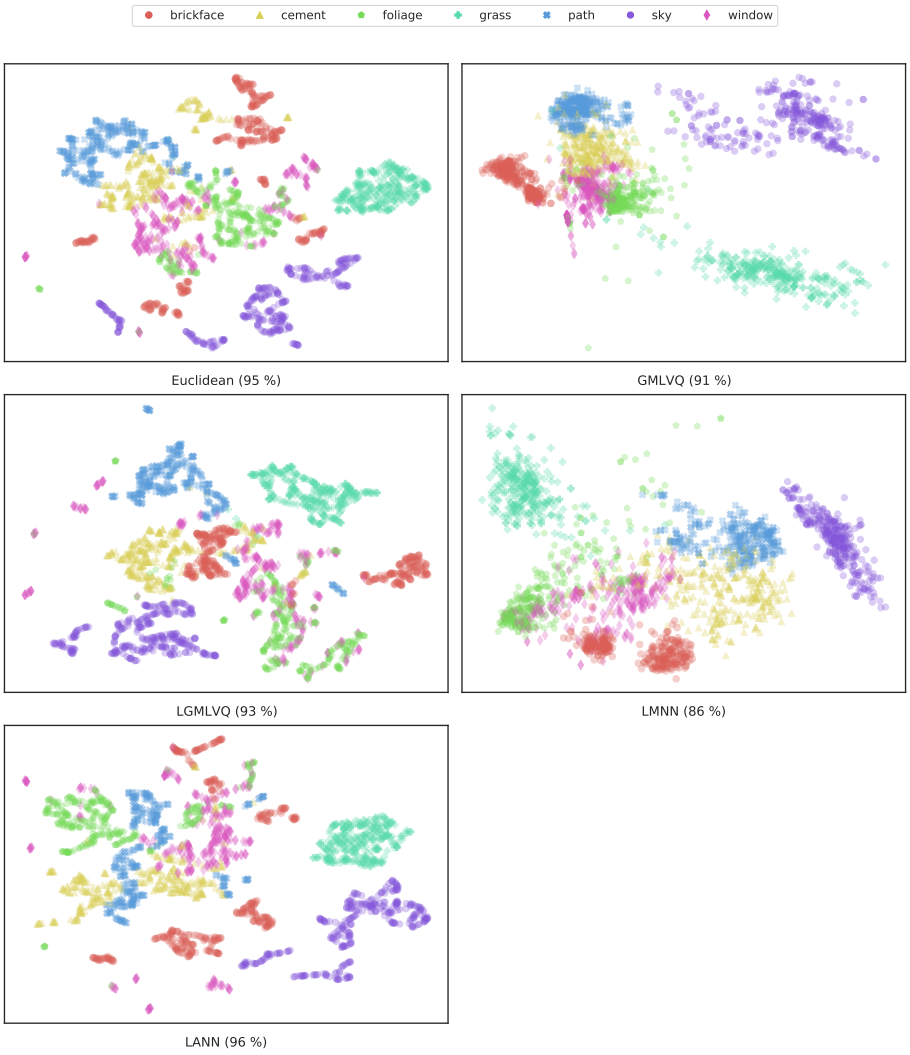


Figure 3: Embeddings of the real-world data set *Image Segmentation*. Below each plot, we indicate the metric or the algorithm (that produces it) used to obtain the embedding, as well as the accuracy when classifying based on the embedding. Class labels are indicated by color and shape.

neighborhoods, its embeddings naturally lend itself well to nearest neighbor classification.

All algorithms result in viable low-dimensional embeddings of the original data. As expected, and in line with our findings in Section 5.1, local metrics do improve the discriminative power when used for embedding. Especially in our artificial data set, where cylinders are present in different orientations, the ability to adapt to local properties is crucial. Furthermore, the results on *Image Segmentation* underline the usefulness of LANN, enabling a better distinction even between classes "foliage" and "window", which appears to be particularly challenging.

6. Conclusions

We have proposed a metric learning scheme which assigns a separate relevance weighting vector to every data point of a kNN classifier, leading to different local relevances of the decision function. Even restricted to local diagonal matrices, the technology is as good as or surpasses popular metric learning schemes such as LNMM. More importantly, the method provides a local explanation of a specific decision of the model given an input $\vec{x}$ rather than a global metric, and it enables online update rules in the form of a stochastic gradient. We have demonstrated the quality and applicability of the learned local metrics via low-dimensional embeddings obtained through state of the art dimensionality reduction.

6.1. Limitations & Future Work

It is subject to future work to integrate this scheme into kNN methods for streaming data and to investigate the suitability to build a reject option based on this representation, as investigated in [14, 33] for the standard Euclidean metric.

Due to the local metrics, common optimizations for nearest neighbor computations are not readily applicable to LANN, so we cannot currently recommend

it for big data computations. However, because during our proposed iterative updates are local, we see a number of promising directions to optimize computations.

How well explanations perform is difficult to quantify, because what constitutes a good explanation depends on concrete applications, and because ground truth is not usually available. In our setting, this issue is compounded by a lack of ground truth for local relevances. Our proposed method can aid in exploratory data analysis accompanied by interactive explanations for predictions, and we are looking forward to releasing a framework for this purpose based on LANN.

Acknowledgements

We gratefully acknowledge support by Honda Research Institute Europe.

References

[1] Bryce Goodman and Seth Flaxman. "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"". In: *AI Magazine* 38.3 (Oct. 2017), pp. 50–57. DOI: 10.1609/aimag.v38i3.2741.

[2] Teuvo Kohonen. *Self-Organizing Maps*. Berlin, Heidelberg: Springer-Verlag, 1997. ISBN: 3540620176.

[3] Maximilian Alber et al. "iNNvestigate Neural Networks!" In: *J. Mach. Learn. Res.* 20 (2019), 93:1–93:8.

[4] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 97–101. DOI: 10.18653/v1/N16-3020.

[5] Barbara Hammer and Thomas Villmann. "Generalized relevance learning vector quantization". In: *Neural networks : the official journal of the International Neural Network Society* 15 8-9 (2002), pp. 1059–68.

[6] Andreas C. Neocleous et al. “Marker selection for the detection of trisomy 21 using generalized matrix learning vector quantization”. In: *International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 3704–3708. DOI: 10.1109/IJCNN.2017.7966322.

[7] Petra Schneider, Michael Biehl, and Barbara Hammer. “Adaptive Relevance Matrices in Learning Vector Quantization”. In: *Neural Computation* 21.12 (2009), pp. 3532–3561. DOI: 10.1162/neco.2009.11-08-908.

[8] Aurélien Bellet, Amaury Habrard, and Marc Sebban. *A Survey on Metric Learning for Feature Vectors and Structured Data*. 2013. arXiv: 1306.6709.

[9] Kilian Q. Weinberger and Lawrence K. Saul. “Distance Metric Learning for Large Margin Nearest Neighbor Classification”. In: *J. Mach. Learn. Res.* 10 (2009), pp. 207–244.

[10] Barbara Hammer, Frank-Michael Schleif, and T. Villmann. “On the Generalization Ability of Prototype-Based Classifiers with Local Relevance Determination”. In: 2005.

[11] Kilian Q. Weinberger and Lawrence K. Saul. “Fast Solvers and Efficient Implementations for Distance Metric Learning”. In: *ICML 2008*. 2008.

[12] Jun Wang, Alexandros Kalousis, and Adam Woznica. “Parametric Local Metric Learning for Nearest Neighbor Classification”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1601–1609.

[13] Yung-kyun Noh, Byoung-tak Zhang, and Daniel D. Lee. “Generative Local Metric Learning for Nearest Neighbor Classification”. In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 1822–1830.

[14] Viktor Losing, Barbara Hammer, and Heiko Wersing. "Self-Adjusting Memory: How to Deal with Diverse Drift Types". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 4899–4903. DOI: 10.24963/ijcai.2017/690.

[15] Jan Philip Göpfert, Heiko Wersing, and Barbara Hammer. "Locally Adaptive Nearest Neighbors". In: *European Symposium on Artificial Neural Networks*. 2020.

[16] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[17] John W Sammon. "A nonlinear mapping for data structure analysis". In: *IEEE Transactions on computers* 100.5 (1969), pp. 401–409.

[18] Leland McInnes, John Healy, and James Melville. *Umap: Uniform manifold approximation and projection for dimension reduction*. 2018. arXiv: 1802.03426.

[19] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[20] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[21] M. Biehl et al. "Matrix relevance LVQ in steroid metabolomics based classification of adrenal tumors". In: *20th European Symposium on Artificial Neural Networks (ESANN)*. 2012, pp. 423–428.

[22] Ethem Alpaydin and Cenk Kaynak. "Cascading classifiers". In: *Kybernetika* 34.4 (1998), pp. 369–374.

[23] Dieter Heck et al. "CORSIKA: A Monte Carlo code to simulate extensive air showers". In: *Report fzka* 6019.11 (1998).

[24] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017.

[25] Vincent G Sigillito, Simon P Wing, Larrie V Hutton, and Kile B Baker. "Classification of radar returns from the ionosphere using neural networks". In: *Johns Hopkins APL Technical Digest* 10.3 (1989), pp. 262–266.

[26] Ronald A Fisher. "The use of multiple measurements in taxonomic problems". In: *Annals of eugenics* 7.2 (1936), pp. 179–188.

[27] Peter W Frey and David J Slate. "Letter recognition using Holland-style adaptive classifiers". In: *Machine learning* 6.2 (1991), pp. 161–182.

[28] Viktor Losing, Barbara Hammer, and Heiko Wersing. "KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift". In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Barcelona: IEEE, 2016, pp. 291–300. ISBN: 978-1-5090-5473-2. DOI: 10.1109/ICDM.2016.0040.

[29] Fevzi Alimoglu and Ethem Alpaydin. "Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition". In: *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96*. Citeseer. 1996.

[30] Ananda L Freire, Guilherme A Barreto, Marcus Veloso, and Antonio T Varela. "Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study". In: *2009 6th Latin American Robotics Symposium (LARS 2009)*. IEEE. 2009, pp. 1–6.

[31] Jonathan J. Hull. "A database for handwritten text recognition research". In: *IEEE Transactions on pattern analysis and machine intelligence* 16.5 (1994), pp. 550–554.

[32] Christina Göpfert, Lukas Pfannschmidt, Jan Philip Göpfert, and Barbara Hammer. "Interpretation of Linear Classifiers by Means of Feature Relevance Bounds". In: *Neurocomputing* 298 (2018), pp. 69–79. ISSN: 1872-8286. DOI: 10.1016/j.neucom.2017.11.074.

[33] Jan Philip Göpfert, Barbara Hammer, and Heiko Wersing. "Mitigating Concept Drift via Rejection". In: *Lecture Notes in Computer Science* (2018). DOI: 10.1007/978-3-030-01418-6_45.