A Variable Kernel Function for Hybrid Unsupervised Kernel Regression

Daniel Lückehe
Department of Geoinformation
Jade University of Applied Sciences
Oldenburg, Germany
daniel.lueckehe@uni-oldenburg.de

Oliver Kramer
Department of Computing Science
University of Oldenburg
Oldenburg, Germany
oliver.kramer@uni-oldenburg.de

ABSTRACT

Dimensionality reduction is an important problem class in machine learning and data mining, as the dimensionality of data sets is steadily increasing. This work is a contribution in the line of research on iterative unsupervised kernel regression (UKR), a class of methods for dimensionality reduction that employ regression methods to find low-dimensional representations of high-dimensional patterns. We introduce a hybrid optimization approach of iteratively constructing a solution and performing gradient descent in the data space reconstruction error (DSRE). Further, we introduce a variable kernel function that increases the flexibility of UKR learning. The variable kernel function increases the model capacity, but introduces new parameters that have to be tuned.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models

Keywords

Dimensionality reduction, optimization, kernel function

1. HYBRID UNSUPERVISED REGRESSION

The problem of dimensionality reduction is to find low-dimensional representations $\mathbf{x}_i \in \mathbb{R}^q$ of high-dimensional patterns $\mathbf{y}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$. The method, we focus on in this paper is UKR, where a regression model f is used to map from the low-dimensional space to the given high-dimensional patterns. In the optimal case, for every pattern it should hold $f(\mathbf{x}_i) = \mathbf{y}_i$, which means that the high-dimensional patterns are perfectly reconstructed with the low-dimensional representation and regression model $f(\cdot)$. With real-world data sets this relation is difficult to achieve. The difference between $f(\mathbf{x}_i)$ and $\mathbf{y}_i$ can be defined as $r(\mathbf{x}_i) = \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2$, which is the DSRE. For a matrix $\mathbf{X} = [\mathbf{x}_i]_{i=1}^N$ of patterns $\mathbf{x}_i$ with $i = 1, \ldots, N$ the DSRE of a the whole manifold is $R(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^N r(\mathbf{x}_i)$.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598459.

As regression model f, we use the Nadaraya-Watson estimator [4]

$$f(\mathbf{X}) = \sum_{i=1}^{N} \mathbf{y}_i \cdot \frac{\mathbf{K}_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{j=1}^{N} \mathbf{K}_h(\mathbf{x} - \mathbf{x}_j)} \tag{1}$$

with kernel function $\mathbf{K}_h$ that usually has a kernel parameter h, e.g., the bandwidth of the kernel. In Nadaraya-Watson estimation, $\mathbf{x}$ itself affects the result of the model. Hence, leave-one-out crossvalidation is employed for the DSRE computation to avoid overfitting. Used kernel functions in this work are the Epanechnikov kernel, which has an optimal mean integrated squared error and the quartic kernel, which is differentiable.

The idea of the hybrid optimization approach is to combine gradient descent optimization for UKR introduced by Klanke and Ritter [2] and iterative solution construction for UNN presented by Kramer [3] to a hybrid iterative dimensionality reduction method (hybUKR). To tune the bandwidth of used kernel function, we use an evolution strategy (ES) [1] for the iterative method. The $(\mu + \lambda)$-ES performs the iterative method multiple times. In the iterative embedding step, patterns are successively embedded, i.e., for pattern $\mathbf{y}_i$ latent candidate positions are generated with Gaussian sampling. The first pattern $\mathbf{y}_1$ is embedded at an arbitrary position, e.g. at $\overline{\mathbf{X}} = [\mathbf{0}]$ and $\overline{\mathbf{Y}} = [\mathbf{y}_1]$. Let $\overline{\mathbf{Y}}$ be the matrix of embedded patterns in iteration i and let $\overline{\mathbf{X}}$ be the corresponding latent positions. For each pattern $\mathbf{y}_i$ with $i > 1$, κ latent candidate positions are sampled in latent space with Gaussian sampling $\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{x}_j, \|\mathbf{y}_i - \mathbf{y}_j\|_2^2\right)$ with $\mathbf{x}_j = [\overline{\mathbf{X}}]_j, \mathbf{y}_j = [\overline{\mathbf{Y}}]_j$ and

$$j = \arg \min_{\mathbf{y}_j = [\overline{\mathbf{Y}}]_j} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 . \tag{2}$$

The candidate position that minimizes the DSRE is finally chosen for the new manifold $\overline{\mathbf{X}}$. Alternatively, the position can be evolved with a $(1+\lambda)$-ES. After η patterns have been embedded, gradient descent in the space of latent variables w. r. t. the DSRE is performed:

$$\mathbf{X}_{\text{mod}} = \mathbf{X} - \alpha \cdot \nabla L(f(\mathbf{X}), \mathbf{Y}) \tag{3}$$

The optimized latent space is $\mathbf{X}_{\text{mod}}$. Stepwidth α is an important parameter and is optimized with an ES in this work. Gradient descent is repeated until no improvement of the DSRE is achieved. In the iterative scheme, special cases are $\eta = 1$, i.e., gradient descent is performed after each single embedding, or $\eta = N$, i.e., gradient descent is only called once after all patterns have been embedded. In the case $\eta = N$, we observed improvements in comparison to initial-

ization with other methods, as the gradient descent process uses the same regression method as the iterative process for creating the low-dimensional space. And it turns out that the learning embeddings are better w. r. t. the DSRE, if the gradient descent process is performed multiple times during the iterative process, i.e., for $\eta < N/2$.

2. VARIABLE KERNEL FUNCTION

To increase flexibility of our approach, we introduce a variable kernel function in this section. Not only the kernels bandwidth parameter is optimized, but also further parameters that determine the characteristics of the variable kernel function.

To combine multiple kernel functions, the following two conditions must hold at the transition of function K_1 to function K_2 at position x:

$$K_1(x) \stackrel{!}{=} K_2(x) \text{ and } K_1'(x) \stackrel{!}{=} K_2'(x) \tag{4}$$

In general, a kernel function has three interesting properties: (1) a maximum that yields $K(\cdot) = \beta$ in case of maximal similarity of two patterns, (2) a decreasing part starting at a defined boundary value (bv) with a defined curvature (e.g. varying slope) for patterns with decreasing similarity, and (3) a tail that determines the weak influence of dissimilar patterns. This leads to the following function definition for our variable kernel function $\hat{K}$ that allows to parameterize properties (1) and (2), while condition (3) equals the properties of the Epanechikov kernel:

$$\hat{K} = \beta \cdot \begin{cases} 1 & \text{if } |x| \leq \xi_0 \\ -(x - x_{bv})^2 + 1 & \text{if } |x| > \xi_0 \text{ and } |x| \leq \xi_1 \\ -m \cdot (x - x_{bv}) + \frac{m^2}{4} + 1 & \text{if } |x| > \xi_1 \text{ and } |x| \leq \xi_2 \\ (x - (\frac{1}{m} + \frac{m}{2} + x_{bv}))^2 & \text{if } |x| > \xi_2 \text{ and } |x| \leq \xi_3 \\ 0 & \text{if } |x| > \xi_3 \end{cases} \tag{5}$$

and

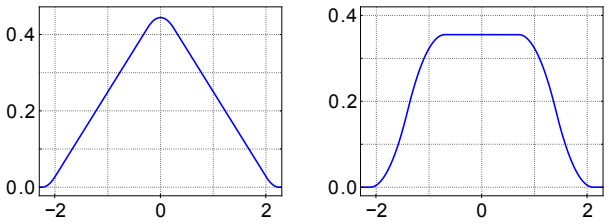
$$\frac{\partial \hat{K}}{\partial x} = \beta \cdot \begin{cases} 0 & \text{if } |x| \leq \xi_0 \\ -2 \cdot (x - x_{bv}) & \text{if } |x| > \xi_0 \text{ and } |x| \leq \xi_1 \\ -m & \text{if } |x| > \xi_1 \text{ and } |x| \leq \xi_2 \\ 2 \cdot (x - (\frac{1}{m} + \frac{m}{2} + x_{bv})) & \text{if } |x| > \xi_2 \text{ and } |x| \leq \xi_3 \\ 0 & \text{if } |x| > \xi_3 \end{cases} \tag{6}$$

with $\xi_0 = x_{bv}, \xi_1 = \frac{m}{2} + x_{bv}, \xi_2 = \frac{1}{m} + x_{bv}$, and $\xi_3 = \frac{1}{m} + \frac{m}{2} + x_{bv}$. For gradient m of the function, it must hold $0 < m \leq \sqrt{2}$, and $x_{bv} \geq 0$ to assure the necessary conditions. To ensure that the area under $\hat{K}$ is one, we derived the following condition:

$$\beta = \frac{2 \cdot m}{4 \cdot m \cdot x_{bv} + m^2 + 2} \tag{7}$$

This function definition allows shapes with flexible curvature. Figure 1 shows two examples, $\hat{K}_1$, a shape with sharp top with settings, and $\hat{K}_2$, a shape with large plateau.

The variable kernel function is employed in the hybUKR embedding process in the following experiments. The ES of hybUKR optimizes the kernel parameters h, x_{bv}, and m. The parameters are initialized as follows: $x_{bv} = 0.2$ and $m = 0.7$. We use the *Digits* data set with '1' and '3', $N = 100$ and a $q = 2$-dimensional latent space. The optimization is based on a $(2+7)$-ES for hybUKR. This turned out to be a good choice in previous experiments. Each experiment is repeated 25 times. We employ Rechenberg's success rule to control mutation strengths. Table 1 shows the outcome of the experiments. We can observe that the



(a) $\hat{K}_1$ with sharp top (b) $\hat{K}_2$ with plateau

Figure 1: Visualization of two variable kernel function parameterizations: (1) $\hat{K}_1$ with $x_{bv} = 0.0$ and $m = 0.5$, (2) $\hat{K}_2$ with $x_{bv} = 0.7$ and $m = 1.4$.

kernel	gradient descent	DSRE
quartic	no	1706 ± 22
	at $n = 100$	1540 ± 16
	at $n = 25, 50, 75, 100$	1518 ± 12
$\hat{K}$	no	1712 ± 26
	at $n = 100$	1535 ± 16
	at $n = 25, 50, 75, 100$	1515 ± 15

Table 1: Comparison of quartic kernel and variable kernel $\hat{K}$ in the hybUKR embedding process w. r. t. DSRE.

gradient descent process improves the experimental results and reduces the standard deviations. The results also show that an alternating scheme is advantageous to reduce the DSRE. The novel variable kernel $\hat{K}$ achieves similar results as the quartic kernel, only a slight, not significant improvement can be observed when combined with gradient descent.

3. SUMMARY AND CONCLUSIONS

In this work, we employed a hybrid optimization approach that combines iterative embeddings with gradient descent. For this sake, we had to employ differentiable Nadaraya-Watson estimator with differentiable kernel functions. To increase the flexibility of adapting latent variables in the low-dimensional space, a variable kernel function has been introduced. It motivates the employment of evolutionary search to adapt to certain data space characteristics. Although the superiority of the increased model capacity of the variable kernel function was only marginal on the *Digits* data set, we expect improvements on other more complex data spaces with potentially varying local data space characteristics.

4. REFERENCES

[1] H.-G. Beyer and H.-P. Schwefel. Evolution strategies - A comprehensive introduction. *Natural Computing*, 1:3–52, 2002.

[2] S. Klanke and H. Ritter. Variants of unsupervised kernel regression: General cost functions. *Neurocomputing*, 70(7-9):1289–1303, 2007.

[3] O. Kramer. Dimensionality reduction by unsupervised k-nearest neighbor regression. In *ICMLA (1)*, pages 275–278, 2011.

[4] E. Nadaraya. On estimating regression. *Theory of Probability and Its Application*, 10:186–190, 1964.