Minimum Modal Regression

Koichiro Yamauchi[1] and Vanamala Narasimha Bhargav[2]
[1]*Department of Computer Science, Chubu University, Matsumoto-cho 1200 Kasugai, Japan*
[2]*Indian Institute of Technology, Guwahati, Assam, India*

Keywords: Modal Regression, Kernel Distribution Estimator, Incremental Learning on a Budget, Kernel Machines, Projection Method.

Abstract: The recent development of microcomputers enables the execution of complex software in small embedded systems. Artificial intelligence is one form of software to be embedded into such devices. However, almost all embedded systems still have restricted storage space. One of the authors has already proposed an incremental learning method for regression, which works under a fixed storage space; however, this method cannot support the multivalued functions that usually appear in real-world problems. One way to support the multivalued function is to use the model regression method with a kernel density estimator. However, this method assumes that all sample points are recorded as kernel centroids, which is not suitable for small embedded systems. In this paper, we propose a minimum modal regression method that reduces the number of kernels using a projection method. The conditions required to maintain accuracy are derived through theoretical analysis. The experimental results show that our method reduces the number of kernels while maintaining a specified level of accuracy.

1 INTRODUCTION

The recent development of microcomputers enables the embedding of complex software into small devices. Machine learning algorithms are one example of such software. One of the authors has previously proposed a learning algorithm for kernel regression in embedded systems (Yamauchi, 2014), but this general regression method estimates the conditional expectation of the dependent variable (***Y***) given the independent variables (***X***=x). In contrast, modal regression (Einbeck et al, 2006) estimates the conditional modes of ***Y*** given ***X***=x. This strategy enables the learning machine to predict a portion of the missing variables from the other known variables according to the given sample distribution. This property is quite different from that of other typical regression methods.

To estimate the conditional modes, partial mean shift (PMS) is an assured method. At first, the PMS method attempts to obtain the joint kernel density and derives it using the gradient ascent. However, if the number of samples is increasing, minimum modal regression is proposed, which can estimate the joint kernel density by projecting the new sample, replacing the old kernel, or adding the new kernel to the sample. The equation for PMS is then modified accordingly.

2 MODAL REGRESSION

Modal regression approximates a multivalued function to search the local peaks of a given sample distribution. Modal regression consists of the kernel density estimator with a PMS method.

2.1 Kernel Density Estimator

The kernel density estimator (KDE) is a variation of the Parzen window (Parzen, 1962).
Let χ be the set of learning samples, and $\chi = \left\{ \mathbf{x}_p \in \Re^n \middle| p = 1,2,\cdots,N \right\}$. The estimator approximates the probability density function by using a number of kernels, namely, the support set S_t. The kernels used are Gaussian kernels, and

$$p(\mathbf{x}) \propto \sum_{i \in S_t} K\left(\frac{\left\| \mathbf{x} - \mathbf{x}_i \right\|}{h_x} \right), \qquad (1)$$

Yamauchi, K. and Bhargav, V.
Minimum Modal Regression.
DOI: 10.5220/0006601304480455
In *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2018)*, pages 448-455
ISBN: 978-989-758-276-9

where

$$K\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{h_x}\right) \equiv \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{h_x^2}\right). \quad (2)$$

Normally, the same number of kernels as that of the dataset is required. However, if the storage capacity of a target device is small, the number of kernels must be restricted. There are several ways to realize the density estimation using a limited number of kernels. Traditionally, self-organizing feature maps or learning vector quantization methods approximate the distribution by using a fixed number of templates.

As mentioned in (Sasaki et al., 2016), the KDE used in modal regression should approximate the peak points of the distribution, rather than the distribution itself. Let $\hat{p}(\mathbf{x})$ be

$$\hat{p}(\mathbf{x}) \equiv \sum_{i \in S_t} K\left(\frac{\|\mathbf{x}-\mathbf{x}_i\|}{h_x}\right), \quad (3)$$

then $\hat{p}(\mathbf{x})$ should satisfy the following condition.

$$\begin{aligned} &\nabla_{\mathbf{x}} \hat{p}(\mathbf{x})\big|_{x=x^*} = \nabla_{\mathbf{x}} p(\mathbf{x})\big|_{x=x^*} = 0 \\ &\nabla_{\mathbf{x}}^2 \hat{p}(\mathbf{x})\big|_{x=x^*} < 0, \quad \nabla_{\mathbf{x}}^2 p(\mathbf{x})\big|_{x=x^*} < 0 \end{aligned}, \quad (4)$$

where $\mathbf{x}^*$ denotes a local peak point of the distribution.

2.2 Partial Mean Shift

Modal regression searches the peaks of the distribution model represented by the KDE. The PMS method realizes quick convergence to the nearest peak from the initial point. Let us denote the initial point as $\mathbf{x}_0$, representing the starting point for the search of the peak points. Thus, modal regression repeats the modification of the current y as follows:

$$y_{new} \leftarrow \frac{\sum_i y_{old} K\left(\frac{|y_{old}-y_i|}{h_y}\right) K\left(\frac{\|\mathbf{X}-\mathbf{X}_i\|}{h_x}\right)}{\sum_j K\left(\frac{|y_{old}-y_j|}{h_y}\right) K\left(\frac{\|\mathbf{X}-\mathbf{X}_j\|}{h_x}\right)}, \quad (5)$$

where $\mathbf{X}$ denotes $\mathbf{X} = \begin{bmatrix} x_1 & \cdots & x_N, y \end{bmatrix}^T$. Note that $\mathbf{X}$ includes y.

3 MINIMUM MODAL REGRESSION

To realize the minimum modal regression, a minimum KDE, which realizes the KDE with a minimum support set, is proposed. Moreover, the KDE should support incremental learning during its service. To this end, we modify an online learning method for kernel perceptrons on a budget and apply the modified method for online learning of the KDE.

The existing kernel perceptron on a budget maintains a minimized or a constant support set by applying projection and pruning with replacement. In this study, we derivate some conditions to make an online learning algorithm for the KDE in order to be used in the modal regression.

In the following section, we use the following relationship to represent the pruning with replacement and a projection of kernels. Therefore, we choose Gaussian kernel for $K(\cdot)$, which is a kind of reproducing kernel. Thus, we have following relationship, referred to as the kernel trick:

$$K\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{h_x}\right) = \left\langle k(\mathbf{x}_j,\cdot), k(\mathbf{x},\cdot)\right\rangle, \quad (6)$$

where $\langle\cdot,\cdot\rangle$ denotes the dot product.

3.1 Minimum KDE

The KDE for modal regression should represent the peak points of the distribution within a certain number of kernels. Therefore, the modal regression finds the $\mathbf{X}_{MP} = \begin{bmatrix} \mathbf{x}_{MP}^T & y_{MP} \end{bmatrix}^T$ which satisfies the following two conditions:

$$\begin{cases} \nabla_x \hat{p}(\mathbf{X})\big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0 \\ \nabla_x^2 \hat{p}(\mathbf{X})\big|_{\mathbf{x}=\mathbf{x}_{MP}} < 0 \end{cases}, \quad (7)$$

where $\hat{p}(\mathbf{X})$ is defined in (3). Next, we describe $\hat{p}(\mathbf{X})$ as a dot product of the corresponding vector in Hilbert space and the input: $\hat{p}(\mathbf{X}) = \langle \hat{p}, k(\mathbf{X},\cdot)\rangle$. As $\hat{p}(\mathbf{X})$ is described by a linear combination of several Gaussian kernels, which is one of the reproducing kernels, we can apply the kernel trick to calculate it. Thus, the KDE is also described by using the kernel method. Therefore, the learning method of the KDE is described as follows. Let us assume that

the KDE used in this study tends to realize a sparse allocation of kernels. Therefore, the KDE normally adds a new kernel when a new sample $(\mathbf{x}_t, y_t)$ is presented. Therefore,

$$\hat{p}_t = \hat{p}_{t-1} + w_t k(\mathbf{X}_t,\cdot),\ S_t = S_{t-1} \cup \{t\}, \quad (8)$$

where S_t denotes the support set at the t th round, $w_t = 1$, and $\hat{p}_t$ is

$$\hat{p}_t = \sum_j w_j k(\mathbf{X}_j,\cdot), \quad (9)$$

where w_j is the extension coefficient for each kernel, whose default value is 1 and $w_j \geq 0$. The KDE is not for regression, so (9) does not contain y_t. Instead, y_t is one element of the centroid of a kernel. Equation (8) represents the same procedure as that of the original kernel distribution estimator. This strategy, however, continues to increment the size of the support set $|S_t|$ forever if the number of datasets is infinite. This is not suitable for an environment in which storage space is limited. S_t should only contain some essential kernels to represent the distribution of inputs.

To maintain a small value of $|S_t|$, we apply an improved version of the kernel perceptron on a budget (Orabona et al., 2008) (He et al., 2012) (Yamauchi, 2013). If we apply their method to the KDE, the KDE attempts to apply the projection or replacement operation instead of appending a new kernel. Therefore, if a condition explained in the latter section is satisfied, the KDE applies the replacement or projection operation. The replacement operation is

$$\hat{p}_t = \hat{p}_{t-1} - w_{i^*} k(\mathbf{X}_{i^*},\cdot) + w_i P_{t-1-i^*} k(\mathbf{X}_{i^*},\cdot) + k(\mathbf{X}_t,\cdot). \quad (10)$$

On the other hand, the projection operation is

$$\hat{p}_t = \hat{p}_{t-1} + P_{t-1^*} k(\mathbf{X}_t,\cdot), \quad (11)$$

where $P_{t-1-i^*} k(\mathbf{X}_{i^*},\cdot)$ denotes the projected vector of the i^* th kernel to the space spanned by the remaining kernels. The projected vector $P_{t-1-i^*} k(\mathbf{X}_{i^*},\cdot)$ is

$$P_{t-1-i^*} k(\mathbf{X}_{i^*},\cdot) = \sum_{j \in S_t \setminus i^*} a_{i^* j} k(\mathbf{X}_j,\cdot). \quad (12)$$

This means that the KDE removes the most ineffective i^* th kernel after projecting the kernel to the space spanned by the remaining kernels. The most ineffective kernel is detected by estimating the approximated linear dependency.

$$i^* = \arg\min_i \{\delta_i\}, \quad (13)$$

where

$$\delta_i = \min_{a_i} \left\| k(\mathbf{X}_i,\cdot) - \sum_{j \in S_t \setminus i} a_{ij} k(\mathbf{X}_j,\cdot) \right\|^2. \quad (14)$$

The following two theorems derivate the condition to maintain the $\mathbf{X}'_{MP}$ s of the peak points, even after the replacement or projection operations.

Theorem 1

Let i^* be the most ineffective kernel in S_{t-1}, which is determined by (13). Let $\hat{p}'_t$ be

$$\hat{p}'_t = \hat{p}_{t-1} - w_i \{k(\mathbf{X}_{i^*},\cdot) - P_{t-1} k(\mathbf{X}_{i^*},\cdot)\}.$$

Let $\mathbf{x}_{MP}$ be the point that satisfies

$$\begin{cases} \nabla_X \langle \hat{p}_{t-1}, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0 \\ \nabla^2_X \langle \hat{p}_{t-1}, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} < 0 \end{cases}.$$

When $\|\delta_{i^*}\|^2 = 0$, we have

$$\begin{cases} \nabla_x \langle \hat{p}'_t, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0 \\ \nabla^2_x \langle \hat{p}'_t, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} < 0 \end{cases}.$$

Theorem 2

Let $\mathbf{X}_t$ be a new input at the t th round, and $P_{t-1} k(\mathbf{x}_t,\cdot)$ be the projected vector of $k(\mathbf{X}_t,\cdot)$ to the space spanned by the kernels at round $t-1$. Let $\hat{p}'_t$ be

$$\hat{p}'_t = \hat{p}_{t-1} + P_{t-1} k(\mathbf{X}_t,\cdot).$$

Let $\mathbf{x}_{MP}$ be a point that satisfies the following condition.

$$\begin{cases} \nabla_x \langle \hat{p}_{t-1}, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0 \\ \nabla^2_x \langle \hat{p}_{t-1}, k(\mathbf{X},\cdot) \rangle \big|_{\mathbf{x}=\mathbf{x}_{MP}} < 0 \end{cases}$$

When $\|\delta_t\|^2 = 0$, we have

$$\begin{cases} \nabla_x \left\langle \hat{p}_t', k(\mathbf{X},\cdot) \right\rangle \Big|_{\mathbf{x}=\mathbf{x}_{MP}} &= \nabla_x K\left(\frac{\|\mathbf{X}_{MP} - \mathbf{X}\|}{h_x} \right) \\ \nabla_x^2 \left\langle \hat{p}_t', k(\mathbf{X},\cdot) \right\rangle \Big|_{\mathbf{x}=\mathbf{x}_{MP}} &= \nabla_x^2 \hat{p}_{t-1} + \nabla_x^2 K\left(\frac{\|\mathbf{X}_{MP} - \mathbf{X}\|}{h_x} \right) \end{cases}$$

.

The proofs for the Theorems 1 and 2 are described in the appendix.

Theorem 2 demonstrates that if $\mathbf{X}_{MP}$ is far from $\mathbf{X}_t$,

$$\nabla_x \left\langle \hat{p}_t', k(\mathbf{X},\cdot) \right\rangle \Big|_{\mathbf{x}=\mathbf{x}_{MP}} = \nabla_x K\left(\frac{\|\mathbf{X}_{MP} - \mathbf{X}\|}{h_x} \right) \cong 0 \quad (15)$$

$$\begin{aligned} &\nabla_x^2 \left\langle \hat{p}_t', k(\mathbf{X},\cdot) \right\rangle \Big|_{\mathbf{x}=\mathbf{x}_{MP}} \\ &\quad = \nabla_x^2 \hat{p}_{t-1} + \nabla_x^2 K\left(\frac{\|\mathbf{X}_{MP} - \mathbf{X}\|}{h_x} \right) \\ &\quad \cong \nabla_x^2 \hat{p}_{t-1} < 0 \end{aligned} \quad (16)$$

From these theorems, the minimum KDE can be described in Algorithm 1.

Algorithm 1: Learning algorithm for the Minimum KDE.

Receive $(\mathbf{X}_t, y_t)$

Detect the most ineffective kernel i^* by using (13) (the lightweight version is (19)).

If $\|\delta_t\|^2 < \varepsilon$

$\hat{p}_t = \hat{p}_{t-1} + P_{t-1} k(\mathbf{X}_t,\cdot), \quad S_t = S_{t-1}$

else if $\|\delta_{i^*}\|^2 < \varepsilon$

$\hat{p}_t = \hat{p}_{t-1} - w_i \left\{ k(\mathbf{X}_{i^*},\cdot) - P_{t-1} k(\mathbf{X}_{i^*},\cdot) \right\}$

$S_t = \{S_{t-1} \setminus \{i\}\} \cup \{t\}$

else

$\hat{p}_t = \hat{p}_{t-1} + k(\mathbf{X}_t,\cdot), \quad S_t = S_{t-1} \cup \{t\}$

Endif

For all i

if $w_i < 0$ // To maintain $w_i \geq 0$

$w_i = 0$

endif

endfor

$t = t + 1$

Return $\hat{p}_t$

3.2 Modified Partial Mean Shift

The minimum KED described in the previous section maintains the minimum size of the support set by applying a projection or pruning with a replacement. Through these processes, the expansion parameter of each kernel w_i has a certain value to represent the target distribution. For example, if $w_i = 2$, the *i*th kernel shares the duty of two kernels. Therefore, we have also improved the PMS method to adjust the solution according to the expansion parameters, as follows.

$$y_{new} \leftarrow \frac{\sum_i y_{old} w_i K\left(\frac{|y_{old} - y_i|}{h_y} \right) K\left(\frac{\|\mathbf{X} - \mathbf{X}_i\|}{h_x} \right)}{\sum_j w_j K\left(\frac{|y_{old} - y_j|}{h_y} \right) K\left(\frac{\|\mathbf{X} - \mathbf{X}_j\|}{h_x} \right)} \quad (17)$$

3.3 Lightweight Learning Algorithm

In Section 3.1, we have already presented the minimum KDE. The algorithm includes the calculation of the approximated linear dependency (ALD) to detect the most ineffective kernel, which has a wasteful computational cost of $O(|S_t|^3)$. The computational cost is too large to execute the minimum KDE. To overcome this difficulty, we need a lightweight version of the minimum KDE.

The lightweight KDE does not use (13) to detect the most ineffective kernel. Instead, the proposed algorithm uses a slightly improved version of a lightweight algorithm from our previous study (Yamauchi, 2014). Therefore, the proposed method chooses the most ineffective kernel, which has the largest value, defined as

$$V_j = \sum_{i \in S_t \setminus j} K\left(\frac{\|\mathbf{X}_j - \mathbf{X}_i\|}{h_x} \right). \quad (18)$$

Note that if the kernel is located in the neighborhood of other kernels, V_j becomes large. There is a high possibility that such a kernel can be represented by a linear combination of the other kernels. Therefore, instead of applying (13), (19) is used.

$$i^* = \arg\max_j V_j \quad (19)$$

Algorithm 2: Minimum modal regression.

If a new learning sample $\mathbf{X}_t = [\mathbf{x}_t \quad y_t]^T$ is given,

Learn the minimum KDE by **Algorithm 1**

endif

If a new query $\mathbf{x}_p$ is given,

For (i=0; i<M; i++)

Select one of a kernel index $k \in \mathbf{N}(\mathbf{x}_p)$ (see (20)) randomly.

set the initial y as $y = X_{k\,N}$.

Set initial $\mathbf{X}$ as $\mathbf{X} = [\mathbf{x}_p^T \quad y]$.

For (r=0; r<R; r++)

Update y by using (17)

Reset $\mathbf{X}$ as $\mathbf{X} = [\mathbf{x}_p^T \quad y]$

endfor

$Ans \leftarrow Ans \cup \{y\}$

endfor

endif

return Ans.

where $\mathbf{N}(\mathbf{x}_p)$ denotes a set of kernels defined below equation.

$$\mathrm{N}(\mathbf{x}_p) = \left\{ j \;\middle|\; K\left(\frac{\|\mathbf{x}_j - \mathbf{x}_n\|}{h_x} \right) > s \right\}, \qquad (20)$$

where s denotes a threshold and we set $s = 0.1$.

4 EXPERIMENT

In this section, some preliminary results of the proposed method are shown.

4.1 Performance for Synthetic Dataset

We tested the proposed method with two synthetic datasets and evaluated its performance.

4.1.1 Third-Order Function

The first dataset is generated by

$$x = y^3 - 4y + n,$$

where n is a uniform random value in the interval of $[-1,1]$. With a changing y in the interval [-3, 3], 8000 datasets were generated. The dataset was presented to the minimum KDE, and the minimal modal regression predicted the values for you from the value of each x. The number of repeats for the prediction (the parameter R in **Algorithm 2**) was 10. The hyper parameters used were $h_x = 0.25$ and $h_y = 0.25$. The evaluation should be made using the mean square error between the desired and predicted values of y.

However, the evaluation of multi-valued output is complex, so we evaluated the proposed method as follows. Instead of making a direct comparison of the resultant and predicted values of y, we calculated the corresponding $\hat{x} = y^3 - 4y$ and compare the actual x with $\hat{x}$. The difference was evaluated by the averaged square error: $E[(x - \hat{x})^2]$.

Figure 1, 2 and 3 show the results of y predicted by the proposed method with $\varepsilon = 0.1, 0.5, 0.9$, respectively. From these figures, we can see that the threshold value ε is small, and the predicted values show a smooth curve. The estimated errors and number of kernels are listed in Table 1. From this table, the estimated error of the modal regression is reduced when the threshold value is small. However, the number of kernels is increased when the threshold value is small. Therefore, there are tradeoff relationships between the error and number of kernels.

Table 1: Number of kernels and the averaged error for the corresponding x for each threshold value.

Threshold (ε)	0.9	0.5	0.1
No. of kernels	124	188	292
$E[(x - \hat{x})^2]$	0.018	0.010	0.0063

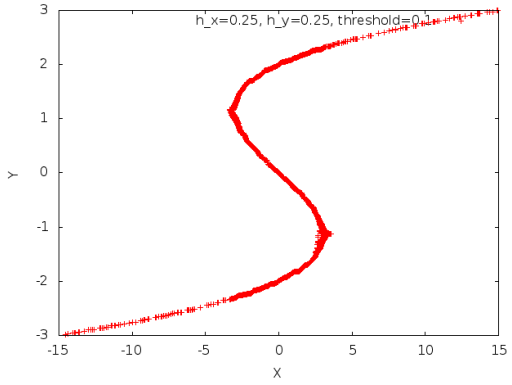

Figure 1: The predicted values from the proposed method with $\varepsilon = 0.1$. The x-axis denotes x and the y-axis denotes the predicted value.

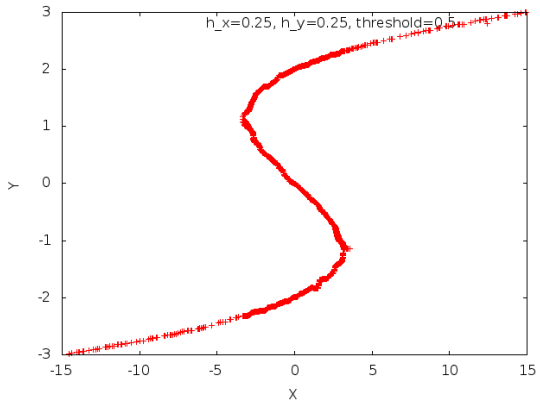

Figure 2: The predicted values from the proposed method with $\varepsilon = 0.5$. The x-axis denotes x and the y-axis denotes the predicted value.

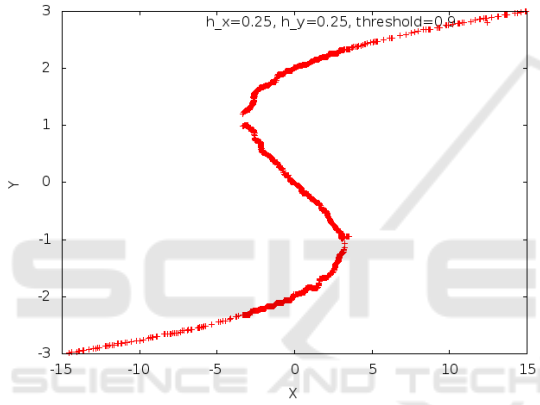

Figure 3: The predicted values from the proposed method with $\varepsilon = 0.9$. The x-axis denotes x and the y-axis denotes the predicted value.

4.1.2 Helix Function

The second dataset is a helix dataset. By using this dataset, we have checked whether our method approximates more complex outputs. The dataset is described as follows.

$$x_t = a_t \cos\theta_t,\ y_t = a_t \sin\theta_t,\ z_t = b_t\theta_t,$$

where $\theta_t = 2\pi t$. We set $a_t = 2 + n_t$, where n_t denotes a uniform random value in the interval [-0.1, 0.1], and $b_t = 3 + n_t$. By increasing t gradually from 0 to 9, 3000 instances were generated. The dataset has a spiral shape. The used hyper parameters were $h_x = 2.0,\quad h_y = 2.0.$ Figure 4 and Figure 5 show the results for a threshold of ε = 0.1 and 0.95. In the case of threshold ε = 0.1, 157 kernels were generated. On the other hand, in the case of threshold ε = 0.95, 45 kernels were generated. In the both cases, the proposed system regenerated almost the same correct multivalued outputs.

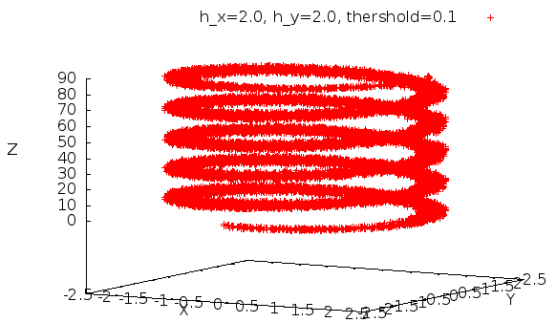

Figure 4: The output of the proposed method of Helix data for a threshold $\varepsilon = 0.1.$

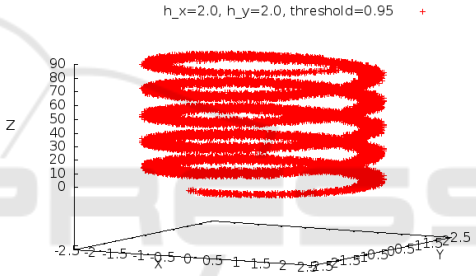

Figure 5: The output of the proposed method of Helix data for a threshold $\varepsilon = 0.95.$

4.2 Performance for Real Dataset

We also tested the proposed method with a real-dataset: Data from the network journey time and traffic flow on highways in England[1]. We used the traffic flow data on January 2006 MIDIAS Site 1030 (LM205) and made the proposed system learn the pairwise data between total carriageway flow versus total flow vehicles above 11.6m. The dataset records the data at every 15 minutes. The four total carriage flows and corresponding speed flow between every 45 minutes are almost the same. Therefore, we picked up the first data of the four data set for the corresponding 45 minutes. By this procedure, we reduced the dataset size to 1/4 (8580 instances). Moreover, each speed data and flow data was normalized by dividing them by 140 and 1400, respectively. The used hyper-parameters are $h_x = 0.15,\quad h_y = 0.2$. From the data plotted in

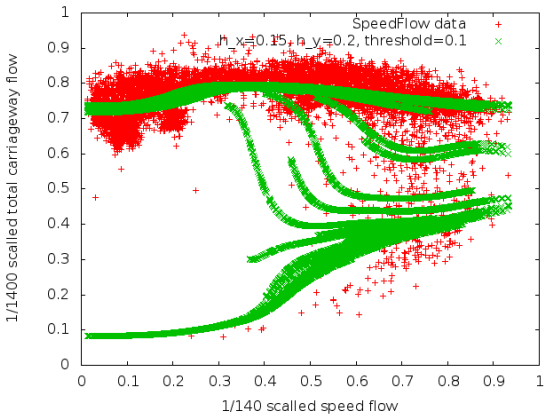

Figure 6 The predicted outputs from the proposed method with $\varepsilon = 0.1$. The generated kernel size was 57.

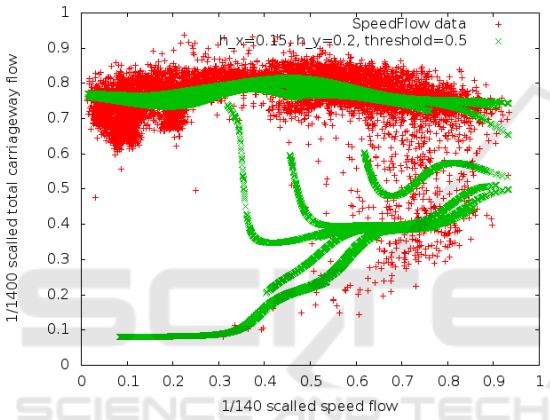

Figure 7: The predicted outputs from the proposed method with $\varepsilon = 0.5$. The generated kernel size was 29.

The predicted outputs from the proposed method with $\varepsilon = 0.1$ and 0.5 are shown in Figure 6 and Figure 7. The kernel sizes were 57 and 29, respectively.

We can see the proposed method predicted more than two distributions in the speed flow.

5 CONCLUSION

In this paper, we proposed a new method for modal regression. While forming the KDE when a new sample is given, it may be projected onto the existing kernel space, it may replace the existing kernel, or a new kernel may be generated with a given sample as the center. This depends on the threshold and the dependencies of each kernel in the existing kernel space. The equation for the PMS method is also modified according to this method by adding weights to the kernels. The experimental results show that the proposed method can approximate the multivalued functions properly, and it also reduces the complexity greatly compared to the case where a kernel is allocated to each sample.

[1]http://tris.highwaysengland.co.uk/detail/trafficflowdata

REFERENCES

Einbeck, J. & Tutz, G. (2006), 'Modelling beyond regression functions: an applica- tion of multimodal regression to speed?flow data', *Applied Statistics* 55(4), 461–475.

He, W. & Wu, S. (2012), 'A kernel-based perceptron with dynamic memory', *Neural Networks* 25, 105–113.

Orabona, F., Keshet, J. & Caputo, B. (2008), The projectron: a bounded kernel-based perceptron, *in* 'ICML2008', pp. 720–727.

Parzen, E. (1962), 'On estimation of a probability density function and mode', *Annals of Mathematical Statistics* 33(3), 1065–1076.

Sasaki, H., Ono, Y. & Sugiyama, M. (2016), Modal regression via direct log-density derivative estimation, *in* A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee & D. Liu, eds, 'Neural Information Processing –23rd International Conference, ICONIP 2016–', Vol. PartII, Springer-Verlag.

Yamauchi, K. (2013), An importance weighted projection method for incremental learning under unstationary environments, *in* 'IJCNN2013: The International Joint Conference on Neural Networks 2013', The Institute of Electrical and Electronics Engineers, Inc. New York, New York, pp. 1–9.

Yamauchi, K. (2014), 'Incremental learning on a budget and its application to quick maximum power point tracking of photovoltaic systems', *Journal of Advanced Computational Intelligence and Intelligent Informatics* 18(4), 682–696.

APPENDIX

The proof of **Theorem 1** is

Proof 1.

From $\left\|\delta_{i^*}\right\|^2 = 0$, we obtain

$$\nabla_x \left\|\delta_{i^*}\right\|^2 = 0 \Leftrightarrow \nabla_x \delta_{i^*} = 0 .$$

Therefore, we also have $\nabla_x^2 \delta_{i^*} = 0$.

From the pruning and replacement operation,

$$\nabla_{\mathbf{x}}\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = \nabla_x\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$- w_i \nabla_x \delta_{i^*}\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$= \nabla_x\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0$$
$$\nabla_{\mathbf{x}}^2\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = \nabla_x^2\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$- w_i \nabla_x^2 \delta_{i^*}\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$= \nabla_x^2\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} < 0$$

This concludes the proof.

The proof of **Theorem 2** is

Proof 2.

From $\|\delta_t\|^2 = 0$, we obtain

$$\nabla_x\|\delta_t\|^2 = 0 \Leftrightarrow \nabla_x \delta_t = 0.$$

Therefore, we also have $\nabla_x^2 \delta_t = 0$.

From the projection operation, we have

$$\hat{p}_t^{'} - k(\mathbf{x}_t,\cdot) = \hat{p}_{t-1} + \delta_t.$$

From this equation, we obtain the following two equations.

$$\nabla_{\mathbf{x}}\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} - \nabla_x K\left(\frac{\|\mathbf{x}-\mathbf{x}_t\|}{h_x}\right)\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$= \nabla_x\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} + \nabla_x \delta_t\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = 0$$
$$\Leftrightarrow \nabla_{\mathbf{x}}\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = \nabla_x K\left(\frac{\|\mathbf{x}-\mathbf{x}_t\|}{h_x}\right)\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$

$$\nabla_x^2\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} - \nabla_x^2 K\left(\frac{\|\mathbf{x}-\mathbf{x}_t\|}{h_x}\right)\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$= \nabla_x^2\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} + \nabla_x^2 \delta_t\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$= \nabla_x^2\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$\Leftrightarrow \nabla_x^2\left\langle \hat{p}_t^{'}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}} = \nabla_x^2\left\langle \hat{p}_{t-1}, k(\mathbf{x},\cdot)\right\rangle\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$
$$+ \nabla_x^2 K\left(\frac{\|\mathbf{x}-\mathbf{x}_t\|}{h_x}\right)\Big|_{\mathbf{x}=\mathbf{x}_{MP}}$$

This concludes the proof.