Moving Least Squares Support Vector Machines for weather temperature prediction

Zahra Karevan, Yunlong Feng and Johan A. K. Suykens

KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Abstract. Local learning methods have been investigated by many researchers. While global learning methods consider the same weight for all training points in model fitting, local learning methods assume that the training samples in the test point region are more influential. In this paper, we propose Moving Least Squares Support Vector Machines (M-LSSVM) in which each training sample is involved in the model fitting depending on the similarity between its feature vector and the one of the test point. The experimental results on an application of weather forecasting indicate that the proposed method can improve the prediction performance.

1 Introduction

In local learning, instead of using all samples to train the model, only those which are in the region of the test point are used for model fitting [1, 2]. For example, in [3] Polynomial Local regression has been investigated. Besides, concerning seasonal behavior of weather variables, in [4], local learning methods have been employed to deploy the local structure of the data to pursue better performance.

Accurate weather forecasting is a major area of interest within the field of climate informatics. Numerical Weather Prediction (NWP) is the most widely used method in the state-of-the-art approaches for weather forecasting. However, one major drawback of NWP is that it is an intense method in terms of computational complexity [5]. Recently, data-driven approaches have been investigated to achieve reliable weather prediction. In our previous work [4], Least Squares Support Vector Machine LSSVM has been used for weather forecasting and the experiments revealed that the performance is competitive with state-of-the-art methods in weather forecasting.

The objective of this paper is to investigate a soft localization in the framework of LSSVM called Moving LSSVM (M-LSSVM). In the proposed method, we deploy the samples in the training set for model fitting while their impact on the model is determined by their similarity to the test point feature vector. This localization is done by defining a weighted cost function in the optimization problem in the primal problem. Moreover, to obtain a good generalization, tuning the parameters is done by using Moving Cross Validation (M-CV) in which each sample affects the validation error based on its similarity to the test point. In this study, in order to evaluate the proposed method, we conduct our experiments on an application of forecasting the maximum and minimum temperature of Brussels for 1 to 6 days ahead.

2 Moving Least Squares Support Vector Machines

In global learning methods, the same weights are considered for all data points in the training data while local learning algorithms assume that the samples in the the test point vicinity are more influential for model fitting. In Moving Least Squares (MLS), it is assumed that the training samples may not have similar importance in function estimation. Thus, each training point has a weight which is based on the distance between the training sample and the test point [6].

In this paper, a soft localization in the framework of Least Squares Support Vector Machines (LSSVMs) is proposed. LSSVM differs from Support Vector Machines (SVMs) in the sense that instead of quadratic programming in SVM, LSSVM results in solving a set of linear equations by taking equality constraints and least squares loss [7].

Considering $x \in \mathbb{R}^d$, $y \in \mathbb{R}$ and $\varphi : \mathbb{R}^d \to \mathbb{R}^h$ where $\varphi(\cdot)$ being the feature map, the model in primal space is formulated as:

$$\hat{y}_x(x) = \hat{w}_x^T \varphi(x) + \hat{b}_x, \tag{1}$$

where $\hat{b}_x \in \mathbb{R}$ and $\hat{w}_x \in \mathbb{R}^h$ are estimated for a given x. Note that $\hat{y}_x(x)$ is depending on x explicitly and implicitly since not only it is a function of x, but also the optimization problem in primal space is done for any fixed x.

Given $s_i(x)$ as a non-negative similarity measure between the ith training data feature vector and any fixed x, the optimization problem for training M-LSSVM model in primal space is as follows:

$$\begin{aligned} (\hat{w}_x, \hat{b}_x, \hat{e}_x) = \min_{w,b,e} \quad & \tfrac{1}{2} w^T w + \tfrac{\gamma}{2} \sum_{i=1}^{N} s_i(x) e_i^2, \\ \text{s. t.} \quad & y_i = w^T \varphi(x_i) + b + e_i, i = 1, ..., N. \end{aligned} \tag{2}$$

Note that Weighted LSSVM, proposed in [8], uses a similar formulation to obtain robust estimation for regression problems. Here a different function is estimated for any given fixed x value.

In the proposed method, the similarity criterion $s_i(x)$ can be either a binary or a real value. The former implies that only part of the training samples in the vicinity of the test point is used to train the model. Nevertheless, for real valued similarity, all samples in the training set are involved for learning the data while their influence on the model fitting is determined by their similarity to the test point. In this paper, we use the Gaussian similarity criterion $s_i(x) = \exp(-||x-x_i||_2^2/h^2)$ and the cosine-based similarity function $s_i(x) = \frac{x^T x_i}{||x|| \times ||x_i||} + 1$ where $||x||$ is the $L2$-norm of the vector x. Note that, the former has a bandwidth parameter (h) to be tuned, while the latter has no tuning parameter. As a specific case, if $s_i(x)$ is set equal to one, (2) represents the classical LSSVM formulation in primal space.

From the Lagrangian $\mathcal{L}(w, b, e; \alpha) = \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} s_i(x)\ e_i^2 - \sum_{i=1}^{N} \alpha_i (w^T$

$\varphi(x_i) + b + e_i - y_i)$, the optimality conditions can be expressed as (3) below.

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i), \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0, \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma s_i(x) e_i, i = 1, ..., N, \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \rightarrow y_i = w^T \varphi(x_i) + b + e_i, i = 1, ..., N, \end{cases} \tag{3}$$

where $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers. Assuming $y = [y_1, \cdots, y_N]^T$ and $1_N = [1, \cdots, 1]^T$ and after eliminating w and e, the dual problem is written as follows

$$\left(\begin{array}{c|c} 0 & 1_N^T \\ \hline 1_N & \Omega + S_\gamma(x) \end{array}\right) \left(\frac{b}{\alpha_i}\right) = \left(\frac{0}{y}\right), \tag{4}$$

where $S_\gamma(x)$ is a diagonal matrix equal to $S_\gamma(x) = diag([\frac{1}{s_1(x)\gamma}; \cdots; \frac{1}{s_n(x)\gamma}])$ and Ω is the kernel matrix where based on Mercer's theorem [9]:

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j) \quad i, j = 1, 2, \ldots, N. \tag{5}$$

In this paper, the Radial Basis Function (RBF) is used as a kernel function which is formulated in (6). In this case, the regularization parameter γ and the kernel bandwidth σ are tuning parameters.

$$K(x_i, x_j) = \exp(-||x_i - x_j||_2^2 / \sigma^2). \tag{6}$$

Considering $\hat{\alpha}_{i,x}$ and $\hat{b}_x$ as the solution of the linear equation system (4), the M-LSSVM model as a function estimator is formulated as follows:

$$\hat{y}_x(x) = \sum_{i=1}^{N} \hat{\alpha}_{i,x} \; K(x, x_i) + \hat{b}_x. \tag{7}$$

Here the notation of the solution ($\hat{\alpha}_{i,x}$, $\hat{b}_x$) stresses that (4) needs to be solved for any given x.

Although the proposed method can improve the performance, it has its own drawback. The proposed method is a time consuming approach since the model fitting should be done for each test point independently. These methods are not suitable when the size of the test set is large. On the other hand, in the case of time series problems, one may leverage the local learning methods since each time there is only one test point to be predicted.

2.1 Tuning the parameters

In order to obtain a good generalization for the models, one may use k-fold Cross-Validation (CV) for tuning the parameters. Obviously, for the proposed method CV is not an ideal tuning approach, since the model is not trained properly for the regions which are not close to the test point.

In [10], the authors suggested Robust Cross Validation to deal with outliers and non-Gaussian noise. In this paper, we use k-fold Moving Cross Validation (M-CV), in which each sample in the validation set affects the validation error based on the similarity between its feature vector to the test point. Let k be the number of folds and N_v be the number of samples in the vth fold. The k-fold M-CV error for any fixed x is calculated as follows

$$Error_{M-CV}(x) = \frac{\sum_{v=1}^{k} \sum_{i=1}^{N_v} s_i(x)\ err_i}{\sum_{i=1}^{N} s_i(x)}, \tag{8}$$

where err_i is a performance evaluation method. In this paper, the Mean Square Error (MSE) and Mean Absolute Error (MAE) are used to measure err_i.

3 Experiments

3.1 Dataset

In this study, data have been collected from the Weather Underground website and include real measurements for weather elements such as temperature and humidity for 11 cities including Brussels, Liege, Antwerp, Amsterdam, Eindhoven, Dortmund, London, Frankfurt, Groningen, Dublin, and Paris. The performance of the proposed method is evaluated on two test sets: (i) from mid-November 2013 to mid-December 2013 (Nov/Dec) and (ii) from mid-April 2014 to min-May 2014 (Apr/May).

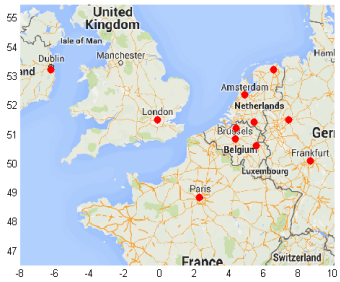

Fig. 1: Cities included in the model

The data cover a time period from beginning 2007 to mid 2014 and comprise 198 measured weather variables for each day. The total number of features is equal to $198 \times lag$, where lag is a tuning parameter which indicates the number of previous days considered in the model. Note that the number of training samples was determined by the number of days from the beginning of 2007 until the day before the test date (varies from 2489 to 2667 points).

3.2 Results and discussion

In this section, we evaluate the proposed method on an application of weather forecasting and as a case study, the prediction of the minimum and maximum temperature in Brussels for 1 to 6 days ahead was considered. The tuning parameters including the lag variable, the kernel bandwidth (σ) and the regularization constant (γ) of M-LSSVM, and the kernel bandwidth parameter in the case of Gaussian similarity (h) were tuned using M-CV. We run our experiments 10 times and in Table 1, the average MAE and MSE of the prediction of the proposed method is compared with those of LSSVM for two test sets. As it is shown, mostly, M-LSSVM outperformed LSSVM. Furthermore, it can be

seen that using the RBF similarity criterion tends to be more advantageous than cosine-based one.

Test set	Days ahead	Temp.	Mean Absolute Error			Mean Square Error		
			LSSVM	M-LSSVM-RBF	M-LSSVM-Cosine	LSSVM	M-LSSVM-RBF	M-LSSVM-Cosine
Nov/Dec	1	Min	1.44	1.44	**1.35**	3.70	3.65	**3.53**
		Max	1.19	**1.03**	1.13	2.88	**2.14**	2.53
	2	Min	**1.55**	**1.55**	1.57	**4.47**	**4.47**	4.50
		Max	1.23	1.26	**1.20**	3	3.11	**2.90**
	3	Min	1.65	1.67	**1.62**	**5.26**	**5.26**	5.29
		Max	1.48	**1.30**	1.48	4.19	**3.76**	3.98
	4	Min	**1.61**	**1.61**	1.64	4.53	**4.50**	4.61
		Max	1.37	1.37	**1.32**	**2.82**	**2.82**	**2.82**
	5	Min	**1.50**	1.53	1.53	**3.96**	4	4
		Max	**1.12**	1.17	**1.12**	1.94	2.10	**1.90**
	6	Min	1.69	1.63	**1.62**	4.69	**4.55**	**4.55**
		Max	1.44	1.41	**1.41**	**3.96**	**3.96**	4.08
Apr/May	1	Min	1.46	**1.29**	1.39	3.22	**3.03**	3.24
		Max	2.25	2.17	**2.09**	7.46	7.28	**7.04**
	2	Min	1.85	**1.80**	1.82	6.39	6.25	**6.19**
		Max	2.16	2.11	**2.08**	7.54	**7.38**	**7.38**
	3	Min	1.70	**1.65**	1.70	5.33	**5.14**	5.33
		Max	**2.43**	**2.43**	2.48	**8.11**	8.17	8.68
	4	Min	1.74	**1.62**	1.75	4.85	**4.65**	4.93
		Max	**2.44**	2.49	2.48	**8.54**	8.62	9.38
	5	Min	**1.96**	**1.96**	**1.96**	**6.40**	6.48	6.60
		Max	2.33	2.23	**2.20**	7.43	7.57	**7.34**
	6	Min	2.18	**2.08**	2.18	8.03	**7.80**	8.03
		Max	**2.50**	**2.50**	2.55	**8.43**	**8.43**	9.31

Table 1: Average MAE and MSE of the predictions by LSSVM and M-LSSVM based on Cosine and RBF similarity $s_i(x)$ for test sets Nov/Dec and Apr/Nov.

Figure 2 represents a comparison between the performance of Weather Underground predictions and the one of the data-driven models over both test sets together for the minimum and maximum temperature in Brussels. As it is depicted, for minimum temperature, data-driven models mostly outperformed Weather Underground predictions and for maximum temperature the performances were competitive.

4 Conclusion

The aim of the present research was to propose a Moving LSSVM method in which each training sample influence on the model fitting is depending on the similarity between its feature vector and the one of the test point. The proposed method was evaluated based on temperature prediction in Brussels for 1 to 6 days days ahead. The experimental results suggest that utilizing local learning can improve the accuracy of forecasting. In addition, data-driven methods have shown competitive performance with the state-of-the-art methods in weather forecasting.

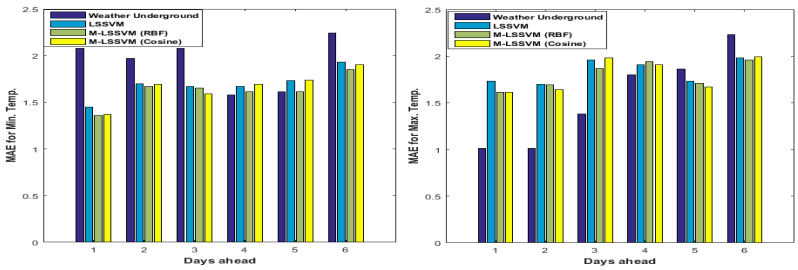

Fig. 2: MAE of the predictions for Weather Underground, LSSVM and M-LSSVM with RBF and cosine based similarity for Max. and Min. temperature

Acknowledgment

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. Research Council KUL: CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grant iMinds Medical Information Technologies SBO 2015 IWT: POM II SBO 100031 Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017).

References

[1] Léon Bottou and Vladimir Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

[2] Clive Loader. *Local regression and likelihood*. Springer Science & Business Media, 2006.

[3] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*, volume 66. CRC Press, 1996.

[4] Zahra Karevan and Johan A. K. Suykens. Clustering-based feature selection for black-box weather temperature prediction. In *Int. Joint Conf. on Neural Networks*, 2016.

[5] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

[6] David Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998.

[7] Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[8] Johan A. K. Suykens, Jos De Brabanter, Lukas Lukas, and Joos Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1):85–105, 2002.

[9] James Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of The Royal Society of London. Series A, containing papers of a mathematical or physical character*, pages 415–446, 1909.

[10] Jos De Brabanter, Kris Pelckmans, Johan A. K. Suykens, Joos Vandewalle, and Bart De Moor. Robust cross-validation score functions with application to weighted least squares support vector machine function estimation. Technical report, KU Leuven, 2003.