Embedding based Regression in Nonlinear System

Dae Yon Jung, Donghwan Kim, Kihyuk Sohn, and Vijay Manikandan Janakiraman

December 16, 2011

Abstract

Analyzing a nonlinear system on high dimensions is not simple, especially with the practical applications. Since the prediction of the output from the dynamical system can be formulated as a regression problem, many previous attempts have been made to solve this regression. With the assumption of nonlinearity in the system, we attempt to apply encoding method, such as sparse coding or local coordinate coding (LCC), to unwrap the nonlinear included in the inputs, and then the linear regression method is used afterwards. We also presented a novel encoding method that is deriven from the locality constrianing idea from LCC, which is called soft assignment clustering. In order to validate the robustness of each combination of methods, the swiss-roll data set and the internal combustion engine (ICE) data set with noise are analyzed.

1 Introduction

A nonlinear system with a high dimensional input is common in the real world. In fact, it is not realistic to assume the linearity in practical system analyses, such as estimating climates or modeling human brain neurons, because of its multivariate dependency and complexity. The most trendy methods to analyze these complex systems, rather than analyzing mathematically, are to encode the data to make the desired information more explicit, such as in sparce coding [1], or to model and learn the relationship between the input and the output with a sufficient amount of data, such as in neural network [2]. By the previous work from numerous fields, these types of methods are proven to give robust analyses of the systems.

One of the nonlinear dynamical system that we focus in this project is the Internal Combustion Engines (ICE), a primary power generator for automobiles. The dynamics of the ICE is totally unknown; only the sequences of inputs which indicate variables at each time such as fuel mass and injection time, and their corresponding outputs, such indicating efficiency and emission powers, are given. The objective is to predict the current output based on the given past outputs and past and present inputs.

In order to achieve this objective, several advanced machine learning methods are applied and comparatively studied to identify the advantages and the disadvantages of each method. Generally, the prediction of the present output based on inputs and outputs in a dynamical system can be formulated as a regression problem. Therefore, regession models such as Support Vector Regression

(SVR) are directly used. SVR is an application of support vector machine (SVM) to a regression problem, having the advantage of possible use of kernels. But, for a large dataset, this kernelization increases computational expenses.

However, using only linear regression (or linear ridge regression) does not give reasonable predictions since it is known that the systems are nonlinear. Thus, encoding methods, which are to transform the inputs to explicit and informative representations, are used before applying the regression method. Sparse coding and its extend with the codebook's locality constraint, which is called Local Coordinate Coding (LCC), are used prior to the linear regression in order to releave the nonlinearity implied in the data.

The significant difference between sparse coding and local coordinate coding is the penalty on the distance between the data points and codebook bases. Inspired by the enhancement that LCC has over sparse coding, we developed a new encoding method that emphasizes that locality constraint. The hard assigned locality constraint is mathematically identical to kmeans clusting. We extend this clustering algorithm to soft assignment so that it can have a greater expressive power than kmeans.

In terms of the evaluations, all the methods are tested on the standard nonlinear swiss roll data, which is introduced in the paper by Roweis et al. [3]. The performance on the ICE data, which is a more practical dataset, is also compared. As a way to check the invariance of the method to noise, different noises are added: noisy dimensions are added to the swiss roll data, and Gaussian and exponential noises are added onto the existing ICE data.

The rest of the report is organized as follows. Section 2 describes the regression pipelines with the existing methods. Especially, our extended novel methods are explained in Section 3. In Section 4, the experimental settings and results are presented. Finally, Section 5 concludes the paper.

2 Model Descriptions

In this section, all the algorithms that are used in this project to solve the regression problem in the dynamical system is explained. For better understanding, regression models are explained first, and then each encoding model, which can be combined with the regression models, is described.

2.1 Regression models

2.1.1 Support Vector Regression (SVR)

The objective of ϵ-SVR algorithm [4], chosen among several SVR methods in this project, is to find a function $f(x)$ that has at most ϵ deviation from the actual target y from all training data x while avoiding the over-fitting. The linear SVR model can be represented as $f(\mathbf{x}) =< \mathbf{w}, \mathbf{x} > +b$, where $\mathbf{w}$ and b are obtained by solving the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}(\zeta^{(n)} + \zeta_*^{(n)}) \\ &\text{subject to} \begin{cases} y^{(n)} - <\mathbf{w}, \mathbf{x}^{(n)}> - b \leqslant \epsilon + \zeta^{(n)} \\ <\mathbf{w}, \mathbf{x}^{(n)}> + b - y^{(n)} \leqslant \epsilon + \zeta_*(n), \qquad \text{for } n = 1, \cdots, N \\ \zeta^{(n)}, \zeta_*^{(n)} \geqslant 0 \end{cases} \end{aligned} \tag{1}$$

By solving the dual problem of (1) with dual variables $\alpha^{(n)}$ and $\alpha_*^{(n)}$, the regression model can be expressed as $f(\mathbf{x}) = \sum_{n=1}^{N}(\alpha^{(n)} - \alpha_*^{(n)}) < \mathbf{x}^{(n)}, \mathbf{x} > +b$. The extension of the algorithm to nonlinear function approximation is straightforward using inner product kernels.

We attached the neural network in front of SVR to additionally capture the nonlinearity of the function to be predicted. In other words, a hidden layer in the neural network gives more explicit representation of the input, and less task will be assigned to the function of SVR.

2.1.2 Linear regression and multi-dimensional output extension

In this work, we used simple linear ridge regression on top of various embedding algorithms, which is formulated as follows:

$$\min_{\mathbf{w},b} \frac{1}{N}\sum_{n=1}^{N} \|y^{(n)} - \mathbf{w}^T\mathbf{x}^{(n)} - b\|^2 + \lambda\|\mathbf{w}\|^2 \tag{2}$$

For the case of multi-dimensional output data $\mathbf{y} \in \mathbb{R}^p$, we use the straightforward extension of the above model which we call a coordinate-wise linear regression as follows:

$$\min_{\mathbf{w}_{j=1}^p, b_{j=1}^p} \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{p} \|y_j^{(n)} - \mathbf{w}_j^T\mathbf{x}^{(n)} - b_j\|^2 + \lambda\sum_{j=1}^{p}\|\mathbf{w}_j\|^2 \tag{3}$$

Similarly, we use the coordinate-wise SVR to predict multi-dimensional output data. [1]

2.2 Encoding Method

2.2.1 Sparse Coding

Sparse coding is one of the commonly used algorithm to transform the inputs to concise and computationally useful representations. Having a group of bases, called the codebook, the linear combinations of these bases are chosen to present the input closest in terms of the least square error. When this code, which indicates the linear combinations, is constrained with a sparsity condition, it is called the sparse code. The idea of sparse coding resembles the systematic structure of the

[1] There exist some methods about the multi-output SVRs [5], but our preliminary result showed that the multi-output SVR is not as good as coordinate-wise SVR in terms of performance and the computational efficiency.

biological neurons [1]. For example, when a visual input flows into the system, several receptive fields of neurons are activated at each moment to determine what the visual is. These pre-existing receptive fields are the codebooks, and the information about the activation is the sparse code that is used as a new representation of the original input.

The mathematical formulation of the sparse coding is to minimize the summation of the squared error and the L_1 regularization, which induces the sparsity:

$$\begin{aligned} \min_{z,\mathbf{C}} \quad & \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{x}^{(n)} - \sum_{k=1}^{K} z_k^{(n)} \mathbf{c}_k\|^2 + \lambda \sum_{k=1}^{K} |z_k^{(n)}| \\ \text{subject to} \quad & \sum_{k=1}^{K} \|\mathbf{c}_k\|^2 \le c \end{aligned}$$

,where $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_K\}$ is a codebook and $z_k^{(n)}$ is a corresponding coding coefficient. These type of coding based on the codebook has the flexibility to vary the size of the codebook, n. When n is greater than the dimension of the input, it will have an overcomplete basis, which will make the sparse representation more meaningful.

2.2.2 Local Coordinate Coding (LCC)

Local coordinate coding [6] is an extension of sparse coding, which can be viewed as a locally constrained sparse coding. In other words, the local coordinate coding enforces to have sparsely activated codebooks which are located near the data point. This algorithm has a nice explanation to the question why locality condition is important when learning a nonlinear function, while the sparse coding does not have theoretical answer to the need of sparsity.

Data represented in a high-dimensional space usually lie on a manifold which has smaller dimension. It is shown in the paper [6] that locally constrained coding can fairly represent the data on a manifold. It is further proven that a high-dimensional nonlinear function can be globally approximated by a linear function with a local coordinate coding.

The local coordinate coding is formulated by minimizing the following objective function, which added locality constraining term $\|\mathbf{x}^{(n)} - \mathbf{c}_k\|^2$ to the sparse coding formulation.

$$\begin{aligned} \min_{z,\mathbf{C}} \quad & \sum_{n=1}^{N} \frac{1}{2} \|\mathbf{x}^{(n)} - \sum_{k=1}^{K} z_k^{(n)} \mathbf{c}_k\|^2 + \lambda \sum_{k=1}^{K} |z_k^{(n)}| \|\mathbf{x}^{(n)} - \mathbf{c}_k\|^2 \\ \text{subject to} \quad & \sum_{k=1}^{K} \|\mathbf{c}_k\|^2 \le c \end{aligned}$$

3 Proposed Encoding Methods

3.1 Soft Assignment of Clustering Algorithm

The success of local-coordinate coding of linear embedding on highly nonlinear manifold is coming from the weighted L_1 penalty where the weights are determined as a distance between the data point and the anchor point. In fact, the Kmeans clustering algorithm is equivalent to the LCC algorithm in the extreme case where $\lambda \rightarrow \infty$[2]. Therefore, the Kmeans clustering algorithm can be interpreted as a coding method that In this section, we consider two algorithms; 1) the clustering algorithm with soft-assignment and 2) the neural network with local encoding and linear decoding functions. To be more specific, the soft-assignment of the clustering algorithm is defined as follows:

$$\text{E-step: } z_k^{(n)} = \frac{\exp(-\frac{1}{\sigma}\|\mathbf{x}^{(n)} - \mathbf{c}_k\|^2)}{\sum_{k'=1}^{K}\exp(-\frac{1}{\sigma}\|\mathbf{x}^{(n)} - \mathbf{c}_{k'}\|^2)} \tag{4}$$

$$\text{M-step: } \min_{\mathbf{C}} \sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \sum_{k=1}^{K} z_k^{(n)}\mathbf{c}_k\|^2 \tag{5}$$

where $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_K\}$ is the set of cluster centroids, $z_k^{(n)} \in [0, 1]$ is the cluster assignment of the data x. The parameter σ plays a role of balancing between locality and the reconstruction error, which is similar to the role of λ in LCC, and the smaller value of σ considers the locality more in soft encoding function. The main advantage of this model compared to the LCC algorithm is at the efficient encoding (i.e., the cluster assignments are given in closed form whereas the LCC requires to solve the expensive weighted L_1 regularized least square problem) while maintaining the locality.

3.2 Encoding Functions of Sparse Autoencoder

Compared to the distributed models such as artificial neural networks or sparse coding, the soft clustering algorithm lacks the expressive power. Therefore, it is natural to extend the model to the factorial algorithms. To that end, we propose a sparse autoencoder (AE), one of the most widely used artificial neural networks in unsupervised learning community. As usual, the sparse AE is composed of an objective function and the encoding, decoding functions. The objective function is given as follows:

$$\min_{\mathbf{C},\mathbf{b},\mathbf{d}} \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_{\text{loss}}(\mathbf{x}^{(n)}) + \lambda\mathcal{L}_{\text{sparse}}(\{\mathbf{x}^{(n)}\}_{n=1}^{N}). \tag{6}$$

The loss function and the sparsity penalty terms are written as

$$\mathcal{L}_{\text{loss}}(\mathbf{x}) = \|\mathbf{x} - g(\{f_k(\mathbf{x})\}_{k=1}^{K})\|^2, \quad \mathcal{L}_{\text{sparse}}(\{\mathbf{x}^{(n)}\}_{n=1}^{N}) = \sum_{k=1}^{K} \rho\log\frac{\rho}{\hat{\rho}_k} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_k} \tag{7}$$

[2] To make this equivalence strict, we need a nonnegativity ($z_k^{(n)} \geq 0$) and the shift invariance property ($\sum_{k=1}^{K} z_k^{(n)} = 1$) of the coefficients as well.

where $\hat{\rho}_k = \frac{1}{N}\sum_{n=1}^{N} f_k(\mathbf{x}^{(n)})$. Common selection of encoding function is sigmoid or hyperbolic tangent functions. In this work, we introduce a new encoding function that explicitly encodes the distance between the data and the feature maps in our model to consider locality. We use the linear decoding function.

$$f_k(\mathbf{x}) = \mathrm{sigm}(-\frac{1}{\sigma}\|\mathbf{x} - \mathbf{c}_k\|^2 + b_k), \quad g(\{f_k(\mathbf{x})\}_{k=1}^{K}) = \sum_{k=1}^{K} w_k f_k(\mathbf{x}) + d, \tag{8}$$

where b_k is the hidden unit bias and d is the input bias. Likewise, we regularize the effect of locality with σ. In fact, the only difference between the one in (8) and the sigmoid encoding function is the existence of σ and the L_2 norm of data, and $\|\mathbf{x}\|^2$ can be removed once we normalize the data before training. For regression problems, however, the norm of the data might be an important factor that discriminates the data and blindly normalizing the data might lose a lot of information. Therefore, our encoding function can be a better fit to this work.

3.3 Local Sparse Coding

While the LCC enjoys the mercy of locality in nonlinear manifold embedding problems, the locality penalty given in (4) may not be the only way of encouraging locality during the coding phase. In this section, we consider the variation of locality constraint that explicitly enforce the sparsity as well as the contraction in the feature space, so that achieves the invariance as well. We call this method as a "local sparse coding (LSC)" and this can be done by solving the following regularized least square problem:

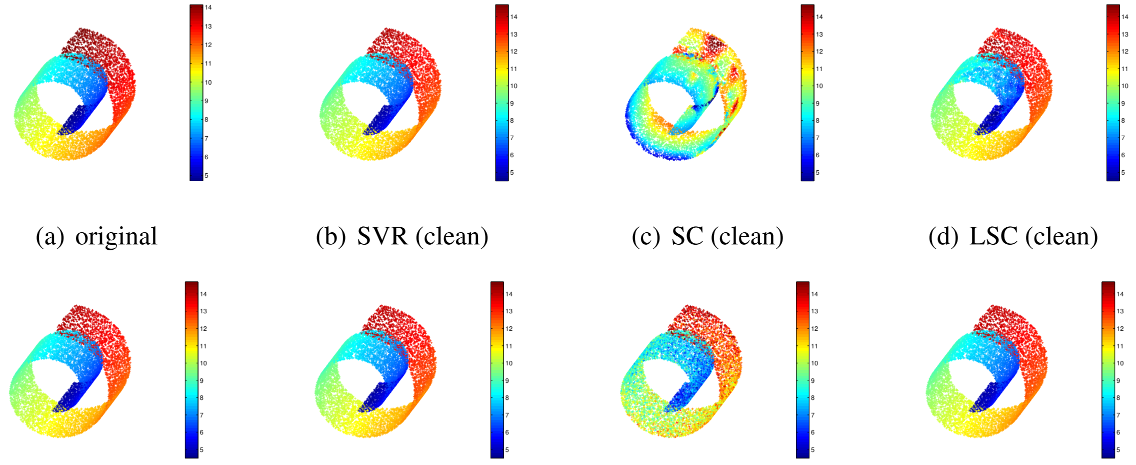
$$\min_z \|\mathbf{x} - \sum_{k=1} z_k \mathbf{c}_k\|^2 + \lambda \sum_{k=1}^{K} \frac{|z - z^{(k)}|}{\|x - \mathbf{c}_k\|^2} \tag{9}$$

$$\text{subject to} \sum_{k=1}^{K} z_k = 1, z_k \geq 0, \forall k = 1, \cdots, K \tag{10}$$

where $z^{(k)} \in \{0, 1\}^K$ is a binary-valued K-dimensional vector whose k-th coordinate is 1 and zero for all other coordinates. The intuition behind the regularizer is the following: when the data x is similar to the one of the anchor point w_k in the Euclidean space, we want the feature representation z of the example x to be similar to the representation of w_k, which is $z^{(k)}$ and by satisfying this, we can reduce the value of regularizer. Furthermore, by having a L_1 distance between the representations in the feature space, we can naturally enforce stronger sparsity than the regularizer in LCC, which can be a potential advantage of the model.

In fact, it is not trivial to solve the optimization problem (9) due to the L_1 norm. Fortunately, the nonnegativity together with the shift-invariance constraints make the problem much simpler and this can be reformulated as follows:

$$\min_z \|x - \sum_{k=1} w_k z_k\|^2 + 2\lambda \sum_{k=1}^{K} \frac{1 - z_k}{\|x - w_k\|^2} \tag{11}$$



(a) original (b) SVR (clean) (c) SC (clean) (d) LSC (clean)

(e) soft clustering (clean) (f) LE-SAE (clean) (g) SVR (noisy) (h) soft clustering (noisy)

Figure 1: Visualization of swiss roll data, where (a) is plot for original test examples, (b)-(f) are the plots for the predicted examples with different embedding and regression methods on clean data, (g)-(h) are the plots with worst and best rmse on noisy data, respectively.

with the constraint in (10). The problem (11) can be solved using iterative methods, such as the sequential minimal optimization (SMO) methods.

4 Experimental results and discussion

We tested the regression models described in Section on two datasets; 1) the swiss roll dataset that is commonly used in evaluating the performance of manifold embedding algorithms, and 2) the internal combustion engine (ICE) dataset where we have a multi-dimensional output data. For all cases, we visually evaluate the algorithms for qualitative purpose, and report the root mean sum of squared error (RMSE) $\sqrt{\frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{p}(y_j^{(n)} - \hat{y}_j^{(n)})^2}$ as a quantitative performance measure.

4.1 Swiss roll dataset

The swiss roll dataset, originally introduced by Roweis et al. [3], is synthetically generated data for the purpose of evaluating the manifold embedding or nonlinear regression algorithms. The input is three dimensional, indicating the x-, y-, and z-coordinates, while the output is one dimensional, indicating a function value between $\frac{3}{2}\pi$ and $\frac{9}{2}\pi$. The original swiss roll data is visualized in Figure 1(a), where the output values are color coded.

We followed the experimental setting used in [6], where the randomly generated 50,000 samples are used for unsupervised codebook (or cluster centroids for clustering algorithm) and each 500 samples randomly selected from them are used to learn supervised regression models and

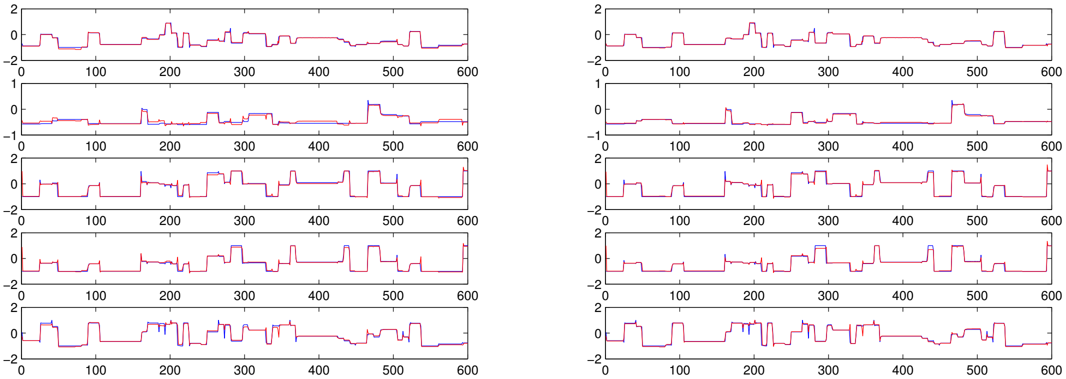
	SR (clean)	SR (noisy)	ICE (clean)	ICE (gaussian)	ICE (exponential)
ϵ-SVR	0.037	1.05	0.23	0.26	0.26
SC	2.19	2.29	0.25	0.28	0.29
LSC	0.24	0.14	0.25	0.26	0.27
nn-SVR	0.031	1.01	**0.22**	0.28	**0.25**
soft clustering	**0.015**	**0.051**	0.23	**0.25**	0.27
LE-SAE	0.028	0.166	0.23	0.26	0.26

Table 1: Regression results on swiss roll (SR) clean and noisy datasets, and ICE clean and noisy datasets with gaussian noise and exponential noise. nn-SVR stands for ϵ-SVR on top of neural network, and LE-SAE is a shorthand notation for the sparse AE with locality encoding function. RMSE between the original and predicted output was reported as a performance measure.

validate the corresponding hyperparameters. Finally, another set of randomly generated 10,000 examples are used for testing with the regression model with the minimum validation error. In addition, we evaluate the regression performance on the noisy swiss roll dataset to see the robustness of the algorithms to noise. As suggested in [6], we generated the 256-dimensional noisy data by adding additional 253-dimensional multivariate gaussian noise with zero mean and identity covariance matrix. By using the Instead of using additive noise, we are able to check the methods' tolerance toward the noise as well as the high-dimensionality [6].

We considered several different algorithms to fit the swiss roll data. First, we considered ϵ-SVR as a shallow model that applies a regression on the raw data. To fit the nonlinearity of the data well, we kernelize the model with radial bases function (rbf). [3] Then, we also implemented several manifold embedding algorithms with linear regression, such as sparse coding, local-coordinate coding, neural networks, clustering with soft-assignment, or local encoding sparse AE, which we described in Section 4. We summarized the results of these algorithms in table 1. As can be seen, we obtained the best rmse result with the clustering algorithm with soft cluster assignments on both clean and noisy swiss roll datasets. Interestingly, most of the algorithms showed reasonably consistent results between clean and noisy datasets. However, the performance of ϵ-SVR on noisy dataset was far from that on the clean dataset, and we further observed a significant overfitting (training rmse 0.05 vs. testing rmse 1.046). This is probably because we used too small number of examples compared to the input dimensionality (500 examples vs. 256 input dimension) for training the ϵ-SVR. On the other hand, however, the computational complexity of the kernelized regression methods increases cubically to the number of training examples, and therefore it is not feasible to increase the number of training examples to avoid the overfitting. We believe that this is a bottleneck of kernelized algorithms, which forces us to use a general regression pipeline that is composed of manifold embedding and the linear regressor on top.

[3] Actually, the linear regression methods on the raw data showed very poor performances, which resulted in 2.628 for linear ϵ-SVR and 2.645 for linear ridge regression. In the main text, we refer ϵ-SVR as a kernelized SVR with rbf kernel unless otherwise stated.



(a) sparse coding (clean)

(b) soft clustering (clean)

Figure 2: Visualization of ICE clean dataset, where (a) is the plot of predicted values (red curve) using sparse coding (the worst fitting model), (b) is the plot of predicted values (red curve) using soft-clustering (the best fitting model). Both figures come with the original data (blue curve). Each subplot corresponds to each output dimension.

4.2 Internal Combustion Engine

The Internal Combustion Engine (ICE) dataset has 20,000 training examples and 8,106 testing examples that are recorded using a simulation model of the spark ignited engine in GT Power software. The fuel mass per cycle (f_m), fuel injection timing (θ_{inj}) and the engine speed (n_{eng}) are varied in multiple steps to constitute the system inputs while the net mean effective pressure (NMEP; an indication of the work output of the engine), combustion phasing (CA50), hydrocarbon (HC) emissions, carbon monoxide (CO) and indicated efficiency ($\textit{eff}_{\text{ind}}$) constitute the outputs of the system. In other words, at each time frame, we have a 3 dimensional input and the dynamic system of ICE generates the 5 dimensional output. For research purpose, the simulated sequence of data is sampled and the signals are concatenated depending on the order of the system (d; number of delays) to form an input $X_t = [x_t; \cdots ; x_{t-d+1}; y_{t-1}; \cdots ; y_{t-d+1}]$ and the output y_t is predicted based on that. All signals are normalized to lie between -1 and 1 before use. To get a better sense of how ICE dataset looks like, we plot the part of the output sequences in Figure 2.

For all experiments, we use the whole training data during the unsupervised feature learning phase, and for the supervised training, we divide the training set into two parts and use the first half for the training the regression model and validate on the other half of the training examples. Once the best hyperparameters are selected via hold-out cross validation, we retrain the regression model using the whole training examples and test with the testing examples. Furthermore, the datasets are augmented with additive gaussian and exponential noise to evaluate the robustness of the algorithms for different types of noise that could possibly be encountered in a real world situation while modeling the ICE.

The summary results are reported in table 1. As can be found in the table, most of the proposed algorithms worked well and the sparse coding with linear regression model, which is known to be a poor method in nonlinear manifold embedding [6], performed a bit worse than the other, but within a reasonable range of error. Actually, visual inspection in Figure 2(a) suggests that the data is not highly nonlinear, and therefore even the sparse coding is enough to fit the ICE model well. Since all other methods showed reasonably similar performance, we additionally plot the predicted output values using the soft clustering algorithm, which might be the most computationally efficient method amongst all presented in this work.

5 Concluding Remarks

In this project, we dealt with the regression problems on the data with highly nonlinear structure. Two regression pipelines, which are 1) the kernelized regression on the raw data, and 2) the combination of embedding on nonlinear feature space and the simple linear regression on top, have been extensively tested on the two datasets, the commonly used swiss roll data and the internal combustion engine data which is known to be a highly nonlinear dynamical system. Throughout the experiments, we verified that the kernelized regression pipeline is actually a descent regression model when we have small number of examples with clear structure. For the case of large-scale and complex data, kernel method turned out to be inefficient, and the second pipeline showed more promising results. Specifically, we have verified an importance of locality in the context of manifold embedding via the set of alternative algorithms such as soft clustering, local encoding sparse autoencoder, and local sparse coding. Further, these methods outperformed the LCC in the regression on swiss roll dataset based on the results in the paper by Yu et al. [6]. [4]

As a future work, we are planning to further explore the properties of locality for other tasks, such as image classification. We envision that the required properties of features in regression and the classification problems are different and therefore the algorithms that we developed in this project might fail to work well in classification problems. Especially, the classification problems require much richer feature representations, and the soft clustering algorithm or the current version of LSC (with shift-invariance constraint) are limited in that sense. Therefore, we would like to investigate more on the sparse autoencoders with local encoding function, as well as to extend the LSC by removing the shift-invariance constraint, thus to have more expressive power.

[4] We have developed our own LCC code that resulted in rmse of 0.011 on swiss role clean dataset, but the code was unstable and therefore we didn't report the results in the main text.

References

[1] H. Lee, A. Battle, R. Raina, and A. Y. Ng, “Efficient sparse coding algorithms,” in *In NIPS*, pp. 801–808, NIPS, 2007.

[2] S. Serikov, *Neural network model of internal combustion engine*, vol. 46. Springer New York, 2010.

[3] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *SCIENCE*, vol. 290, pp. 2323–2326, 2000.

[4] H. Drucker, C. J. Burges, L. Kaufman, C. J. C, B. L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *IEEE Transactions on Neural Networks*, 1996.

[5] D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, and G. Camps-Valls, “Multioutput support vector regression for remote sensing biophysical parameter estimation,” *Geoscience and Remote Sensing Letters, IEEE*, vol. 8, pp. 804 –808, july 2011.

[6] K. Yu, T. Zhang, and Y. Gong, “Nonlinear learning using local coordinate coding,” *NIPS*, 2009.