Output-related feature representation for soft sensing based on supervised locality preserving projections

Xiaofeng Yuan, Yalin Wang, Chunhua Yang, Weihua Gui and Qingchao Jiang

Abstract—Locality preserving projections (LPP) is a useful tool for learning the manifold of high dimensional data, which is a linear approximation of nonlinear Laplacian Eigenmap (LE). However, the original LPP algorithm is an unsupervised method that extracts features without any reference to the output information. In this paper, a supervised LPP (SLPP) framework is proposed for output-related feature extraction in soft sensor applications. In the SLPP framework, the output information is utilized to guide the procedures for constructing the adjacent graph and calculating the weight matrix, with which the intrinsic structure of the data can be better described. Two specific SLPP algorithms are described. For performance evaluation of the proposed methods, experiments on a numerical example and an industrial ironmaking process are carried out. The results show the effectiveness of the proposed framework.

I. INTRODUCTION

To guarantee process security and improve process efficiency, process control and monitoring are becoming more and more important for modern industrial plants.[1, 2] These techniques are highly dependent on the timely and accurate measurement of key process variables. However, due to severe measurement environment, expensive analyzer cost and large measurement lag, these variables may not be measured accurately on time. Thus, soft sensors are utilized to provide frequent estimations of these difficult-to-measure variables through those easy-to-measure ones by some mathematical models.[3] Especially, data-driven soft sensors have been extensively researched and largely used during the past years since numerous of process data can be collected in modern industrial plants.[4, 5] A lot of soft sensor methods, like principal component regression (PCR)[6, 7], partial least squares (PLS)[8, 9], artificial neural network (ANN)[10, 11] and support vector machine (SVM)[12, 13] have been successfully applied to areas like chemical engineering, biochemical engineering, metallurgical industrial, and pharmaceuticals industry, etc.

Due to the largely installed measurement devices in modern industrial plants, process data often exhibit characteristics of high-dimensionality and information redundancy. Thus, it is necessary and useful to preprocess data by feature representation, which is also known as dimensionality reduction or feature learning. Then, the reduced features can be served as the new input variables for the regression model of the soft sensor. By representing high-dimensional process data through low-dimensional feature representation, it can make the model algorithms more stable and performance more accurate. By far, a lot of methods have been developed for dimensionality reduction. Among them, principal component analysis (PCA)[14] is a classic linear technique that projects the data along the directions of maximal variance. The main limitation of PCA is that it is designed to extract features by maintaining the global Euclidean structure of data while the local structure of data is not considered, which may make this approach noneffective in dealing with nonlinear data distributions. Thus, some manifold learning methods have been proposed to capture the local data structure, like Laplacian Eigenmap (LE)[15], locally linear embedding (LLE)[16] and Isomap[17]. However, most of those methods can only find the manifold for the training data while they are difficult to obtain the embedding for the testing data since no explicit mapping from the input space to the manifold space is given. This is also known as the out-of-sample problem. To overcome this drawback, He et al. proposed a method, named locality preserving projections (LPP)[18], to explore the intrinsic geometrical structure of data. Different from LE, LLE and Isomap, LPP is a linear algorithm for manifold learning. It can be regarded as the linear approximation of nonlinear LE. LPP has been successfully applied to areas like pattern recognition and process monitoring.

Nonetheless, LPP is completely unsupervised, in which only the input information is considered. The output information is discarded for feature representation. As a matter of fact, the output information is significant for supervised feature learning. In this way, a few supervised LPP (SLPP) algorithms[19, 20] were proposed for classification problems in order to utilize the class label information. For soft sensor modeling, little attention has been paid for supervised LPP in regression problems. In this paper, a new supervised LPP framework is proposed for feature representation in soft sensing area. In this SLPP framework, the output information is utilized in two aspects. First, To better construct the adjacent graph, both input and output are utilized to determine neighbors in LPP. Thus, full information is taken into account to identify the neighboring nodes. Second, the weights in SLPP are calculated by input as well as output information. Hence, the new weight is a combining index to make a trade-off between input and output information. By incorporating the output information into the feature learning procedure, the extracted features are more relevant to the output. Thus, they are more proper to construct the regression model and can make the prediction more accurate.
The rest of this paper is organized as follows. Section II gives a brief description of the unsupervised LPP algorithm. Then, the new supervised LPP framework is proposed in section III. Section IV demonstrates the effectiveness and flexibility of the proposed SLPP framework by two case studies. At last, some conclusions are reached in section V.

II. UNSUPERVISED LOCALITY PRESERVING PROJECTIONS

LPP is a linear approximation of the nonlinear Laplacian Eigenmap. The algorithm contains three steps: constructing the adjacent graph, calculating the weight and Eigenmaps. It is an example of the generic linear dimensionality reduction. Given a set of \( N \) input data points \( \mathbf{x}_i \in \mathbb{R}^n, i = 1, 2, \ldots, N \), represent them with low-dimensional points \( \mathbf{z}_i \in \mathbb{R}^m, i = 1, 2, \ldots, N \) by linear transformation \( \mathbf{z}_i = \mathbf{A}^T \mathbf{x}_i \), where \( \mathbf{A} \in \mathbb{R}^{m \times n} \) is the transformation matrix.\(^{[15]}\) In LPP, the feature representation procedure is obtained by the following steps.

A. Constructing the adjacent graph

Let \( 
\mathbf{G} \) denote a graph with \( N \) nodes for the data set, with its \( i \)th node denoting the \( i \)th data sample \( \mathbf{x}_i \). If sample \( \mathbf{x}_i \) and \( \mathbf{x}_j \) are identified as neighbors, then an edge will be put between the \( i \)th and \( j \)th nodes. There are many criteria to define the neighbors. For example, the \( K \) nearest neighbors (KNN) is most widely used. In KNN, if \( \mathbf{x}_i \) is one of the \( K \) nearest neighbors of \( \mathbf{x}_j \) and \( \mathbf{x}_j \) is also among the \( K \) nearest neighbors of \( \mathbf{x}_i \), an edge will be put between node \( i \) and node \( j \). The distance between samples in input space, \( d_{i,j} = \| \mathbf{x}_i - \mathbf{x}_j \| \) is used to determine nearest neighbors.

B. Calculating the weight

After the construction of the adjacent graph, we need to weight the edges. Let \( \mathbf{W} \) be the weight matrix, with \( \mathbf{W}_{ij} \) denoting the weight of the edge connecting node \( i \) and \( j \). If there is no such edge, \( \mathbf{W}_{ij} \) will be set as 0. Otherwise, the weight can be determined by the heat kernel

\[
\mathbf{W}_{ij} = e^{- \frac{d_{ij}}{\sigma}}
\]  

(1)

where \( \sigma \) is a tuning parameter that controls the decreasing rate of the weight with the distance. Thus, \( \mathbf{W} \) is a sparse symmetric \( N \times N \) matrix.

C. Find the optimal Linear Embedding\(^{[18]}\)

In order to obtain the optimal linear low-dimensional embedding, LPP preserves the local neighborhood information in a certain sense. That is to say, the neighbor information in original high-dimensional space will be preserved in the low-dimensional subspace. Hence, when mapping the weighted adjacent graph to a new line, connected points should also stay as close together as possible. Assume \( \mathbf{a} \) is such a transformation vector. \( \mathbf{z}_i \) is the corresponding mapped point of \( \mathbf{x}_i \) to line \( \mathbf{a} \). Hence, \( \mathbf{z}_i = \mathbf{a}^T \mathbf{x}_i \).

To keep the local neighborhood information, a reasonable criterion for obtaining the mapping vector is to minimize the following objective function under appropriate constraints.

\[
J = \min \frac{1}{2} \sum_{i,j} (z_i - z_j)^T \mathbf{W}_{ij}
\]  

(2)

By setting such an objective function, it tries to guarantee that if \( \mathbf{x}_i \) and \( \mathbf{x}_j \) are close, then their mapped points \( \mathbf{z}_i \) and \( \mathbf{z}_j \) can also stay close. Let \( \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \) and \( \mathbf{z} = [z_1, z_2, \ldots, z_N] \). Then we have \( \mathbf{z} = \mathbf{a}^T \mathbf{X} \). By some mathematical operations, The objective function can be rewritten as

\[
J = \frac{1}{2} \sum_{i,j} (z_i - z_j)^T \mathbf{W}_{ij} = \frac{1}{2} \sum_{i,j} (\mathbf{a}^T \mathbf{x}_i - \mathbf{a}^T \mathbf{x}_j)^T \mathbf{W}_{ij}
\]  

(3)

\[
= \mathbf{a}^T (\mathbf{X} (\mathbf{D} - \mathbf{W}) \mathbf{X}^T) \mathbf{a} = \mathbf{a}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{a}
\]  

where \( \mathbf{D} \) is a diagonal matrix with its diagonal entries \( \mathbf{D}_{ij} = \sum_{j} \mathbf{W}_{ij} \), in which \( \mathbf{D}_{ij} \) is the sum of the \( i \)th column of the \( \mathbf{W} \). \( \mathbf{D}_{ij} \) is a index to reflect the importance of \( z_i \). Therefore, the constraint is imposed as

\[\mathbf{z}^T \mathbf{D} \mathbf{z} = \mathbf{a}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{a} = 1\]

(4)

At last, by optimizing the objective function (3) under constraint (4), the transformation vector \( \mathbf{a} \) can be obtained from the following generalized eigenvalue problem

\[\mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{a} = \mathbf{a}^T \mathbf{X} \mathbf{D} \mathbf{X}^T \mathbf{a}\]

(5)

where \( \mathbf{a} \) is the eigenvector corresponding to the smallest eigenvalue of the above eigen decomposition problem.

It is easy to show that the columns of the transformation matrix \( \mathbf{A} \), \( \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m \), are the \( m \) eigenvectors corresponding to the \( m \) least eigenvalues of (5).

III. SUPERVISED LPP

The original LPP is completely unsupervised, which cannot guarantee that the extracted features are related to the output variables. Thus, they may not be appropriate to be used for new inputs in sequential regression model. To alleviate this problem, a supervised LPP framework is proposed in this section. Then, we will give two specific SLPP models.

---

Fig. 1. The proposed SLPP framework
As can be seen in section II, the original LPP mainly consists of three steps: constructing the adjacent graph, calculating the weight matrix and finding the optimal linear embedding. It is unsupervised because only the input information are utilized to carry out the first two steps. However, the output information is not used. In this way, the learned features cannot guarantee to have desired regression properties. It would be more helpful to utilize the output information to guide the projection procedure for the input data. Under this guidance, the output information is incorporated to construct the adjacent graph and calculate the weight matrix in the SLPP. Fig. 1 gives the basic schematic diagram of the supervised LPP framework. Suppose the \( N \) output data points are \( y_i, i = 1, \ldots, N \).

A. SLPP1

For SLPP1, the output information is utilized in the following way. First, the input and output variables are combined to form a new variable \( t_i = [t_{x}, t_{y}] \) in the complete variable space. To construct the adjacent graph, the following distance are used to determine the \( K \) nearest neighbors.

\[
d_{i,j} = \|t_i - t_j\| \tag{6}
\]

Different from the original LPP, which calculate the distance only in the input space, SLPP1 determine neighbors by taking account of input and output information simultaneously. This is more reasonable since only those points that are close in the whole variable space are the expected neighbors. After the construction of the adjacent graph using the new distance, we can define the new weights for this graph. Similarly, the new weights are defined on the whole variable space

\[
W_{i,j} = \frac{e^{rac{d_{i,j}}{\sigma}}}{\sum_{k \in N_i} e^{rac{d_{i,k}}{\sigma}}} \quad \text{if } t_i \text{ and } t_j \text{ are identified as } K \text{ nearest neighbors of each other} \tag{7}
\]

After calculation of the weight matrix, the optimal Linear Embedding can be obtained by (5).

B. SLPP2

The first method can include the output information to construct the adjacent graph. However, it cannot change the relative importance of input or output distance. Hence, another approach can be used alternatively. In this method, the input and output distances are first calculated separately, then they are combined to form the final distance by weighting average, as shown in the following equation.

\[
\begin{align*}
\hat{d}_{x,i,j} &= \|x_i - x_j\| \\
\hat{d}_{y,i,j} &= \|y_i - y_j\| \\
\hat{d}_{i,j} &= \lambda \hat{d}_{x,i,j} + (1 - \lambda) \hat{d}_{y,i,j}
\end{align*} \tag{8}
\]

Then, the weight can also be defined as

\[
W_{i,j} = \begin{cases} \\
\frac{e^{rac{d_{i,j}}{\sigma}}}{\sum_{k \in N_i} e^{rac{d_{i,k}}{\sigma}}} & \text{if } t_i \text{ and } t_j \text{ are identified as } K \text{ nearest neighbors of each other} \\
0 & \text{otherwise}
\end{cases} \tag{9}
\]

where \( \lambda \) is a tuning parameter to change the relative importance of weights between input and output space. If \( \lambda = 1 \), this is the same with the original unsupervised NPE. If \( \lambda = 0 \), only the output information is employed to construct the adjacent graph. If \( 0 < \lambda < 1 \), both input and output information are utilized to determine neighbors. In order to give sufficient importance on the output information, a good choice of parameter \( \lambda \) is around 0.5.

IV. SLPP-BASED SOFT SENSOR

The supervised LPP algorithm can be directly used for feature extraction in soft sensor applications. Suppose the input and output training data are \( X_{\text{train}} \) and \( Y_{\text{train}} \). First, the supervised LPP model will be constructed on the training data \( X_{\text{train}} \), and the transforming matrix \( A \) will be obtained. In this way, both the training data and testing data can be projected to the manifold space as

\[
\begin{align*}
Z_{\text{train}} &= A^T X_{\text{train}} \\
Z_{\text{test}} &= A^T X_{\text{test}}
\end{align*} \tag{10}
\]

Then a least square regression model will be modeled between the output \( Y_{\text{train}} \) and extracted feature \( Z_{\text{train}} \) as

\[
Y_{\text{train}} = \theta^T Z_{\text{train}} \tag{11}
\]

At last, the output prediction for the testing data is

\[
\hat{Y}_{\text{test}} = \theta^T Z_{\text{test}} \tag{12}
\]

V. CASE STUDIES

In this section, two case studies are carried out to test the validation of the proposed SLPP. One is a numerical example and another one is an industrial process application. To validate the superiority of the proposed framework, three methods are used for feature extraction in both studies, which are LPP, SLPP1 and SLPP2. The extracted features will be used as the input for the least squares regression model in order to predict the output variable. The root mean square error (RMSE) is employed as the index to indicate the prediction performance.

A. Numerical example

In this part, a numerical example is simulated to test the performance of the proposed S-NPE algorithm. The data are set as
\[ a \sim U(0,1); \quad b = \frac{3\pi}{2} (1 + 2a) \]
\[ x_i = b \times \cos b; \quad x_2 = 21 \times U(0,1); \quad x_3 = b \times \sin b \]
\[ x_4 = 21 \times N(0,1), i = 4, 5, 6 \] (13)

where \( U(0,1) \) represents uniform distribution between range 0 and 1, and \( N(0,1) \) stands for Gaussian distribution with mean of 0 and variance of 1. In this example, three input variables \( (x_1, x_2, x_3) \) are used in the setting of the Swiss roll data, which exhibits characteristic of high nonlinearity. Additionally, three other variables generated from the normal distribution are added. The output is defined as
\[ y = \frac{x_1}{10} + \frac{x_1^{1/3}}{5} \] (14)

For model construction, 500 data points are generated. Half of the data are used for training and the remaining for testing. First, several parameters should be decided. The number of nearest neighbors \( K \) is set as 5, which can preserve the most similar neighbors in the adjacent graph. Parameter \( \sigma \) in the weight function is decided as 1. In SLPP2, parameter \( \lambda \) is chosen with a proper value of 0.5 in order to place more importance on the output information. To effectively extract the least features for regression model, the number of extracted features is two since the output variable is only related to only two variables. Then, the prediction results of the three methods are shown in Table I.

<table>
<thead>
<tr>
<th>Method</th>
<th>LPP</th>
<th>SLPP1</th>
<th>SLPP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>0.3096</td>
<td>0.2059</td>
<td>0.0914</td>
</tr>
</tbody>
</table>

From the comparative results list in Table I, it is easy to see that LPP gives the worst prediction results since its RMSE value is the biggest. This is because LPP is an unsupervised dimensionality reduction method. It can only extract features by considering the neighboring information of input space. However, the features extracted in this way may not be related to the output prediction. Even, certain features may be irrelevant to the output variable. Different from that, the adjacent graph and weight matrix are constructed and calculated by the input additionally with output information in SLPP1 and SLPP2. Thus, the features are extracted in a supervised manner. They can improve the relevance of features with the output variable. Using these features for soft sensor regression modeling, they can improve the prediction accuracy. This can be seen from the results shown in Table I. Moreover, since SLPP2 plays more importance on the output information when carrying out feature projection, the features are more related with the output than those of SLPP1. Therefore, SLPP2 can further improve the prediction performance than SLPP1.

Specifically, the detailed prediction results of the three methods on the testing samples are figured in Fig. 2. It can be seen that the two supervised LPP methods can match better with the real output line than unsupervised LPP.

**B. Blast furnace ironmaking process**

The blast furnace is an important plant for ironmaking process. This process has been considered in literature[21]. The description of the process is revisited below. The main purpose of this process is to transfer iron oxides into liquid iron, which is also known as hot metal. It is a complex industrial process due to its in-furnace complex chemistry. Under the high temperature and pressure, the severe measuring circumstance makes online measurement of some process variables very difficult. So sensor is a useful tool to obtain prediction of key process variables for process control and monitoring. Typically, the process is divided into five different parts from the top to bottom, which are lumpy zone, cohesive zone, active coke zone, hearth-deadman and tuyere raceways. Fig. 3 shows a flow chart of the blast furnace ironmaking process[21]. First, the solid iron ore is fed into the sintering machine to get sintered ore. Also, the coking coal is thrown into the coke oven to be coked. Then, they are charged into the furnace from its top layer by layer. Meanwhile, heated air is injected together with pulverized coal or oil through the tuyeres at the bottom. With the proceeding of ironmaking, the solid materials gradually descend and the heated air goes upward. In this process, CO is generated due to oxidation and then the ore is reduced. After that, the final product is formed and hot metal is accumulated in the hearth. Meanwhile, slag mainly consisted of CaO and SiO₂ is produced as a by-product floating on the top of hot metal. In this stage, the main chemical reactions are

\[ 2C + O₂ \rightarrow 2CO \]
\[ Fe₂O₃ + 3CO = 2Fe + 3CO₂ \]
\[ CaCO₃ = CaO + CO₂ \]
\[ CaO + SiO₂ = CaSiO₃ \] (15)

The hot metal is the immediate product of the ironmaking process. The quality of the hot metal is an important factor to evaluate wellness of the reaction. Especially, the silicon content in the hot metal is an important index indicating the quality of hot metal. So timely measuring of the silicon is necessary. But as we mentioned before, the environment for on-line measurement is severe and the off-line analysis has a
large time lag. Thus soft sensor is utilized here to estimate the silicon content via 10 easy-to-measure variables.

![Image of the blast furnace ironmaking process](image_url)

Fig. 3. The flowchart of the blast furnace ironmaking process[21]

There are totally 154 samples. To build the soft sensor model, the first half samples are used for model construction, and the rest half samples are used as the validation data set. Similar with the numerical example, the number of nearest neighbors K is set as 5, which can preserve the most similar neighbors in the adjacent graph. Parameter in the weight function is decided as 1. In SLPP2, parameter is chosen with a proper value of 0.5 in order to place more importance on the output information. And the number of latent variables are chosen as 3. The prediction results on this process are tabulated in TABLE II.

<table>
<thead>
<tr>
<th>Method</th>
<th>LPP</th>
<th>SLPP1</th>
<th>SLPP2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>0.2171</td>
<td>0.2152</td>
<td>0.2124</td>
</tr>
</tbody>
</table>

From the above results, it can be seen that the prediction accuracy of LPP is the worst among the three methods since it is an unsupervised feature extraction method. It cannot guarantee the features are related to the output. Hence, the extracted features may be not appropriate for output prediction. On the contrary, SLPP1 and SLPP2 can utilize the output information to guide feature extraction procedure. It can improve the relevance of the extracted feature with the output. Hence, they can improve the prediction performance than LPP.

![Image of detailed prediction results](image_url)

Fig. 4 Detailed prediction results of the three methods on the ironmaking process (red: real values; blue: predicted values)

Furthermore, the detailed prediction results on the testing samples are provided in Fig. 4. It is easily seen that the predicted output lines of SLPP1 and SLPP2 can track better with the real output line. This can also show the superiority of the supervised LPP framework.

VI. Conclusion

In this paper, a new supervised locality preserving projections framework is proposed for feature extraction in soft sensor regression problems. In the SLPP framework, the output information is used to guide the learning of the manifold of data in the adjacent graph and weight calculation steps. Two specific SLPP algorithms are proposed to utilize the output information. Experiments are carried out on a numerical example and an industrial debutanizer column example to compare prediction performance between the unsupervised LPP and the two SLPP methods, the results have shown the superiority of the proposed methods. Beside the two proposed SLPP algorithms, there are other techniques that can utilize the output information to construct SLPP methods.

REFERENCES


