Chapter 5
Nonparametric Bayesian Method for Robot Anomaly Diagnose

Abstract In this chapter, we introduce two novel anomaly diagnose methods using the Bayesian nonparametric hidden Markov models when anomaly triggered, including i)multi-class classifier based on nonparametric models, ii) sparse representation by statistical feature extraction for anomaly diagnose. Additionally, the detail procedure for anomaly sample definition, the supervised learning dataset collection as well as the data augmentation of insufficient samples are also declared. We evaluated the proposed methods with a multi-step human-robot collaboration objects kitting task on Baxter robot, the performance and results are presented of each method respectively.

5.1 Introduction

In addition, anomaly diagnoses refers to the process of clearly classifying anomalies using supervised learning on the premise that the robot detects the occurrence of anomalies. Due to the uncertainty, diversity and unpredictability of anomalies, the robot's anomaly diagnoses can effectively improve the safety performance of the robot system and provide guarantee for subsequent anomaly repair behaviors. Therefore, the online abnormal monitoring and diagnoses of robots is the focus and difficulty of robots' long-term autonomous operation.

5.2 Related Work

The robot introspection not only needs to implement the movement identification and monitoring of anomalies, but also to diagnose the types of anomalies for providing sufficient support in subsequent recovery. The samples used for anomaly diagnoses in this chapter would have the same modal information as in anomaly monitoring and belong to multidimensional time series. Since the problem of multidimensional time series classification is a supervised learning problem [15–17], which aims to determine the labels of multidimensional time series of the same structure

X. Zhou et al., *Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection*, https://doi.org/10.1007/978-981-15-6263-1_5

and length, Orsenigo et al. [18] proposed a discrete support vector machine (Discrete Support Vector Machine) Machine (DSVM) classification method. This method benefits from the concepts of warping distance and softened variable margin and realizes the classification of multidimensional time series. Lee et al. [19] proposed a time series classification method based on K-Nearest Neighbor [20] (K-Nearest Neighbor, KNN), which was successfully used to evaluate the traffic prediction application of the mobile telecommunications industry. Seto et al. [21] combined dynamic time warping [8, 22] (DTW) and KNN to derive a multivariate time series classification based on dynamic time warping template selection, and applied it to human activity recognition. In addition to the above-mentioned distance-based multidimensional time series classification method, the method of establishing feature vectors through dimensionality reduction methods has also received attention [23–25]. Nanopoulos et al. [26] proposed a method based on statistical feature extraction to establish feature vectors, and then trained a multi-layer perceptron (MLP) from the feature vectors and target categories to achieve the classification of time series. Utomo et al. [27] proposed to classify the multidimensional data of the hospital intensive care unit by the method of multidimensional compression description (MultiCoRe), which extracted the features including time domain and frequency domain for classification. Jaakkola et al. [28] introduced a method combining HMM and SVM for classifying protein domains. This method can also be applied to other fields of biological sequence analysis. Raman et al. [29] proposed the use of a multi-layer HDP-HMM modeling method to achieve classification of human movements, training HDP-HMM from all movement motion samples, and building a multi-objective classifier from this model. Test the size of the log-likelihood function of the specimen to achieve classification. Dilello et al. [30] used sHDP-HMM to classify robots with multiple modal anomalies. The model assumes that the observations are independent of each other, which weakens the correlation between abnormal data to some extent. Although the training of neural networks with small samples is easy to cause overfitting and difficult to achieve online monitoring, but because of its better modeling capabilities and less data pre-processing process, multivariate time based on deep learning Sequence classification is currently being extensively studied by scholars [31–34].

Besides the aforementioned that multivariate time series classification has received significant interest in areas such as healthcare, object recognition, and human action recognition [35–37]. Meanwhile, most of the solutions to the classification problem of multidimensional time series are solved by supervised learning, that is, the multi-objective classifier learning is performed by manually labeled samples in advance by humans. In view of this, the anomaly diagnoses problem considered in this chapter is mainly for the classification of conventional anomaly samples with multimodel observation of robotic systems. In robotics, complex semi-autonomous systems that have provided extensive assistance to humans, anomalies are occur occasionally [39]. For this reason, enabling accurate, robust, and fast anomaly monitoring and diagnoses in robots has the potential to enable more effective, safer, and more autonomous systems [38, 49]. Execution monitoring, especially with the focus of detecting and classifying anomalous executions has been well studied in robotics [40]. Daehyung Park et al.[39] introduced a multimodal execution monitoring system to detect and

classify anomalous executions for robot-assisted feeding. Specifically, the classifier labelled not only the anomaly type but also the cause of the anomaly by using an artificial neural network system. The neural net successfully integrated the anomaly monitoring and diagnoses for assisting a person with severe quadriplegia. However, due to the HMM anomaly detector limitations, the classified accuracy for anomaly types and causes are 90.0%, 54.0% respectively. In [41], Bjreland et al.introduced a monitoring system that also can detect, classify, and correct anomalous behaviors using a predictive model. Those two integrated system give us a lot of inspiration and confidence for improving the robot introspection with the anomaly monitoring and diagnoses.

The anomaly diagnoses has been used to determine the source of anomalies while running manipulator or mobile robots [42]. Several common time series classification algorithms are distance based metrics, such as the k-nearest neighbors (KNN) approach have proven to be successful in classifying multivariate time series [43]. Plenty of research indicates Dynamic Time Warping (DTW) as the best distance based measure to use along KNN [44]. Besides the distance based metrics, the feature based algorithms have been used over the years [45], which rely heavily on the features being extracted from the time series data or modeling the time series with the parametric methods. Hidden State Conditional Random Field (HCRF) and Hidden Unit Logistic Model (HULM) are two successful feature based algorithms that have led to state of the art results on various benchmark datasets [46]. However, HCRF is a high computational efficiency algorithm that detects latent structures of the input time series using a chain of k-nominal latent variables. Further, the number of parameters is linearly increasing of latent states required. To overcome this, HULM proposes using H binary stochastic hidden units to model 2^H latent structures of the data with only $O(H)$ parameters. Another approach for multivariate time series classification is by applying dimensional reduction techniques or by concatenating all dimensions of a multivariate time series into a univariate time series [47].

5.3 Problem Statement

Another main task of robotic introspection is to accurately evaluate the potential pattern of real-time multi-modal sensing data. It is not limited to the identification of robot movement and abnormal monitoring, but also has the ability to classify conventional abnormalities. Anomaly diagnoses mainly refers to the identification of conventional learned abnormal types through supervised learning after the robot detects the occurrence of abnormalities. An intelligent robot system includes but is not limited to the following types of abnormalities: system abnormalities, internal sensors anomalies, motion instruction failures, noise or damage to sensors and actuators, robot-human-environment interactions, or external disturbances in the environment. Park et al. [50] carried out a diagnoses of the types of robot anomalies and their causes, and integrated this function into a human-robot interactive robot system for feeding disabled people. The types of anomalies considered in this section mainly

come from external disturbances caused by system anomalies (kinematic anomalies, sensor failures, etc.) or human improper interference behavior or robot ends in a human-machine collaboration environment. In order to improve the compactness and portability of the proposed robotic sensing system, this section still adopts the method of non-parametric Bayesian hidden Markov model to implement multi-class anomaly diagnoses.

5.4 Collection and Augmentation of Anomaly Sample

The collection of anomalous samples in this book takes into account the precursory period and the duration of a certain period of time when the anomaly occurs, which is determined by the value of the given anomalous window range win_len (for ease of adjustment, the duration of the precursory period and duration is usually equal, both are half of win_len). As shown in Fig. 5.1, a sample with an anomaly type of "tool collision" is extracted (light red background area) around the trigger time (black vertical dashed line) of the previous anomaly monitor.

In addition, due to the great randomness and uncertainty of the occurrence of anomalies, the number of samples between different types of anomalies is extremely unbalanced, and for some anomalies (collision between the robot and the environment), the impact on the robot body may even cause damage There is no guarantee

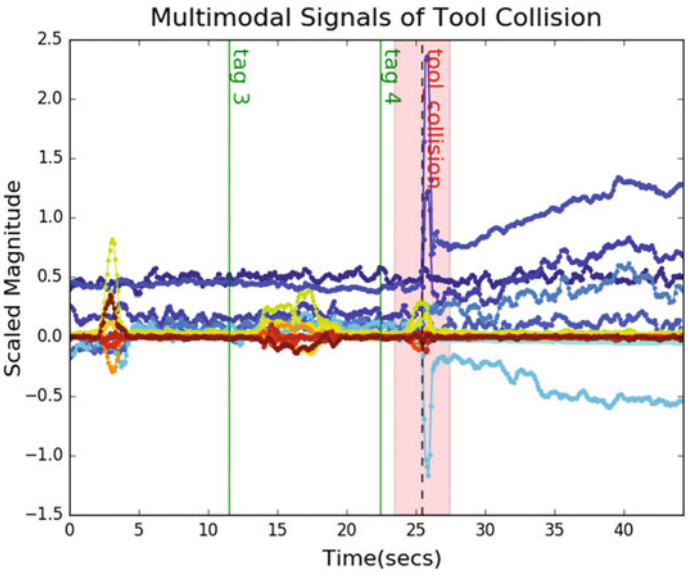

Fig. 5.1 An example of extracting an "tool collision" anomaly sample with a given when anomaly detected

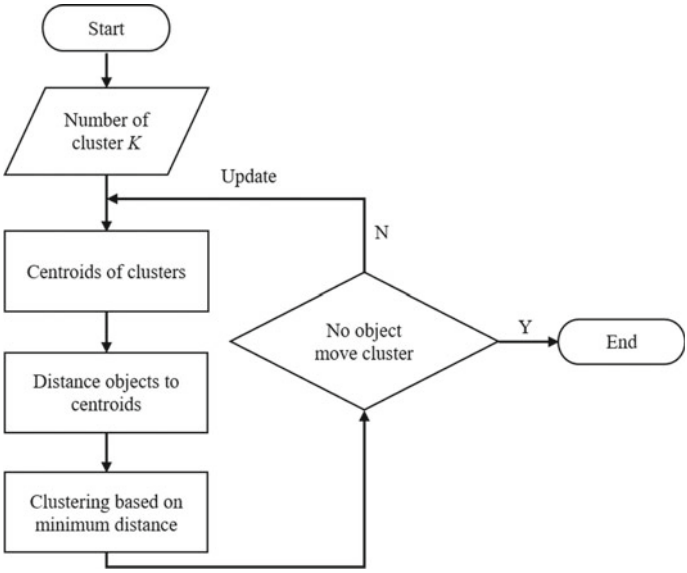

Fig. 5.2 The implementation flowchart of K-means Algorithm

for the number of samples for model training. In view of this problem, this article considers the data enhancement processing for the abnormal types of fewer samples collected. The main goal is to generate synthetic data based on the statistical characteristics of the data.

The proposed multimodal augmentation method is inspired by the $K-means$ clustering method in traditional machine learning, combined with Dynamic Time Warping (DTW) to enhance the data. Among them, the implementation flowchart of the $K-means$ clustering method is shown in Fig. 5.2.

In the algorithm, K represents the number of categories, Means represents the mean. As the name suggests, this method is a method for Dimensional or multidimensional data points, and even the time series considered in this article) clustering algorithm, the core idea is to use a preset K value and the initial centroid of each category (Centroid) to distance (Euclidean distance, Manhattan distance and Time series distance measure (DTW) is used to cluster the data points that are closer, so that the average value of the clustered iteration is preferentially obtained with the smallest objective function value, as shown in the following formula:

$$J=\sum_{j=1}^{K}\sum_{i=1}^{N}||x_i^{(j)}-c_j||^2 \tag{5.1}$$

where, the symbol J is the objective function of the $K-means$ algorithm; K is the number of clusters; N is the number of data points; $||\cdot||$ is expressed as the distance

Algorithm : Data Augmentation for Sparse Anomaly Class

Input: $\{y_n\}_n^{N_c}$: Sparse dataset with N_c data-points (sequences), each of length T_n;
N: Number of data-points for class $c \in \mathcal{C}$;
N_b: For determining the number of centroids for K-means clustering
N_k: Number of the iteration of K-means;
N_d: Number of runs for DTW averaging algorithm;
K: Number of centroids for K-means clustering
Result: Synthetic anomaly sample generation: $\{\hat{y}_s\}_n^{N_s}, N_s > N_c$

```
K ← ceil(|N|/N_b). ceil(x): The smallest integer greater than or equal to x
while N_k > 0 do
    CENTROIDS ← pick K random data points from {y_n}_n^{N_c};
    Assign each data point y_n to the nearest CENTROID using DTW;
    Remove CENTROIDS with only one data-point;
    for CENTROID in CENTROIDS do
        CENTROID ← update the centroid using the subset of data-points which are assigned to centroid;
    end
    {ŷ_s}_n^{N_s} ← append the CENTROIDS
    N_k− = 1
end
```

Fig. 5.3 The pseudocode for data augmentation for sparse anomaly class

function between the data points and the centroid; and c represents the centroid of the category. From the implementation of the $K - means$ algorithm, we know that by clustering data points close to the centroid, and data points in the same category have similar statistical characteristics, the goal of satisfying data enhancement is to hope to generate similar data based on sparse dataset New data for characteristics. To this end, the proposed algorithm, after initializing K random centroids, loops through the following two steps: (1) Dividing. Divide each data point to the nearest centroid according to the distance measurement of DTW; (2) Update. According to the selected mean measurement method, the centroid is moved to the center of various types of designated data points. This time, each time the updated centroid is used to generate a composite data, the algorithm's pseudo code is shown in Fig. 5.3.

Meanwhile, an example of one-dimensional data enhancement for an abnormal type is shown in Fig. 5.4. As can be seen from Fig. 5.4, the proposed algorithm can effectively capture the statistical (such as peak and mean) characteristics of the original data.

5.5 Anomaly Diagnose Based on Nonparametric Bayesian Model

The anomaly diagnoses is triggered once an anomaly is detected. A system can possibly address a wide variety of types of anomalies including low-level hardware anomalies: sensor and actuator noise or breakage; mid-level software contingencies like: logic errors or run-time exceptions; high-level misrepresentations: poor modeling of the robot, the world, their interactions, or external disturbances in the

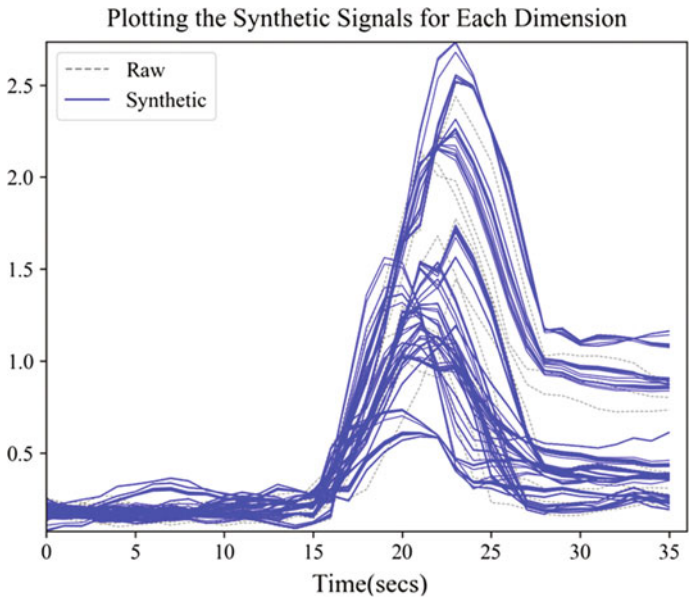

Fig. 5.4 An example for illustrating the data augmentation of anomaly sample

environment). In [48, 50], Park et al.identified both the anomaly class and the cause. In this work, we deal with anomalies caused by external disturbances generated either by intrusive human behavior or resulting from poor modeling or anticipatory ability on the robot's end.

5.5.1 Multiclass Classifier Modeling

We consider the problem of multiclass classification based on multivariate time series, that is, to find a function $f(y^n) = c^n$ given the $i.i.d$ training data $Y = \{y^n\}_{n=1}^N$, $C = \{c^n\}_{n=1}^N$, where $y^n = [y_0^n, y_1^n, ..., y_{T_n}^n]$ is an observation sequence and $c^n \in \{1, .., c\}$ its corresponding anomaly class. An observation $y_t^n \in \mathbb{R}^d$ consists of the same multimodal features as described in Sect. 4.6.1 at time-step t. Our objective is diagnoses, where given a new test observation sequence $\hat{y}$, we have to predict its corresponding anomaly class $\hat{c}$. Here, the $\hat{y}$ is recorded by considering a $window\_size$ around the anomaly occurred moment, and the $\hat{c}$ is labeled manually during training procedure. We represent each anomaly class by a separate sHDP-VAR-HMM as outlined in Sect. 4.6.1, the Θ_c are the parameters for class c. It would be straightforward to estimate the posterior density of parameters $p(\Theta_c)|Y, C)$ by

Algorithm 1: Training model for anomaly diagnosis

Input:
N_c: Number of sequences for class $c \in \mathcal{C}$;
$\{y_n\}_n^{N_c}$: Dataset with N_c sequences, each of length T_n;
N_i: Number of the maximum iteration for learning;
N_r: Number of runs for the whole learning;
random_state: The random number generator;
k_splits: Number of folds;
a, b, d, e: Hyper-prior for concentration parameters;
ν, Δ, V, M, s_F: Hyper-prior for the MNIW distribution;
κ: The self-transition bias;
K: The truncation active states.
Result: sHDP-VAR-HMM models for each class
$\{y_n\}_{train}, \{y_n\}_{test}$ = KFold_split($\{y_n\}_n^{N_c}, k\_splits, random\_state$)
for *k in k_splits* **do**
  for *i in* N_r **do**
    for *n in* N_i **do**
      if *not converged* **then**
        Θ_c, *loss* = HDPHMM($\{y_n\}_{train}, a, b, d, e, \nu, \Delta, V, M, s_F, \kappa, K$)
      else
        sHDP-VAR-HMM with Θ_c and *loss*
      end
    end
    $\{loss\}_i^{N_r} \leftarrow loss$
    if $i == N_r$ **then**
      $\Theta_c \leftarrow$ with the minimum loss value
  end
  $\mathcal{L}_{test\_mean} = \frac{\sum_{i=1}^{N_{test}} p(\{y_n\}_i|\Theta_c)}{N_{test}}, i \in N_{test}$
  $\{\mathcal{L}_{test\_mean}\}_k^{k\_splits} \leftarrow \mathcal{L}_{test\_mean}$
  if $k == k\_splits$ **then**
    return Θ_c with the maximum $\mathcal{L}_{test\_mean}$
end

Fig. 5.5 An example for illustrating the progress for training nonparametric Bayesian model for anomaly diagnoses

$$p(\Theta_c|Y, C) = p(\Theta_c) \prod_{c^n=c} p(y^n|\Theta_c) p(c). \tag{5.2}$$

That is, each sHDP-VAR-HMM is trained separately and a conditional density $p(y|c)$ for each class is trained throughout the process as defined in Fig. 5.5.

5.5.2 Experiments and Results

- **Anomaly Dataset in Kitting Experiment**

The dataset captures sensory-motor signatures of the Kitting task under anomalous scenarios as outlined in this paper. A total of 136 anomalous events were recorded in 142 experimental executions across all skills. The proportions for each anomaly class were as follows: TC 15.7%, HC 16.7%, HCO 16.7%, NO 13.0%, OS 16.7%, WC 15.7%, and OTHER 5.6%. To intuitively explain, analyze, and propose corresponding diagnose methods, all the anomalies identified using our proposed method are taken into further consideration. That is, we extract all the anomaly data points from each abnormal movement and concatenate them with labels, in which the anomalous data point is also restricted with the same features as considered in anomaly identification. Those identified anomalies are visualized via the t-distributed Stochastic Neighbor Embedding (t-SNE) method [51] and labelled manually, as shown in Fig. 5.6.

- **Anomaly Diagnoses Window Considerations and Online Recording**

Note that when an anomaly monitoring is flagged, we consider a window of time duration $\pm window\_size$ for the subsequent anomaly diagnoses.[1] In cases where an anomaly is detected towards the beginning of a skill execution, and the duration of existing data prior to the detection is less than our $window\_size$. In this work, we do not extract data for diagnoses as we deem a minimal presence of signals before and after the detection crucial for the diagnoses. The sensory-motor data collected at this stage, allows to build basic models of the anomalies described in Chap. 4.

Our system recorded sensory data online. Upon detection of an anomaly, the time-step at which the flag occurred is recorded. Then, we record (online) the multimodal signatures (F/T, Cartesian velocity, and Tactile signals) before and after the anomaly (also referred to as the sample) according to the $window\_size$. Signals were re-sampled at 10 Hz to achieve temporal synchronization for all modalities and the further preprocessed as described in preprocessing step.

- **Parameter Settings and Results**

To avoid overfitting, we performed *3-fold* cross validation for each anomaly type separately. An 18 dimensional feature vector was used as presented in Equation 4.53. We compare against five baselines consisting of parametric HMMs (with a fixed number of hidden states), various observation models, and various variational inference methods. The training and testing dataset for all the diagnoses methods was the same in this comparison.

(1) Parametric HMM Settings: We train 3 differing types of HMMs for each anomaly class. Each HMM uses four different numbers of hidden states $K \in \{3, 5, 7, 10\}$ to train each class. We need to estimate the transition matrix, mean, and covariance parameters. To this end, K-Means++ [52] clusters the data and yields

[1] We use the Redis database for this purpose (https://redis.io/) as rosbags can only be processed offline.

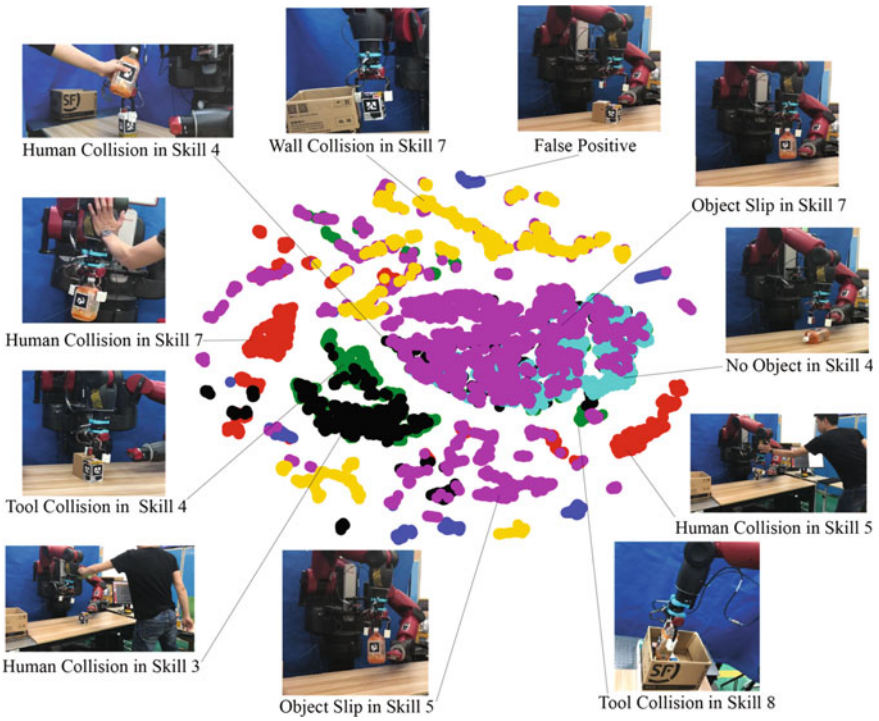

Fig. 5.6 Visualization of all the identified anomalies using t-SNE method. We heuristically label the clusters according to the analysis of unexpected anomalies and the data points density in such a kitting experiment

initial estimates. During testing, we evaluate a test sample against all classes and select the class with the largest log-likelihood. The parametric HMMs result summary is presented in Table 5.1.

∘ *HMM-Gauss-EM:* A classifier based on a classical HMM with Gaussian observations was trained independently for each anomaly class. The standard Baum-Welch Expectation Maximization algorithm [53] was used to learn the HMM model.

∘ *HMM-Gauss-VB:* A classifier based on classical HMM with Gaussian observation was trained independently for each anomaly class. The standard Variational Bayesian (VB) inference algorithm [54] was used to learn the HMM model.

∘ *HMM-AR-VB:* A classifier based on a classical HMM with first-order autoregressive Gaussian observations was trained independently for each anomaly class. The standard VB inference algorithm [54] was used to learn the HMM model.

In conclusion, we find that for parametric HMMs (i) the best diagnoses accuracy rate was 95.7% when using 5–7 fixed states; (ii) Variational inference algorithms outperformed the standard EM algorithm; (iii) The Autoregressive observation model classified better than the Gaussian model due to it's linear conditional nature; (iv) Parametric HMMs, in general, are less effective to jointly model the dynamics of the

Table 5.1 The diagnoses accuracy of both classical parametric HMMs and non-parametric HMMs. The best model (observation model and number of hidden states) for each anomaly type is highlighted in blue

States	Methods	TC	HC	HCO	NO	OS	WC	OTHER	Total Accuracy
3	HMM-Gauss-EM	1.0	0.89	0.89	0.78	0.83	1.0	0.67	0.876
	HMM-Gauss-VB	1.0	0.94	0.95	0.96	0.88	1.0	0.67	0.915
	HMM-AR-VB	1.0	0.89	0.90	0.93	0.94	1.0	0.67	**0.929**
5	HMM-Gauss-EM	1.0	0.78	0.84	0.79	0.79	1.0	0.67	0.838
	HMM-Gauss-VB	0.94	0.94	0.89	0.86	0.85	1.0	0.67	0.889
	HMM-AR-VB	1.0	0.89	0.95	1.0	0.98	1.0	0.67	**0.957**
7	HMM-Gauss-EM	1.0	0.83	0.95	0.71	0.77	0.83	0.67	0.826
	HMM-Gauss-VB	1.0	0.83	0.86	0.79	0.82	1.0	0.67	0.863
	HMM-AR-VB	1.0	0.95	1.0	0.93	0.96	1.0	0.67	**0.957**
10	HMM-Gauss-EM	0.94	0.72	0.85	0.86	0.79	0.94	0.67	0.826
	HMM-Gauss-VB	0.94	0.83	0.95	0.72	0.86	1.0	0.67	0.876
	HMM-AR-VB	0.95	0.83	0.82	0.93	0.92	0.95	0.67	**0.889**
10	HDPHMM-Gauss-VB	0.94	0.95	0.95	0.86	0.94	0.90	0.67	0.915
	HDPHMM-Gauss-moVB	0.88	0.95	0.95	0.93	0.94	0.95	0.83	0.929
	HDPHMM-AR-VB	0.94	0.84	0.86	0.93	0.96	0.95	0.83	0.915
	HDPHMM-AR-moVB	1.0	0.95	0.95	0.93	0.98	1.0	1.0	**0.971**

robotic task. Therefore, we consider a Bayesian nonparametric verision of HMMs with a hierarchical prior which shares statistical strength across the training samples.

(2) Nonparametric HMMs Setting: We also train 4 kinds of classifiers base on non-parametric HMMs, independently for each anomaly class. We specify the truncation number of states as K=10 as explained in Sect. 4.6.1. Comparing to the parametric HMMs, the actual number of hidden states of each anomaly class is automatically learned from data in non-parametric fashion. The same procedure during testing as described in parametric HMMs. Benefits from the automatic state inference with HDPHMM, the auto-regressive correlation of the anomaly data, and the effective variational inference techniques. The summary of diagnoses results of non-parametric HMMs are presented in Table 5.1. Those numbers in blue are denoted the optimal diagnoses accuracy of specific anomaly type across all the methods, respectively.

∘ *HDPHMM-Gauss-VB:* A classifier based on HDPHMM with Gaussian observation was trained independently for each anomaly class. The standard VB inference algorithm [55] was used to learn the HDPHMM model. Similiar to the method proposed in [57], instead of the blocked Gibbs sampling algorithm, we learn the model by variational Bayesian inference.

∘ *HDPHMM-Gauss-moVB:* A classifier based on HDPHMM with Gaussian observation was trained independently for each anomaly class. A memoized online variational inference algorithm (moVB) [56] based on scalable adaptation of state complexity is used to learn this HDPHMM model.

∘ *HDPHMM-AR-VB:* A classifier based on HDPHMM with first-order autoregressive Gaussian observation was trained independently for each anomaly class. The standard VB inference algorithm [55] was used to learn the HDPHMM model.

∘ *HDPHMM-AR-moVB:* Finally, our results were evaluated on the HDPHMM with first-order autoregressive Gaussian observation for each anomaly class. A memoized online variational inference algorithm (moVB) [56] based on scalable adaptation of state complexity was used to learn this HDPHMM model.

Given that non-parametric sHDP-VAR-HMM learns the complexity of the model from the data, it produces more accurate models as is reflected by the higher diagnoses accuracies shown in Table 5.2. The learned number of states for the different anomaly types is shown in Table 5.3. Note that for equivalent parametric HMMs, a tedious model needs to be computed for each class to optimize the state cardinality between types. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. It is evident from the confusion matrix that the diagnoses outperforms other baseline methods.

5.5.3 Discussion

The multiclass classifier that is flagged when an anomaly is detected, was also tested through the sHDP-VAR-HMM on seven anomalies and six baseline methods. Our evaluations showed that we could not only detect anomalies reliably (overall accuracy

Table 5.2 Confusion matrix for diagnoses results on kitting experiment dataset with seven anomalies by HDPHMM-AR-moVB method

	TC	HC	HCO	NO	OS	WC	FP
TC	1.00	0.00	0.00	0.00	0.00	0.00	0.00
HC	0.00	1.00	0.00	0.00	0.00	0.00	0.00
HCO	0.00	0.00	0.95	0.00	0.05	0.00	0.00
NO	0.00	0.07	0.00	0.93	0.00	0.00	0.00
OS	0.00	0.00	0.00	0.00	1.00	0.00	0.00
WC	0.00	0.00	0.00	0.00	0.00	1.00	0.00
FP	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Table 5.3 The number of hidden states being active for different anomaly classes using HDPHMM-AR-moVB method. An active state is one in which at least one observation is assigned to this state

TC	HC	HCO	NO	OS	WC	FP
4	7	5	4	8	6	3

of 91.0%, as presented in Chap. 4) but also classify them precisely (overall accuracy of 97.1%).

With regards to anomaly diagnoses, anomalies usually occur from various sources such as sensing errors, control failures, or environmental changes. If a robot identifies the type, it may be beneficial to prevent or at least recover from the anomalous situation. In our proposed diagnoses method, we trained the sHDP-VAR-HMM model for each anomaly type separately. To address this, Natraj Raman had proposed a signal discriminative HDP-HMM for all classes albeit with an extra level that is class specific. Thus, an interesting future work direction consists in investigating a multilevel sHDP-VAR-HMM for all classes for multiclass classification and the extensions of the observation model by using higher order autoregressive Gaussian models.

Development of robot introspection is expected to have a direct impact on a large variety of practical applications, such as that can be used to estimate the log-likelihood of failure and prevent the failure during robot manipulation task. Also, the improvement of safety in human-robot collaborative environment by assessing the quality of learned internal model for each skill, which can speed up the anomaly recovery and/or repair process by providing the detailed skill identification and anomaly monitoring.

There are a number of limitations in our work. Currently we do not explicitly reduce the influence of outliers occasionally found in the derived log-likelihood values for specific hidden states. The outliers have a significant impact on the calculation of the threshold and our approach needs to address them specifically to avoid their impact. Additionally, we note the fact that our kitting experiment was not conducted under real factory conditions or in a real household daily task. Thus the verifiability of the work in real-world settings is unclear and further testing in real-factory conditions is necessary. The kitting experiment provides a proof-of-concept and the authors would like to extend their work to actual scenarios through corporate partners.

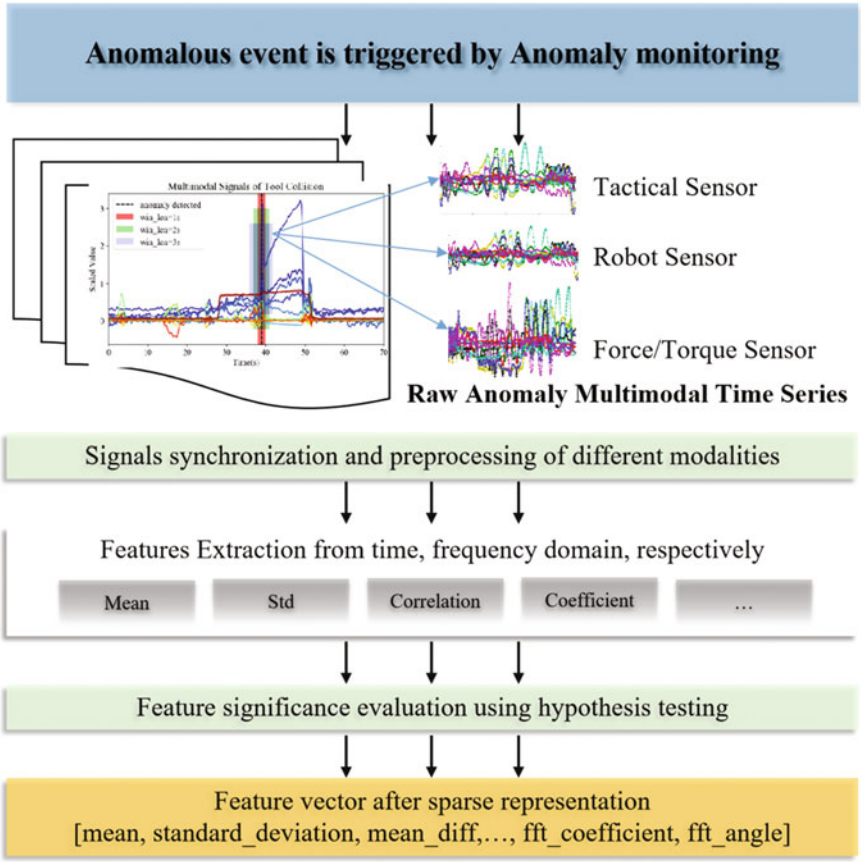

Fig. 5.7 The implementation procedure of sparse representation

5.6 Anomaly Classifier Based on Feature Extraction

The procedure of the sparse representation system for multimodal time-series is shown in Fig. 5.7.

5.6.1 Anomaly Sample Collection

- **Sensory Preprocessing**

The original multimodal sensory data includes a 6 DoF force and torque signals from F/T sensor, a 6 DoF Cartesian velocity signal from a manipulator's end-effector, 56 DoF tactical signals from a left and a right tactile sensor panels. To produce

more consistent data content, we do not directly concatenate individual data sources, instead temporal synchronization is done across modalities at $10HZ$. Also, different preprocessing techniques are done for specific modalities.

Wrench modality: Takes a force and torque time-series vector sequence and for each element represents the magnitude of each dimension $(f_x, f_y, f_z, t_x, t_y, t_z)$. Empirically, we wish that anomalies HC and TC can effectively flag external perturbations caused from different directions. We also consider the norm of force n_f and torque n_t respectively:

$$n_f = \sqrt{f_x^2 + f_y^2 + f_z^2}, \quad n_t = \sqrt{t_x^2 + t_y^2 + t_z^2} \tag{5.3}$$

Velocity modality: We measure the linear (l_x, l_y, l_z) and angular (a_x, a_y, a_z) Cartesian velocity (the endpoint state of a Baxter right hand), which are reported with respect to the base frame of the robot. As with the wrench source, we also consider the norm of the linear velocity n_l and angular velocity n_a respectively.

$$n_l = \sqrt{l_x^2 + l_y^2 + l_z^2}, \quad n_a = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{5.4}$$

Taxel modality: Due to the high dimensionality of tactical sensor, we do not process all the original signals as the model input. After empirical testing, the standard deviation of each tactile sensors s_l, s_r were selected as the preferred features and defined as:

$$s_l = \sqrt{\frac{1}{28}\sum_{i=1}^{28}(l_i - \mu_l)}, \quad s_r = \sqrt{\frac{1}{28}\sum_{i=1}^{28}(l_i - \mu_r)} \tag{5.5}$$

where the $\mu_l = \frac{1}{28}\sum_{i=1}^{28} l_i$ and $\mu_r = \frac{1}{28}\sum_{i=1}^{28} r_i$ is the mean of each tactical panel, respectively.

The above three preprocessing operations capture unexpected changes in the original signals since the signals are slightly different from the robot variable progress in normal executions. We then concatenate all the features to represent robot executions both in nominal and anomalous cases. Note that raw concatenated features can easily result in high False Positive Rates (FPR) during execution due to significant task execution variability across the same task in our experiments. For instance, the F/T signals vary different across differing objects of different weights; or similarly, human collisions which occur from different directions and magnitudes.

A standardization method is used to scale original signals ξ_o by its mean and standard deviation according to:

$$\xi(*) = \frac{\xi_o(*) - mean(\xi_o(*))}{std(\xi_o(*))}, \quad * \in \{f_x, f_y, ...s_l, s_r\}. \tag{5.6}$$

Finally, our eighteen dimensional multimodal signal(wrench, velocity, and tactical modalities) is represented as:

$$
\begin{aligned}
y_t = [&\xi(f_x), \xi(f_y), \xi(f_z), \xi(t_x), \xi(t_y), \xi(t_z),\\
&\xi(n_f), \xi(n_t), \xi(l_x), \xi(l_y), \xi(l_z), \xi(a_x),\\
&\xi(a_y), \xi(a_z), \xi(n_l), \xi(n_a), \xi(s_l), \xi(s_r)].
\end{aligned} \tag{5.7}
$$

5.6.2 Anomaly Features Extraction

In order to keep temporal consistency, anomaly signals are considered for a given *window_size* which is fixed around events flagged as anomalous. For instance, if a tool collision is detected as shown in Fig. 5.8, then *window_size* can evaluate how the diagnoses reactivity performs in our system. Generally, the *window_size* will equal a power of two (the a preferred size when including the Fast Fourier Transformation (FFT) between the time and frequency domain).

As described above, the sparse representation is applied for each extracted sample window. According to our previous work on anomaly diagnoses [20], the features are extracted in both the time domain and frequency domain. For anomaly diagnoses, we empirically consider the independent features and the corrective features along a specific modality signal as in Eq. 5.7. Here, the original multimodal signals with 18 DoFs and 12 feature types are considered in both the time and frequency domains, where the final feature vector is of length 558.

A: Independent features

Here, we calculate the features along a specific modality signal $\xi\{*\} = (x_1, x_2, ..., x_n)$ with n data points

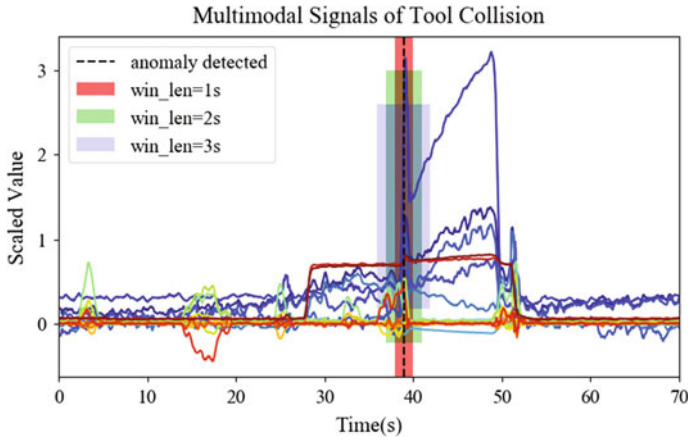

Fig. 5.8 Our robot introspection system for extracting the anomaly data when anomaly detected. For instance, the data of tool collision is represented with a given $window\_size = \pm 2$ s in red background

1. mean

$$\mu = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) \tag{5.8}$$

2. standard_deviation

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{5.9}$$

3. mean_diff: calculate the mean over the differences between subsequent values

$$\mu_{diff} = \frac{1}{n}\sum_{i=1}^{n-1}(x_{i+1} - x_i) \tag{5.10}$$

4. mean_abs_diff: calculate the mean over the absolute differences between subsequent values

$$\mu_{abs\_diff} = \frac{1}{n}\sum_{i=1}^{n-1}|x_{i+1} - x_i| \tag{5.11}$$

5. abs_energy: calculate the absolute energy, that is the sum over the squared values

$$e_{abs} = \sum_{i=1}^{n} x_i^2 \tag{5.12}$$

B: Correlative features

1. autocorrelation: calculate the autocorrelation of the specified lags $k \in \{1, 2, 3, 4\}$ given the mean μ and variance σ^2, respectively.

$$\mathscr{R}_k = \frac{1}{(n-k)\sigma^2}\sum_{t=1}^{n-k}(x_t - \mu)(x_{t+k} - \mu) \tag{5.13}$$

2. mean_autocorrelation: calculate the mean of the autocorrelation which taken over all possible lags $l \in \{1, ..., n\}$

$$\mu(\mathscr{R}) = \frac{1}{n-1}\sum_{l=1}^{n}\mathscr{R}_l \tag{5.14}$$

3. std_autocorrelation: calculate the standard deviation of the autocorrelation which taken over all possible lags $l \in \{1, ..., n\}$

$$\sigma(\mathscr{R}) = \sqrt{\frac{1}{n-1}\sum_{l=1}^{n}(\mathscr{R}_l - \mu(\mathscr{R}))^2} \tag{5.15}$$

4. ar_coefficient: get the first 5 coefficients by fitting the unconditional maximum likelihood of an autoregressive model $\mathscr{AR}(p)$ with order p. In case of our implementation, the order is a fixed $p = 10$. The $\mathscr{AR}(5)$ model is defined as

$$x_t = \varphi_0 + \sum_{i=1}^{p=10} \varphi_i x_{t-i} + \varepsilon_t \tag{5.16}$$

where ε_t is drawn from a Gaussian white noise with a mean of zero and unit variance.

5. partial_autocorrelation: calculate the value of partial autocorrelation function of given lag $k \in \{1, 2, 3, 4, 5\}$, denoted $\alpha(k)$, is the autocorrelation between x_t and x_{t+k} with the linear dependence of x_t on x_{t+1} through x_{t+k-1} removed. As such, the function is defined as

$$\begin{aligned} Cov(x_t, x_{t-k}) &= Cov(x_t, x_{t-k}|x_{t-1}, .., x_{t-k+1}) \\ var(x_t) &= Var(x_t|x_{t-1}, ..., x_{t-k+1}) \\ var(x_{t-k}) &= Var(x_{t-k}|x_{t-1}, .., x_{t-k+1}) \\ \alpha(k) &= \frac{Cov(x_t, x_{t-k})}{\sqrt{var(x_t)var(x_{t-k})}} \end{aligned} \tag{5.17}$$

C: Spectrum-based features

1. fft_coefficient: Get the first 5 Fourier coefficients of real part by FFT algorithm, respectively. In this implementation, we define the discrete Fourier transform function as

$$\begin{aligned} \mathscr{F} &= \sum_{m=0}^{n-1} a_m \exp\{-2\pi \frac{mk}{n}\}, \quad k = 0, ..., n-1 \\ a_m &= \exp\{2\pi i f m \Delta t\} \end{aligned} \tag{5.18}$$

where, Δt is the sampling interval.

2. fft_angle: Calculate the angle of obtained complex value $\mathscr{F}$ of the first 5 Fourier coefficients, respectively.

$$\theta = \tan^{-1} \frac{\mathscr{F}_{imag}}{\mathscr{F}_{real}} \tag{5.19}$$

5.6.3 Experiments and Results

- **Experimental Setup**

A kitting experiment consists of 5 basic nodes: Home, Pre-pick, Pick, Pre-place, and Place. The experiment is implemented in the following order with those nodes by 6 skills with the ROS-SMACH,[2] (Skill 1): Home → Pre-pick; (Skill 2): Pre-pick → Pick; (Skill 3): Pick → Pre-pick; (Skill 4): Pre-pick → Pre-place; (Skill 5): Pre-place → Place; (Skill 6): Place → Pre-place, as shown in Fig. 4.11.

The primary goal of the kitting task is designed to transport 6 different objects to a fixed container. The right arm of Baxter humanoid robot is used to pick objects and equipped with a 6 DoF Robotiq FT sensor, 2 Baxter-standard electric pinching fingers. Each finger is further equipped with a multimodal tactile sensor that a four by seven taxel matrix that yield absolute pressure values. The left hand camera is placed flexibly in a region that can capture objects with a resolution of 1280×800 at 1 fps (we optimize pose accuracy and lower computational complexity in the system). The use of the left hand camera facilitated calibration and object tracking accuracy. All code was run in ROS Indigo and Linux Ubuntu 14.04 on a mobile workstation with an Intel Xeon processor, 16GB RAM, and 8 cores.

When robot collaboratively works with human in a shared workspace, so many external disturbances are likely to occur. Those anomalies in Fig. 4.12 are considered in the Kitting experiment, which including the following 7 types: Tool Collision (TC) that may be derived from the visual error or the user accidentally collide with the object during robot moving to grasp it (see Fig. 4.12b); Human Collision (HC) is usually happened by a user to unintentionally collide with the robot arm in the human-robot collaboration environment(see Fig. 4.12a). We treat the human collision differently from whether the robot carrying object or not. Thus, Human Collision with Object (HCO) is assumed that the human collision while robot carrying object from the node $Pre-pick$ to $Pre-place$ (see Fig. 4.12d). The object have been knock down by the robot during grasping may induce the No object (NO) or missed-grasps(see Fig. 4.12f). Another common anomaly is Object Slip (OS) that the picked object may slip from the robot's gripper if the grasping pose is not optimal or the robot moves at high speed. Finally, the False Positive (FP) is labeled when some unexpected disturbances may be detected by the anomaly detector for a variety of reasons, for instance, the system error, the object is placed at unreachable zone, without feasible inverse kinematic solution, and so on. In the rest of this paper, we intentionally achieve the spare representation of the seven types of anomalies while preserved sufficient diagnoses accuracy, respectively.

- **Results and Analysis**

The dataset contains a total of 108 samples from 137 experimental recordings of kitting task and the proportion for each anomaly are TC 15.7%, HC 16.7%, HCO 16.7%, NO 13.0%, OS 16.7%, WC 15.7%, FP 5.6%, respectively. For evaluating the

[2] http://www.ros.org/SMACH.

Table 5.4 Multiclass classifiers

Index	Classifier	Comment
1	BernoulliNB	Binarizing the dataset with the boolean threshold μ of each madality axis, which indicates that feature values below or equal to the threshold are replaced by 0, otherwise, by 1. The smoothing parameter $\alpha = 1.0$ and with a uniform prior on the class distribution
2	GaussianNB	Default settings
3	DecisionTree	A non-parametric supervised learning method used for diagnoses. A major advantage is that it does not require huge data preparation
4	RandomForest	Default settings
5	LinearSVC	Default settings
6	LogisticRegression	Default settings
7	SGDClassifier	Preprocessing procedure is added to convert the dataset with zero mean and unit variance
8	RidgeClassifier	Default setting
9	KnnDtw	n_neighbors=2 and max_warping_window=10

performance of the proposed sparse representation of multivariate time-series, we took the following 9 representative classifiers[3] into consideration and the parameter settings are described respectively in Table 5.4.

As described above, the feature selection is a process where the most significant features in predicting the outcome are selected automatically. However, irrelevant features decrease the model's accuracy and increase computational cost. We therefore calculate the p_value of each extracted feature by using the hypothesis testing method in spscitefresh2016. That is, we preform a singular statistical test checking the hypotheses for each extracted feature $f_1, f_2, ..., f_n$,

$$\begin{aligned} H_0^i &= \{x_i \text{ is irrelevant for predicting class } y\}; \\ H_1^i &= \{x_i \text{ is relevant for predicting class } y\}; \end{aligned} \tag{5.20}$$

The result of hypothesis test in Eq. 5.20 is a p_value, which assess the probability that feature x_i is not relevant for predicting class y. As such, we define the score of feature by calculating the negative logarithmic value on the p_value. Large scores $-\log(p\_values)$ indicate features, which are relevant for predicting the target.

The performance of different classifiers is shown in Fig. 5.9, as you can see (reading left to right on the graph), the accuracy indicates to increase as the number of features are added, until a point beyond which there seems to be too few features for the classifier to make any reliable conclusions. Specifically, those features are extracted from a sorted feature vector in descending order by score values.

[3] http://scikit-learn.org/.

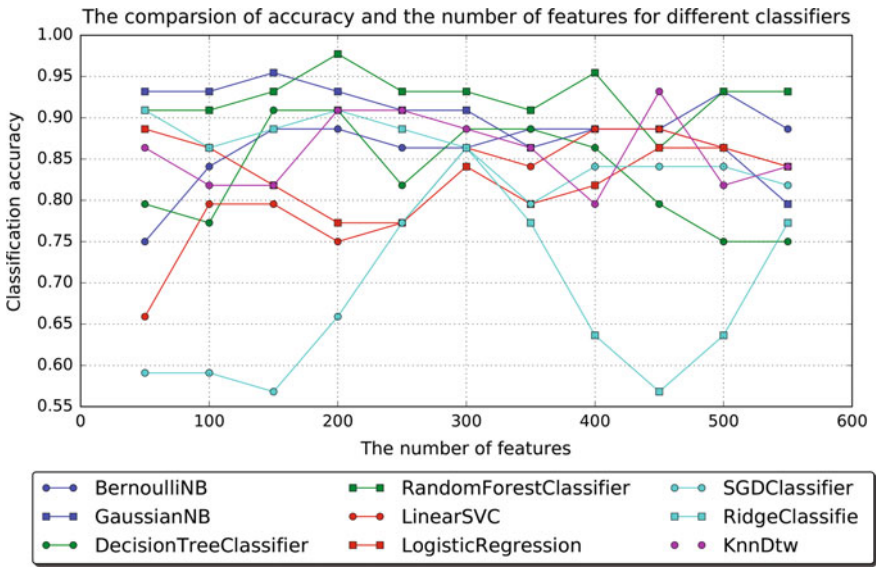

Fig. 5.9 The comparison between diagnoses accuracy and the number of feature on different classifiers

5.6.4 Discussion

This work implements sparsely represent the recorded multimodal time-series with relevant features as small as possible while preserving diagnoses accuracy. We propose the multivariate features are extracted in both time domain and frequency domain and not only consider the static statistical characteristics, but also including the correlation and interaction of each dimensional sensory signal. Results indicate that the data set can be significantly reduced up to 72.2% $\sim$ 86.1% (the number of features is 100 and 200, respectively) of the raw data while keep the average diagnoses accuracy at about 85% with small data preparation. Future work should therefore include analyzing the trade-off between the value $window\_size$ and the diagnoses accuracy. So as to represent the recorded multimodal time-series with relevant features as small as possible while preserved diagnoses accuracy.

5.7 Summary

In this chapter, anomaly diagnose methods using the Bayesian nonparametric hidden Markov models and sparse representation by statistical feature extraction when anomaly triggered are introduced. Specifically, the detail procedure for anomaly sample definition, the supervised learning dataset collection as well as the data augmen-

tation of insufficient samples are also presented. The proposed methods is verified with a multi-step human-robot collaboration objects kitting task on Baxter robot, the performance and results are presented of each method respectively. That is, a multi-target classifier based on the nonparametric Bayesian model is proposed, Which using the sHDP-VAR-HMM model to model the anomaly sample data of various anomaly types. For the task of Baxter robot performing kitting experiment, resulting in the diagnoses accuracy of a total of 7 types of anomaly events in is 97.1%. Additionally, for the sparse representation for anomaly diagnoses, we extract a total of 31 features of time and frequency domains of the anomaly sample. Then, the extracted features importance analysis using the hypothesis testing method, after that the filtered features are verified on 9 common out-of-the-box supervised learning methods for multi-class diagnoses in *sklearn*.

References

1. Krogh A, Larsson BÈ, Von Heijne G, et al. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes[J]. J Mol Biol. 2001;305(3):567–80.
2. Och FJ, Ney H. The alignment template approach to statistical machine translation[J]. Comput Linguist. 2004;30(4):417–49.
3. Och FJ, Ney H. A comparison of alignment models for statistical machine translation[C]. In: Proceedings of the 18th conference on computational linguistics-volume 2. Association for Computational Linguistics;2000. p. 1086–90.
4. Karlof C, Wagner D. Hidden Markov model cryptanalysis[C]. In: International workshop on cryptographic hardware and embedded systems. Springer;2003. p. 17–34.
5. Green PJ, Noad R, Smart NP. Further hidden Markov model cryptanalysis[C]. In: International workshop on cryptographic hardware and embedded systems. Springer;2005. p. 61–74.
6. Bhar R, Hamori S. Hidden Markov models: applications to financial economics[M]. Springer Science Business Media;2004.
7. Rogemar S Mamon, Robert J Elliott. Hidden markov models in finance[M]. Springer;2007.
8. Xu Z, Li S, Zhou X, Wu Y, Cheng T, Huang D. Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties[J]. Neurocomputing. 2019;329(1):255–66.
9. Veenendaal A, Daly E, Jones E, et al. Sensor tracked points and HMM based classifier for human action recognition[J]. Comput Sci Emerg Res J. 2016;5:4–8.
10. Ren L, Patrick A, Efros AA et al. A data-driven approach to quantifying natural human motion[C]. ACM Trans Graph (TOG). ACM;2005;24:1090–97.
11. Hovland GE, McCarragher BJ. Hidden Markov models as a process monitor in robotic assembly[J]. Int J Robot Res 1998;17(2):153–68.
12. Alshraideh H, Runger G. Process monitoring using hidden Markov models[J]. Quality Reliab Eng Int. 2014;30(8):1379–87.
13. Chuk TY, Chan AB, Shimojo S et al. Mind reading: discovering individual preferences from eye movements using switching hidden Markov models[C]. In: Proceedings of the 38th annual conference of the cognitive science society, CogSci 2016.
14. Manogaran G, Vijayakumar V, Varatharajan R et al. Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering[J]. Wirel Person Commun. 2018;102(3):2099-16.
15. Geurts P. Pattern extraction for time series classification[C]. In: European conference on principles of data mining and knowledge discovery. Springer;2001. p. 115–27.

16. Hüsken M, Stagge P. Recurrent neural networks for time series classification[J]. Neurocomputing. 2003;50:223–35.
17. Graves A. Supervised sequence labelling[M]. Supervised sequence labelling with recurrent neural networks. Springer;2012. p. 5-13.
18. Orsenigo C, Vercellis C. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification[J]. Pattern Recognit. 2010;43(11):3787–94.
19. Lee Y-H, Wei C-P, Cheng T-H, et al. Nearest-neighbor-based approach to time-series classification[J]. Decis Support Syst. 2012;53(1):207–17.
20. M-L Zhang, Z-H Zhou. A k-nearest neighbor based algorithm for multi-label classification[C]. In: 2005 IEEE international conference on granular computing. IEEE;2005. vol. 2, p. 718–21.
21. Seto S, Zhang W, Zhou Y. Multivariate time series classification using dynamic time warping template selection for human activity recognition[J]. 2015. arXiv:151206747.
22. Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series[C]. In: KDD workshop. Seattle, WA, 1994, vol. 10, p. 359–70.
23. Baydogan MG, Runger G, Tuv E. A bag-of-features framework to classify time series[J]. IEEE Trans Pattern Anal Mach Intell. 2013;35(11):2796–802.
24. Han M, Liu X. Feature selection techniques with class separability for multivariate time series[J]. Neurocomputing. 2013;110:29–34.
25. Fulcher BD, Jones NS. Highly comparative feature-based time-series classification[J]. IEEE Trans Knowl Data Eng. 2014;26(12):3026–37.
26. Nanopoulos A, Alcock R, Manolopoulos Y. Feature-based classification of time-series data[J]. Int J Comput Res. 2001;10(3):49–61.
27. Utomo C, Li X, Wang S. Classification based on compressive multivariate time series[C]. In: Australasian Database Conference. Springer;2016. p. 204–14.
28. Jaakkola T, Diekhans M, Haussler D. A discriminative framework for detecting remote protein homologies[J]. J Comput Biol. 2000;7(1–2):95–114.
29. Raman N, Maybank SJ. Action classification using a discriminative multilevel HDP-HMM[J]. Neurocomputing. 2015;154:149–61.
30. Di Lello E. Bayesian Time-Series Models: Expert Knowledge-Driven Inference and Learning for Engineering Applications[J]. 2015.
31. Karim F, Majumdar S, Darabi H, et al. LSTM fully convolutional networks for time series classification[J]. IEEE Access. 2018;6:1662–9.
32. Karim F, Majumdar S, Darabi H, et al. Multivariate LSTM-FCNs for Time Series Classification[J] 2018. arXiv:180104503.
33. Che Z, Purushotham S, Cho K, et al. Recurrent neural networks for multivariate time series with missing values[J]. Sci Rep. 2018;8(1):6085.
34. Yi Zheng, Qi Liu, Enhong Chen, et al. Time series classification using multi-channels deep convolutional neural networks[C]//International Conference on Web-Age Information Management. Springer, 2014: 298-310.
35. Fu Y, editor. Human activity recognition and prediction. Switzerland: Springer; 2016.
36. Geurts P. Pattern extraction for time series classification. In: European conference on principles of data mining and knowledge discovery, 3. Berlin, Heidelberg: Springer; 2001. p. 115–27.
37. Pavlovic V, Frey BJ, Huang TS. Time-series classification using mixed-state dynamic Bayesian networks. In: Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149). IEEE;1999 Jun 23. vol. 2, p. 609–15.
38. Xu Z, Li S, Zhou X, Cheng T. Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties[J]. Neurocomputing. 2019;357(1):271–81.
39. Park D, Kim H, Hoshi Y, Erickson Z, Kapusta A, Kemp CC. A multimodal execution monitor with anomaly classification for robot-assisted feeding. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2017 Sep 24. p. 5406–13.
40. Pettersson O. Execution monitoring in robotics: a survey. Robot Auton Syst. 2005;53(2):73–88.
41. Bjäreland M. Model-based execution monitoring. InLinköping Studies in Science and Technology, Dissertation No 688. 2001. http://www.ida.liu.se/labs/kplab/people/marbj.

42. Muradore R, Fiorini P. A PLS-based statistical approach for fault detection and isolation of robotic manipulators. IEEE Trans Ind Electron. 2011;59(8):3167–75.
43. Orsenigo C, Vercellis C. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. Pattern Recognit. 2010;43(11):3787–94.
44. Seto S, Zhang W, Zhou Y. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In: 2015 IEEE symposium series on computational intelligence. IEEE;2015 Dec 7. p. 1399–06.
45. Karim F, Majumdar S, Darabi H, Harford S. Multivariate lstm-fcns for time series classification. Neural Netw. 2019;1(116):237–45.
46. Pei W, Dibeklioğlu H, Tax DM, van der Maaten L. Multivariate time-series classification using the hidden-unit logistic model. IEEE Trans Neural Netw Learn Syst. 2017;29(4):920–31.
47. Baydogan MG, Runger G. Learning a symbolic representation for multivariate time series classification. Data Min Knowl Discov. 2015;29(2):400–22.
48. Park D, Erickson Z, Bhattacharjee T et al. Multimodal execution monitoring for anomaly detection during robot manipulation[C]. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE;2016. p. 407–14.
49. Xu Z, Li S, Zhou X, Yan W, Cheng T, Dan H. Dynamic neural networks for motion-force control of redundant manipulators: an optimization perspective [J]. IEEE Trans Ind Electro, Early access. 2020. https://doi.org/10.1109/TIE.2020.2970635.
50. Park D, Kim H, Hoshi Y et al. A multimodal execution monitor with anomaly classification for robot-assisted feeding[C]. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE;2017. p. 5406–13.
51. Bunte K, Haase S, Biehl M, Villmann T. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. Neurocomputing. 2012;1(90):23–45.
52. Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. Stanford; 2006 Jun 7.
53. Welch LR. Hidden Markov models and the Baum-Welch algorithm. IEEE Inf Theory Soc Newslett. 2003;53(4):10–3.
54. McGrory CA, Titterington DM. Variational Bayesian analysis for hidden Markov models. Aust N Z J Stat. 2009;51(2):227–44.
55. Hughes MC, Sudderth EB. Bnpy: Reliable and scalable variational inference for bayesian nonparametric models. In: Proceedings of the NIPS probabilistic programming workshop, Montreal, QC, Canada. 2014. p. 8–13.
56. Hughes MC, Sudderth E. Memoized online variational inference for Dirichlet process mixture models. In: Advances in neural information processing systems. 2013. p. 1133–41.
57. Di Lello E, Klotzbücher M, De Laet T, Bruyninckx H. Bayesian time-series models for continuous fault detection and recognition in industrial robotic tasks. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2013 Nov 3. p. 5827–33.