Robust Predictive Maintenance for Robotics via Unsupervised Transfer Learning

Arash Mahyari[1], Thomas Locher[2]
[1]amahyari@ihmc.org, [2]thomas.locher@dfinity.org
[1]Florida Institute for Human and Machine Cognition (IHMC), Ocala, Florida, USA
[2]DFINITY, Switzerland

Abstract

Industrial robots play an increasingly important role in a growing number of fields. Since the breakdown of a single robot may have a negative impact on the entire process, predictive maintenance systems have gained importance as an essential component of robotics service offerings. The main shortcoming of such systems is that features extracted from a task typically differ significantly from the learnt model of a different task, incurring false alarms. In this paper, we propose a novel solution based on transfer learning which addresses a well-known challenge in predictive maintenance algorithms by passing the knowledge of the trained model from one task to another in order to prevent the need for retraining and to eliminate such false alarms. The deployment of the proposed algorithm on real-world datasets demonstrates that the algorithm can not only distinguish between tasks and mechanical condition change, it further yields a sharper deviation from the trained model in case of a mechanical condition change and thus detects mechanical issues with higher confidence.

Introduction

Robots have revolutionized the manufacturing process by performing tasks more efficiently and accurately at a lower operational cost. However, sudden breakdowns or malfunctions of robots can result in a steep decrease in production quality and quantity, which often entails substantial financial losses. Robot failures and malfunctions can have various causes, e.g., moving heavy objects continuously leads to the deterioration of robots gears over time. As a result of this deterioration, gaps can appear between the gear teeth, which results in so-called *backlash*. Backlash is a clearance or lost motion in a mechanism.

In order to prevent unscheduled maintenance due to sudden disruptions of normal operation, *predictive maintenance* has gained a lot of attention recently (Hornung et al. 2014; Mobley 2002). The goal of predictive maintenance is to prevent unexpected equipment failure by constantly monitoring the performance and condition of equipment in oper-

ation and to extrapolate from the current condition when corrective maintenance will be required (Grall et al. 2002; Hashemian 2010). The focus of this paper is to identify hardware issues through analyzing the internal signals of the robots using machine learning algorithms.

There have been several research activities recently in this domain (Hornung et al. 2014; Park et al. 2016; Sjöstrand, Blanc, and Lindin 2012; Pinto and Cerquitelli 2019; Moisescu et al. 2018; Sathish et al. 2019). The problem with existing predictive maintenance algorithms is that they raise false alarms when the task for which data is collected differs from the task used during the training phase. When the task changes, features such as frequency components extracted from data change as well. This change in the features causes predictive-maintenance systems to erroneously raise an alarm, although the robot is healthy. To avoid false alarms due to an operational change, current systems require the model to be retrained when altering the task of the robot.

In this paper, we propose to use *transfer learning*, also known as knowledge transfer and domain adaptation, as an intermediate step in order to avoid retraining the model every time the task of the robot changes. In the proposed method, the training data is collected while the robot is healthy and the extracted features of the healthy robot are in a subspace called *source domain*, $\mathcal{D}_s$. The test data is collected while the robot is performing another task, and its features are in the *target domain*, $\mathcal{D}_t$. The goal of the transfer learning algorithm is to transfer the model learned in the source domain to the target domain (Pan and Yang 2010). We assume that labeled data of healthy and faulty robots are not available in the target domain. Therefore, unsupervised methods must be used to transfer the knowledge between the domains. In this paper, we propose to use *manifold alignment* (Wang and Mahadevan 2009; Boucher et al. 2015), which is a local-preserving algorithm that finds a common subspace of the source and the target domains. Figure 1 shows the flowchart of the proposed system. Furthermore, because it is not convenient to ask the users of the robot to run the transfer learning algorithm each time the task of the robot changes, we assume that the test data is from the target domain. Thus, the manifold alignment algorithm is continuously applied to the features extracted from the test data to make sure that the comparison between healthy data and the test data is performed in the same domain.

[†]This work was done while A. Mahyari worked at the ABB Robotics R&D Center, 3055 Orchard Dr., San Jose, CA 95134, and T. Locher worked at ABB Research, Baden-Dättwil, Switzerland.
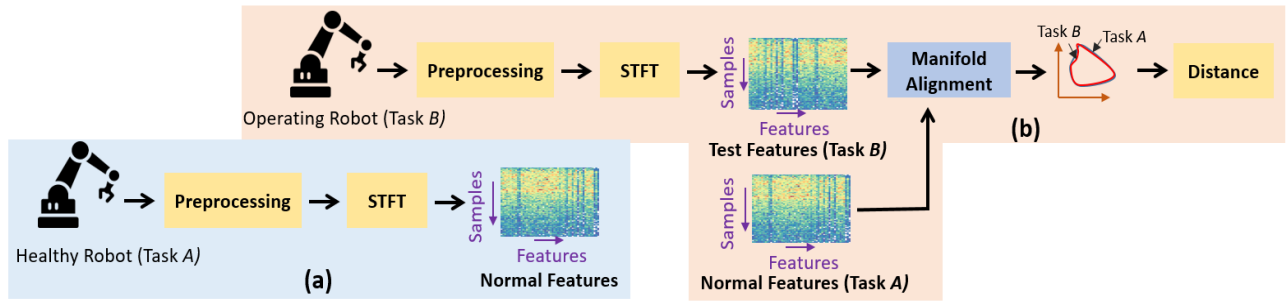[*] Parts of this paper are patented.

Figure 1: Flowchart of the proposed work: (a) Training from the healthy robot with task A; (b) Evaluating (test) the operating robot in the customer facility with unknown task B using domain adaptation.

Scenario

Dataset

The dataset is collected from two robot types. Robot $\mathcal{A}$ is from a series of single arm robots with 6 axis. Their reach is up to 3m (10ft) and their payload is around 100kg (220lb). Robot $\mathcal{B}$ is also from a series of 6-axis single arm robots and reach of up to 4m (13ft) and a maximum payload of 600kg (1320lb). Three signals "position, speed, and torque" are recorded from the controller of these robots for each axis separately (three samples per time from each axis). The sampling frequency of speed and torque signals are $2kHz$ while the position signal is sampled at a frequency of $250Hz$. Since the dataset comprises real-world data from robots in operation, we cannot provide more details about the robots, nor their specific tasks due to confidentiality obligations. However, we believe that the generic nature of our proposed predictive maintenance method makes it possible to apply it successfully for a broad range of robot types and tasks.

Pre-processing

Since the sampling frequency of the position signal is lower than the other two signals, as mentioned above, the position signal is upsampled to get the same number of data points for each signal. The position signal is upsampled by first inserting 7 data points with value zero between pairs of actual signals, which artificially increases the data rate to $2kHz$. Next, the zero values are overwritten using the cubic interpolation method, which fits a third degree polynomial function to the samples of the position signal (Fritsch and Carlson 1980). The fitted polynomial function is used to approximate the values of position signals at the newly inserted sample points. After pre-processing, each signal $x(t) = [speed(t), position(t), torque(t)]^T$ provides one sample at a time t, where $speed(t)$, $position(t)$, and $torque(t)$ are scalar values.

Feature Extraction

After the data is pre-processed, a certain set of features is required to be extracted from the given three signals. The features should not only accurately represent normal data but also discriminate normal from anomalous data well. In this paper, the short-time Fourier transform (STFT) of the

three signals in $x(t)$ and their combinations is computed as the set of features. The reason for choosing STFT is that it adequately represents the time-frequency distribution of the signals, suitable to identify various anomalies. For example, gearbox malfunctioning and backlash appears as vibration during robot operation. STFT captures these vibrations as different frequency components at different time steps.

Detection

The absolute values of STFT magnitude are used to build the subspace representing the healthy robot. Naturally, the performance of the anomaly detection mechanism critically hinges upon the accuracy of this subspace. Before delving into our proposed method to derive such a subspace in the subsequent section, we briefly summarize a common and straightforward approach: principal component analysis (PCA) can be applied to form the subspace, i.e., the principal components of the absolute STFT values constitute the subspace associated with a healthy robot. Given this subspace, training and test datasets are projected to this subspace using the principal components. The ℓ_2-norm distance between the training dataset (of a healthy robot) and the test dataset in this subspace is used as a criterion to identify whether the test data represents a healthy or faulty robot. If the test dataset comes from a healthy robot, the subspace should be able to represent the features of the test data well and the distance to the training data in this subspace should be small. On the other hand, a large distance is an indication of a faulty robot.

Formally, let $\mathbf{X}_s$ and $\mathbf{X}_t$ represent absolute STFT values of training (healthy) and test data, respectively, where rows are samples and columns are the flattened absolute STFT values of a particular sample. Our distance criterion is defined as $d := \|\mathbf{X}_s\mathbf{P} - \mathbf{X}_t\mathbf{P}\|_2^2$, where $\mathbf{P}$ is the matrix whose columns are the principal components. We formulate the detection problem as: $H_0 : d \geq \varepsilon$ vs. $H_1 : d < \varepsilon$, where ε is the threshold determining whether the test data represents a healthy or faulty condition. The parameter ε is determined by the significance level, α, or the percentile of the distribution.

Proposed Method

Let $\mathcal{X}_s$ represent the feature space in the source domain, and $\mathbf{X}_s \in R^{N \times K}$ are K features (absolute STFT values) of N training samples drawn from this space. Let

---

The actual robot names cannot be provided for confidentiality reasons as the robots are used in production.

$\mathbf{X}_t \in R^{N \times K}$ be K features of N test samples (absolute STFT values) drawn from the target feature space $\mathcal{X}_t$ and collected while the robot is in operation. Since the application of the robot during training and test are different, their corresponding feature spaces are different, i.e., $\mathcal{X}_s \neq \mathcal{X}_t$. Therefore, there is a need for a transfer learning algorithm to reduce the difference between these two spaces while preserving the geometric properties (Pan et al. 2011). This reduction can be achieved by finding a common subspace between source and target spaces through minimizing a certain cost function. Manifold alignment is an unsupervised tranfer learning (or domain adaptation) algorithm that provides a closed-form solution (Boucher et al. 2015; Wang and Mahadevan 2009). Having a closed-form solution makes it possible to implement it in a computationally efficient manner, which can be a requirement for applications with real-time constraints. For this reason, we chose manifold alignment to perform domain adaptation. The manifold alignment algorithms replaces the application of PCA as discussed in the previous section. Hence, the input to the manifold alignment algorithm is the STFT values, $\mathbf{X}_s$ and $\mathbf{X}_t$, and the output is the computed distance d. More precisely, the manifold alignment algorithm computes low-rank embeddings (LREs) of $\mathbf{X}_s$ and $\mathbf{X}_t$ in a joint subspace. The LREs are then used to calculate the distance.

We will now discuss how the LREs of the source and target features, $\mathbf{X}_s$ and $\mathbf{X}_t$, are calculated. The objective is to minimize the following loss function:

$$\min_{R_s} \tfrac{1}{2}\|\mathbf{X}_s - \mathbf{X}_s\mathbf{R}_s\|_F^2 + \lambda\|\mathbf{R}_s\|_* \\ \min_{R_t} \tfrac{1}{2}\|\mathbf{X}_t - \mathbf{X}_t\mathbf{R}_t\|_F^2 + \lambda\|\mathbf{R}_t\|_*, \quad (1)$$

where $\lambda > 0$, $\|.\|_F$ and $\|.\|_*$ are Frobenius and spectral norms, respectively. In this equation, $\mathbf{X}_s\mathbf{R}_s$ and $\mathbf{X}_t\mathbf{R}_t$ are the low rank embeddings of $\mathbf{X}_s$ and $\mathbf{X}_t$, respectively, and $\mathbf{R}_s$ and $\mathbf{R}_t$ are their reconstruction coefficient matrices. $\mathbf{R}_s := \hat{\mathbf{V}}_s(\mathbf{I} - \hat{\mathbf{S}}_s^{-2})\hat{\mathbf{V}}_s^T$ and $\mathbf{R}_t := \hat{\mathbf{V}}_t(\mathbf{I} - \hat{\mathbf{S}}_t^{-2})\hat{\mathbf{V}}_t^T$, where $\hat{\mathbf{S}}_s$ is the diagonal matrix of all singular values greater than one and matrix $\hat{\mathbf{V}}_s$ is comprised of the corresponding right-singular vectors. The block reconstruction coefficient matrix is: $\mathbf{R} = \left(\begin{smallmatrix} \mathbf{R}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_t \end{smallmatrix}\right)$.

The inter-set correspondence between the samples of the training and test datasets is represented by $\mathbf{C} = \left(\begin{smallmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{smallmatrix}\right)$, where $\mathbf{I}$ is the identity matrix.

After finding the LRE of source and target samples, the projection matrices from the source and the target space into the common subspace and the embedding of the source and target samples are calculated by minimizing

$$(1-\mu)\|\mathbf{F} - \mathbf{R}\mathbf{F}\|_F^2 + \mu \sum_{i,j=1}^{N} \|\mathbf{F}_i - \mathbf{F}_j\|^2 C(i,j), \quad (2)$$

where $\mu \in [0,1]$ determines the importance of the local geometry (first term) vs. the inter-set correspondence (second term). The cost function can be simplified to the following expression (Boucher et al. 2015):

$$(1-\mu)(\mathbf{I} - \mathbf{R})^T(\mathbf{I} - \mathbf{R}) + 2\mu\mathbf{L}, \quad (3)$$

where $\mathbf{L}$ is the Laplacian matrix of $\mathbf{C}$. This cost function is minimized by replacing $\mathbf{F} = [\mathbf{F}_s\mathbf{F}_t]^T$ with the d smallest eigenvectors of Equation 3. $\mathbf{F}$ is the d-dimensional embedding of N training and N test features in the common subspace.

Since $\mathbf{X}_s$ is assumed to be the training features collected when the robot is healthy, the d dimensional embedding of the test dataset is compared to the embedding of the test dataset using the Euclidean distance, i.e., $\delta(t) = \sqrt{\|\mathbf{F}_s - \mathbf{F}_t\|_2^2}$. If $\mathbf{X}_t$ is anomalous, its distance is larger than for $\mathbf{X}_s$. In order to perform hypothesis testing on the test data, the metric δ is calculated for several normal datasets to build a probability distribution function. The empirical distribution of the metric is the positive half of the Laplace distribution with $\mu = 0$, and any $\mathbf{X}_t$ outside of the confidence interval is marked as anomalous.

Experimental Results

To show the efficacy of the proposed method, the aforementioned standard PCA-based approach is used as well. The results obtained using our proposed transfer learning method are then compared against this PCA-based solution.

Real-World Dataset

Robot $\mathcal{A}$ Three signals (speed, position, torque) of the fourth axis of robot $\mathcal{A}$ were collected for 190 consecutive days. The dataset on each day is 3 seconds long. No break down or anomaly was reported for this axis of the robot; however, the task of the robot changed three times during these 190 days. After pre-processing, the dataset on day 1 is used as the training data (healthy robot) and the respective STFT values constitute the extracted features $\mathbf{X}_s$. The datasets of the following days are used as the test dataset to form $\mathbf{X}_t$. $\delta(t)$ is calculated for each day with respect to the first day to identify any changes with respect to the first day. The assumption is that the robot is healthy on the first day, which can be the time of inspection or commissioning at the robot manufacturing facility. Figure 2(a) shows $\delta(t), \ t \in \{1, 2, \ldots, 190\}$. It is apparent in this figure that none of the changes in the task of the robot is identified as anomalous behavior when using our proposed method based on transfer learning (domain adaption). On the other hand, Figure 2(b) shows a different picture when using PCA projection. All three changes in the task of the robot are obvious in this figure. The resulting changes in the computed distances indicate that the use of conventional methods such as PCA fails to distinguish between a change in the mechanical condition of the robot from a change in its task. A second observation when comparing these two figures is the magnitude of the distance function $\delta(t)$. The $\delta(t)$ values in Figure 2(b) are significantly larger than those of Figure 2(a). The larger distances provide further evidence that $\mathbf{X}_s$ and $\mathbf{X}_t$s are not in the same subspace. On the other hand, the small values of $\delta(t)$ in Figure 2(a) indicate that their projection onto the common subspace renders them comparable.

Robot $\mathcal{B}$ In the second experiment, three signals of the fifth axis of robot $\mathcal{B}$ were collected for 148 days. The dataset on each day is again 3 seconds long as in the first experiment. The axis of the robot broke down on the 149th day.
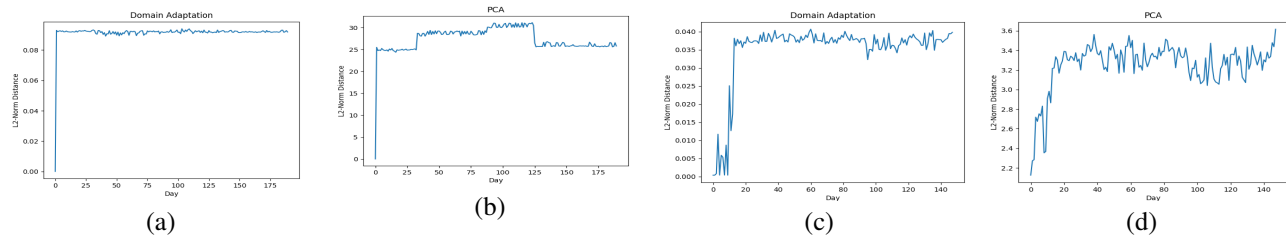
Figure 2: **(a) and (b):** Robot $\mathcal{A}$: first experiment using a 190-day data trace without any anomalies and three changes in the task of the robot. (a) Distance values over 190 days calculated using the proposed method. (b) Distance values over 190 days calculated using the PCA-based method. Three changes in the task are apparent on the 30th, 87th, and 125th day in (b), whereas there is no discernible change in the distance in (a). **(c) and (d):** Robot $\mathcal{B}$: second experiment using a 148-day data trace with an anomaly on the last day and no change in the task of the robot. (c) Distance values over 148 days calculated using the proposed method. (d) Distance values over 148 days calculated using the PCA-based method. The anomaly is apparent on the last day in both (c) and (d) but the relative change is more pronounced using the proposed method.

However, the task of the robot never changed during the entire 148 days. The training and test data sets are constructed as before: The first day is considered the training data and the features are extracted from this dataset as $\mathbf{X}_s$. The extracted features of the other 147 days form $\mathbf{X}_t$. The test data is compared to the training data using the ℓ_2-norm to obtain the distance $\delta(t)$ at time t. Figure 2(c) shows the computed distances of the proposed algorithm. As in the first experiment, PCA is also used to build the subspace of the healthy robot, and the resulting distance function is depicted in Figure 2(d). According to the two plots in Figure 2, both methods perform equally well in identifying the trend of $\delta(t)$ leading to the breakdown of the axis on the 149th day. However, comparing the $\delta(t)$ values of both plots demonstrates the training and test data are being compared in the same subspace in the proposed algorithm in contrast to the PCA-based method in Figure 2(d). As a result, there is a significantly sharper relative increase in the distance around and on the day the anomaly occurred when using our proposed method, implying it can more distinctly identify anomalies.

Conclusion

The challenge with the application of predictive maintenance systems in robotics is that the model requires to be retrained every time the task of the robot changes using conventional methods. We showed that transfer learning (domain adaptation) addresses this challenge. While we used manifold alignment algorithm to project the features extracted from the training and test data onto a common subspace, it is worth investigating other unsupervised transfer learning algorithms. We used distance-based methods to compare the test data with the training data in the common subspace to identify any anomaly. Future work will consider time-series analysis in the common subspace to capture the system dynamics.

References

Boucher, T.; Carey, C.; Mahadevan, S.; and Dyar, M. D. 2015. Aligning mixed manifolds. In *AAAI*, 2511–2517.

Fritsch, F. N., and Carlson, R. E. 1980. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis* 17(2):238–246.

Grall, A.; Dieulle, L.; Bérenguer, C.; and Roussignol, M. 2002. Continuous-time predictive-maintenance scheduling for a deteriorating system. *IEEE transactions on reliability* 51(2):141–150.

Hashemian, H. M. 2010. State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement* 60(1):226–236.

Hornung, R.; Urbanek, H.; Klodmann, J.; Osendorfer, C.; and Van Der Smagt, P. 2014. Model-free robot anomaly detection. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 3676–3683.

Mobley, R. K. 2002. *An introduction to predictive maintenance*. Elsevier.

Moisescu, M. A.; Sacala, I. S.; Dumitrache, I.; and Caramihai, S. 2018. Retracted: Predictive maintenance and robotic system design. *Journal of Fundamental and Applied Sciences* 10(4S):234–239.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22(2):199–210.

Park, D.; Erickson, Z.; Bhattacharjee, T.; and Kemp, C. 2016. Multimodal execution monitoring for anomaly detection during robot manipulation. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 407–414.

Pinto, R., and Cerquitelli, T. 2019. Robot fault detection and remaining life estimation for predictive maintenance. *Procedia Computer Science* 151:709–716.

Sathish, V.; Orkisz, M.; Norrlof, M.; and Butail, S. 2019. Data-driven gearbox failure detection in industrial robots. *IEEE Transactions on Industrial Informatics*.

Sjöstrand, N.; Blanc, D.; and Lindin, A. 2012. Dynamic maintenance plan for an industrial robot. US Patent 8,185,346.

Wang, C., and Mahadevan, S. 2009. A general framework for manifold alignment. In *AAAI Fall Symposium: Manifold Learning and Its Applications*.