2019 IEEE Conference on Control Technology and Applications (CCTA)
Hong Kong, China, August 19-21, 2019

Domain Adaptation Based Fault Detection in Label Imbalanced Cyberphysical Systems

B. Taskazan[1] J. Miller[1] U. Inyang-Udoh[2] O. Camps [1] M. Sznaier[1]

Abstract— **In this paper we propose a data-driven fault detection framework for semi-supervised scenarios where labeled training data from the system under consideration (the "target") is imbalanced (e.g. only relatively few labels are available from one of the classes), but data from a related system (the "source") is readily available. An example of this situation is when a generic simulator is available, but needs to be tuned on a case-by-case basis to match the parameters of the actual system. The goal of this paper is to work with the statistical distribution of the data without necessitating system identification. Our main result shows that if the source and target domain are related by a linear transformation (a common assumption in domain adaptation), the problem of designing a classifier that minimizes a miss-classification loss over the joint source and target domains reduces to a convex optimization subject to a single (non-convex) equality constraint. This second-order equality constraint can be recast as a rank-1 optimization problem, where the rank constraint can be efficiently handled through a reweighted nuclear norm surrogate. These results are illustrated with a practical application: fault detection in additive manufacturing (industrial 3D printing). The proposed method is able to exploit simulation data (source domain) to substantially outperform classifiers tuned using only data from a single domain.**

I. INTRODUCTION

Successful use of cyber-physical systems (CPS) in critical applications hinges on the ability to detect and isolate faults. Traditionally, fault detection methods can be divided into two types: (i) model based and (ii) data-driven approaches [21]. **Model Based** rely on a model of the system to identify anomalous outputs (for instance by designing a filter whose output is small under nominal conditions and large in the presence of faults [24]). While this class of methods has proven to be very successful, their dependence on models can prevent their use in many CPS applications, such as 3D printing, that rely on complex processes where models are a-priori unknown and must be estimated and validated through a computationally expensive identification step. As an alternative, **data-driven** methods, rooted in Machine Learning ideas, exploit the statistical properties of the data to detect anomalies (e.g. data points whose probability of originating from a non-faulty system is low). An extensive survey on existing approaches from the machine learning community can be found in [14]. Additional recent contributions include [15] and [16] which use sum-of-squares methods to derive a statistical model for error detection, and [13] which uses moments to determine the presence of an error using on-line sensor readings. These methods are computationally attractive as they avoid explicitly identifying the plant, and the process of training often reduces to a convex optimization problem. However, designing a data-driven classifier typically requires a large amount of labeled training data. This requirement can be problematic in situations where generating this data can be costly or may require operating the plant in unsafe conditions.

To circumvent these difficulties, we propose a new approach to fault detection for scenarios where labeled training data from the actual system under consideration is highly imbalanced (labels from one class are scarce), but labeled data from a related system is readily available. Examples of this situation include cases where extensive experiments can be performed on a prototype of the plant, or where a generic simulator is available but tuning is needed to match each actual system. Under the assumption that the statistical distributions of the data generated by the surrogate and actual systems can be mapped by a linear (or a composition of polynomial and linear) transformation, our main result shows that this transformation and an optimal classifier can be found by solving a convex optimization problem subject to a single non-convex norm equality constraint. Further, efficient solutions to this problem, with optimality certificates, can be found by first recasting it into a rank-constrained optimization, which in turn can be relaxed to a convex optimization by using nuclear norm proxies. This technique is illustrated in the problem of fault-detection in a 3D printer, using a combination of data generated by a simulator (source) and data from an actual printer (target).

The main contributions of the paper are:

- A new domain-adaptation based framework for semi-supervised fault detection in cyberphysical systems, specifically tailored to scenarios where only a small amount of labeled faulty data generated by the actual plant is available
- Theoretical results showing that the problem above can be reduced to a convex optimization subject to a single norm equality type constraint. Moreover, this problem can be relaxed to an efficient convex optimization, with optimality certificates.
- Application of these results to the problem of detecting faults in additive manufacturing.

The paper is organized as follows. In section II we introduce the notation used in this paper, together with some required background results in domain adaptation and

This work was supported in part by NSF grants CNS–1646121, CMMI–1638234 and ECCS–1808381, and AFOSR grant FA9550-15-1-0392. [1] ECE Department, Northeastern University, Boston, MA 02115. emails: {taskazan.b,miller.jare}@husky.neu.edu, {camps,msznaier}@coe.neu.edu, [2] Mechanical Engineering, RPI, Troy, NY 12080, email: inyanu@rpi.edu.

soft-margin support vector machines (SVMs). In section III we precisely state the problem under consideration and show that the design of an SVM classifier that minimizes a misclassification measure over all the available data reduces to a convex optimization problem. Section IV applies these results to a practical non-trivial problem, fault detection in additive manufacturing. Finally, section V summarizes our results and points out to directions for further research.

II. Preliminaries

In this section we introduce the notation used in the paper and some background results on domain adaptation and support vector machines (SVM) based classification.

A. Notation

$\mathbf{M}$	matrix with elements M_{ij}
$\mathbf{M}(i:i+m, j:j+n)$	$(m+1)\times(n+1)$ submatrix of $\mathbf{M}$ with M_{ij} in its upper left corner
$\mathbf{M}^T$	transpose of matrix $\mathbf{M}$
$\|\mathbf{M}\|_*$	nuclear norm of $\mathbf{M}$
$\|\mathbf{M}\|_F$	Frobenious norm of $\mathbf{M}$
$\sigma_r(\mathbf{M})$	r^{th} singular value of $\mathbf{M}$
$\mathbf{M} \succeq 0$	$\mathbf{M}$ is positive semi-definite
$\|\mathscr{S}\|$	cardinality of the set $\mathscr{S}$
$loss(\mathbf{M}, b)$	total hinge loss parametrized over a matrix and a scalar

B. Domain Adaptation and Covariance Alignment

Domain adaptation (DA) is a set of statistical methods that leverage labeled data from one domain - source - to perform a task (such as inference) in a related domain - target - where few labeled data points are available. Briefly, the underlying idea is to reduce the performance degradation of models designed using data from a single domain by minimizing the statistical gap between domains. Roughly, DA methods can be partitioned into unsupervised (those that use only labels from the source domain), and semi-supervised which take advantage of a few labeled samples from the target domain to increase performance as compared to unsupervised learning. Some common approaches to the adaptation problem are distance minimization [10], [11], [8], subspace mapping [12], [7], [6], [9] and correlation alignment [5], [6]. Among these methods, [5], [12], [9] and [7] are unsupervised methods while the rest can be extended to semi-supervised learning.

The approach that we pursue in this paper is inspired by [5], where the authors propose to solve domain adaptation problems by finding a linear transformation that minimizes the distance between Σ_s and Σ_t, the covariances of the source and target data, respectively. Under the assumption that the source and target data are related by a linear transformation $\mathbf{A}$ (that is $\mathbf{x}_t = \mathbf{A}\mathbf{x}_s$), this leads to the following optimization problem:

$$\begin{aligned} \mathbf{x}_t = \mathbf{A}\mathbf{x}_s \rightarrow \Sigma_t = \mathbf{A}\Sigma_s\mathbf{A}^T = \Sigma_s' \\ \min_{\mathbf{A}} ||\Sigma_s' - \Sigma_t||_F^2 = \min_{\mathbf{A}} ||\mathbf{A}\Sigma_s\mathbf{A}^T - \Sigma_t||_F^2 \end{aligned} \tag{1}$$

It can be easily seen that all the solutions to the problem above are of the form

$$\mathbf{A} = \Sigma_t^{\frac{1}{2}}\mathbf{U}\Sigma_s^{\frac{-1}{2}} \text{ s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbf{I} \tag{2}$$

where the unitary matrix $\mathbf{U}$ is a free parameter. In the absence of additional information about the target labels, the original formulation in [5] simply selects $\mathbf{U} = \mathbf{I}$, which intuitively can be interpreted as whitening the source data with its covariance and recoloring it with the target's covariance. In contrast, in this paper we will consider a semi-supervised scenario where a few target labels are provided and use the additional degrees of freedom provided by $\mathbf{U}$ to optimize classification performance. The advantages of this approach are illustrated in Fig. 1 with a simple two class case, where the source and target domains are linearly related but have different covariances as shown in Fig. 1(a) and Fig. 1(b). Fig. 1(c) shows the results of using a classifier trained with source data points transformed using standard covariance alignment (e.g taking $\mathbf{U} = \mathbf{I}$ in (2)). As illustrated there, this classifier performs poorly on the target domain (see Fig. 1(b)). On the other hand, using the additional degrees of freedom provided by $\mathbf{U}$, leads to virtually perfect classification (Fig. 1(d)).

C. Soft Margin SVM Classifiers

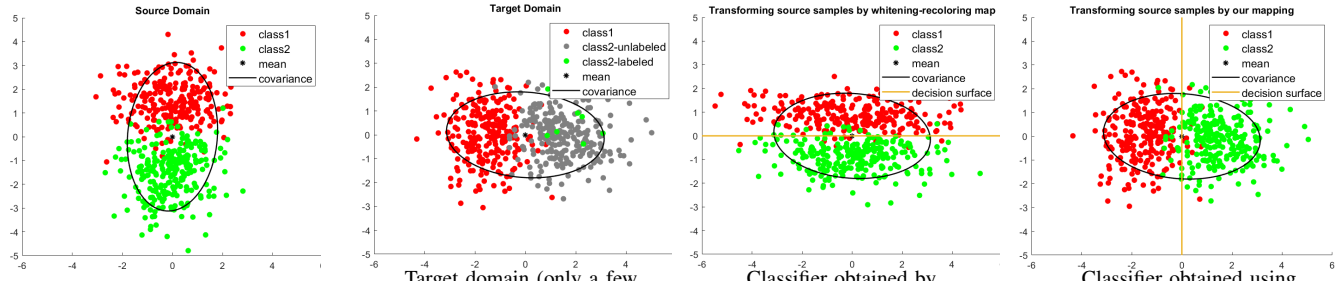
SVMs are supervised binary classifiers which categorize test examples according to which side of a given hyperplane they fall. In the case of linearly separable data, so called hard margin SVMs learn a hyperplane lying halfway between the two parallel hyperplanes supporting the two categories of training data. If the data is not necessarily linearly separable, as in many real world problems, the following soft margin formulation [22], [23] is widely used: given training samples $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1,...,l$, and labels $\mathbf{y} \in \mathbb{R}^l$ where $y_i \in \{1,-1\}$, find a hyperplane with normal $\mathbf{w}$ and offset b to minimize the following cost.

$$\begin{aligned} &\min_{\mathbf{w},b,\varepsilon_i} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\varepsilon_i \\ &\text{subject to } y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \varepsilon_i \\ &\varepsilon_i \geq 0, i = 1,...,l \end{aligned} \tag{3}$$

Here $\phi(\mathbf{x}_i)$ maps the data vectors to a higher dimensional space where the data is (close to) linearly separable, resulting in a nonlinear classifier. For linear SVMs, $\phi(\mathbf{x}_i) = \mathbf{x}_i$. Here $C > 0$ is an hyperparameter determining the trade-off between classification accuracy and size of margin. If C is large enough so that the regularization term $\frac{1}{2}||w||^2$ is negligible, then the soft margin formulation above behaves like a hard margin classifier. Since the dimension of the vector $\mathbf{w}$ depends on the dimension of the data vectors, in cases where this dimension is high, a dual formulation of (3) is used [25]. However, this formulation cannot directly be used in conjunction with the DA proposed here.

III. Semi-Supervised Domain Adaptation for Fault Detection and Classification

The goal of this paper is to develop an efficient fault detection algorithm for cyber-physical systems operating in



(a) Source domain ground truth

(b) Target domain (only a few labels from the second class are known)

(c) Classifier obtained by conventional covariance alignment (unsupervised)

(d) Classifier obtained using the proposed method (semi-supervised)

Fig. 1. A toy example: The goal is to classify target data (b) using source information (a) and labeled target samples. Note that only a few samples from the second class are available. Covariance alignment performs poorly ((c) vs (b)) while the proposed method has almost perfect performance ((d) vs (b)).

scenarios where there is imbalanced labeled training data generated by the actual plant under consideration, but where training data from a related system is readily available. An example of this situation are cases where reasonably high-fidelity simulations of the CPS under consideration are available, but need to be tuned on a case-by-case basis to match the parameters of the actual system. Rather than performing a costly non-linear identification of the system, the approach proposed here works directly with the statistical distribution of the data, by designing a classifier that leverages both the labels from the actual system (the "target" domain) and those obtained from the simulator (in this case the "source").

A. Design of a Domain-Adapted Classifier

In this paper, we will use as classifiers the well known soft margin SVMs described in section II-C. This choice is motivated by the fact that, in addition to their proven success, training of these classifiers reduces to a convex optimization problem that, as we show in the sequel, can be modified to handle the semi-supervised scenario of interest here. Note that in principle, this choice assumes that the different classes are linearly separable. However, more complex cases can be handled by simply lifting the data to a higher dimensional space using kernel methods. As briefly discussed in section V, a similar idea can be used to extend the domain adaptation based approach presented here to non-linearly separable classes.

In the sequel we consider for simplicity two-class scenarios (e.g. faulty/non-faulty CPS), but the approach can be easily generalized to the multi-class case by a straightforward modification of the objective function. In this context, the problem of interest here can be formally stated as:

Problem 1: Given

(i) source ($\mathscr{X}_s$) and target ($\mathscr{X}_t$) training data sets, with covariances Σ_s and Σ_t

(ii) labeled subsets $\mathscr{X}_{s,l} \subset \mathscr{X}_s$ and $\mathscr{X}_{t,l} \subset \mathscr{X}_t$, with $N_s \doteq |\mathscr{X}_{s,l}|$, $N_t \doteq |\mathscr{X}_{t,l}|$, $N_s \gg N_t$

find a transformation $\mathbf{A}$: source $\rightarrow$ target and a SVM based classifier that minimizes the classification loss function over the joint labeled data set $\mathscr{X}_l \doteq \mathbf{A}\mathscr{X}_{s,l} \cup \mathscr{X}_{t,l}$.

Using the primal hinge loss formulation of linear SVMs described in section II-C leads to the following explicit expression for minimizing the total loss over $\mathscr{X}_l$:

$$
\begin{aligned}
(\mathbf{w}^*, b^*, \mathbf{U}^*) = \underset{\mathbf{w},b,\mathbf{U}}{\operatorname{argmin}} \; & C\sum_{i=1}^{N_s} max(0, 1 - y_{s,i}(\mathbf{w}^T\mathbf{A}\mathbf{x}_{s,i} + b)) \\
& + \sum_{j=1}^{N_t} C_y \, max(0, 1 - y_{t,j}(\mathbf{w}^T\mathbf{x}_{t,j} + b)) + \frac{1}{2}\|\mathbf{w}\|_2^2 \\
\text{subject to } & \mathbf{U}^T\mathbf{U} = \mathbf{I} \text{ and } \mathbf{A} = \Sigma_t^{\frac{1}{2}}\mathbf{U}\Sigma_s^{-\frac{1}{2}}
\end{aligned}
\tag{4}
$$

where $\mathbf{x}_{s,i}, \mathbf{x}_{t,i}, y_{s,i}$ and $y_{t,i}$ denote the elements of $\mathscr{X}_{s,l}$ and $\mathscr{X}_{t,l}$ and the corresponding labels, respectively. Since the target training set is unbalanced in terms of faulty and nominal class samples, a different value of the hyper parameter $C_y = \{C_1, C_{-1}\}$ is chosen for each class, with a higher penalty for the faulty class. Note that the optimization problem above is a convex optimization problem over the Stiefel manifold defined by $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. However, since the objective function is non-differentiable, standard manifold optimization techniques (e.g. curvilinear gradient descent, Riemannian trust-region) [18] are not directly applicable. In principle this difficulty can be circumvented by using alternative formulations where the objective is smooth, but this entails introducing an additional variable and an associated inequality constraint for each data point in (4), substantially increasing the computational complexity. As an alternative, in this paper we will exploit the fact that (4) is a semi-algebraic optimization problem [17] that can be efficiently solved using moments based techniques. We begin by rewriting (4) in terms of fewer variables, subject to a single (non-convex) norm constraint. Define

$$
\begin{aligned}
\mathbf{z}_t &\doteq (\Sigma_t^{\frac{1}{2}})^T\mathbf{w} \\
\mathbf{z}_s &\doteq \mathbf{U}^T\mathbf{z}_t
\end{aligned}
\tag{5}
$$

Rewriting (4) in terms of these new variables leads to:

$$
\begin{aligned}
\min_{\mathbf{z}_s,\mathbf{z}_t,b} L(\mathbf{z}_s, \mathbf{z}_t, b) \doteq \; & C\sum_{i=1}^{N_s} max(0, 1 - y_{s,i}(\mathbf{z}_s^T\Sigma_s^{-\frac{1}{2}}\mathbf{x}_{s,i} + b)) \\
& + \sum_{j=1}^{N_t} C_y \, max(0, 1 - y_{t,j}(\mathbf{z}_t^T\Sigma_t^{-\frac{1}{2}}\mathbf{x}_{t,j} + b)) + \frac{1}{2}\mathbf{z}_t^T\Sigma_t^{-1}\mathbf{z}_t \\
\text{subject to } & \|\mathbf{z}_s\|_2^2 = \|\mathbf{z}_t\|_2^2
\end{aligned}
\tag{6}
$$

where we exploit the fact that $\mathbf{U}^T\mathbf{U}=\mathbf{I} \iff \|\mathbf{z}_s\|_2^2=\|\mathbf{z}_t\|_2^2$.

Since the objective function in (6) is semi-algebraic and the constraint is polynomial, in principle a sequence of convex relaxations whose solution is guaranteed to converge to the optimal can be obtained by proceeding as in [17]. While this approach works well for small size problems, it quickly becomes intractable due to the combinatorial growth of the number of variables as the order of the relaxation increases. As an alternative, in this paper we propose to reformulate (6) as a rank-constrained optimization that in turn can be relaxed to a convex optimization by using the nuclear norm surrogate for rank.

Theorem 1: Problem (6) is equivalent to the following rank-constrained optimization:

$$\begin{aligned}
&\min_{b,\mathbf{M}\succeq 0} \text{loss}(\mathbf{M},b) \doteq C\sum_{i=1}^{N_s} max(0,1-y_{s,i}(\mathbf{m}_s^T\Sigma_s^{-\frac{1}{2}}\mathbf{x}_{s,i}+b)) \\
&+\sum_{j=1}^{N_t} C_y\, max(0,1-y_{t,j}(\mathbf{m}_t^T\Sigma_t^{-\frac{1}{2}}\mathbf{x}_{t,j}+b))+\frac{1}{2}\mathbf{m}_t^T\Sigma_t^{-1}\mathbf{m}_t \\
&\text{subject to} \\
&M_{11}=1 \\
&\sum_{i=2}^{n+1} M_{ii}=\sum_{j=n+2}^{2n+1} M_{jj} \\
&\text{rank}(\mathbf{M})=1
\end{aligned} \tag{7}$$

where $\mathbf{m}_s \doteq \mathbf{M}(1,2:n+1)$ and $\mathbf{m}_t \doteq \mathbf{M}(1,n+2:2n+1)$.

Proof: Denote by $\mathbf{z}_s^*,\mathbf{z}_t^*,b_1^*$ and $\mathbf{M}^*,b_2^*$ the solutions to (6) and (7) respectively. Define $\mathbf{v}^T \doteq \begin{bmatrix}1 & \mathbf{z}_s^{*T} & \mathbf{z}_t^{*T}\end{bmatrix}$ and $\hat{\mathbf{M}} \doteq \mathbf{v}\mathbf{v}^T$. By construction, $\text{loss}(\hat{\mathbf{M}},b_1^*)=L(\mathbf{z}_s^*,\mathbf{z}_t^*,b_1^*)$ and $\hat{\mathbf{M}}$ is a feasible solution of (7). Hence

$$\text{loss}(\mathbf{M}^*,b_2^*) \le \text{loss}(\hat{\mathbf{M}},b_1^*)=L(\mathbf{z}_s^*,\mathbf{z}_t^*,b_1^*)$$

Since the optimal solution $\mathbf{M}^*$ to (7) has rank 1 and $M_{11}=1$, $\mathbf{M}$ can be factored as $\mathbf{M}^*=\mathbf{v}\mathbf{v}^T$ with $\mathbf{v}=\begin{bmatrix}1 & \mathbf{m}^T\end{bmatrix}^T$. Let $\hat{\mathbf{z}}_s=\mathbf{m}(1:n)$ and $\hat{\mathbf{z}}_t=\mathbf{m}(n+1:2n)$. By construction $(\hat{\mathbf{z}}_s,\hat{\mathbf{z}}_t)$ are feasible for (6) and

$$L(\mathbf{z}_s^*,\mathbf{z}_t^*,b_1^*) \le L(\hat{\mathbf{z}}_s,\hat{\mathbf{z}}_t,b_2^*)=\text{loss}(\mathbf{M}^*,b_2^*)$$

Hence $L(\mathbf{z}_s^*,\mathbf{z}_t^*,b_1^*)=\text{loss}(\mathbf{M}^*,b_2^*)$. ■

Remark 1: Recall that an $n\times n$ symmetric matrix has $\frac{n(n+1)}{2}$ distinct entries. Thus, $\mathbf{M}$ in (6) involves $(2n+1)(n+1)$ variables, since its first row is $\begin{bmatrix}1 & \mathbf{z}_s & \mathbf{z}_t\end{bmatrix}$. On the other hand, in the original formulation (4) the first row of $\mathbf{M}$ is given by $\begin{bmatrix}1 & \mathbf{w} & \text{vec}(\mathbf{U})\end{bmatrix}$, and hence $\mathbf{M}$ has $\frac{(n^2+n+1)(n^2+n+2)}{2}$ degrees of freedom. Thus, (6) leads to substantial computational complexity reduction in scenarios where n is not small, that is $\mathbf{x}$ has a large number of features.

B. A Convex Relaxation

In principle, (7) is generically NP-hard, due to the rank constraint. A tractable convex relaxation can be obtained by replacing rank by its convex envelope, the nuclear norm [19], leading to Algorithm 1 outlined below. Briefly, the main idea here is to seek rank 1 solutions to (7) by minimizing the nuclear norm of $\mathbf{M}$ (using the re-weighted heuristic proposed in [19]), subject to the constraint $\text{loss}(\mathbf{M},b) \le \mu$. The algorithm can then be used to perform a simple line search on μ to find the lowest value that yields rank 1 solutions[1]. Alternatively, μ can be easily tuned on the validation set provided by the label-rich source domain.

Algorithm 1 Domain Adaptive SVM Training

Input: *Source*: source domain data, *Target*: target domain data, μ: upper bound on loss, *Maxit*: maximum number of iterations, d: Space dimension

$Xsr \leftarrow PCA_d(Source)$, $Target \leftarrow PCA_d(Target)$
X_{tr} : Few target samples as training instances
X_{tt} : The rest of the target samples as testing instances
$\Sigma_t \leftarrow cov(X_{tt} \cup X_{tr})$, $\Sigma_s \leftarrow cov(X_{sr})$
Initialization : $\mathbf{W}=\mathbf{I}_{(2d+1)\times(2d+1)}$
for $i=1$ to *Maxit* **do**
  minimize $J(\mathbf{M}) \doteq \text{Trace}(\mathbf{W}\cdot\mathbf{M})$
  subject to:
  $\mathbf{M}(1,1)=1$
  $\mathbf{M}\succeq 0$
  $loss(\mathbf{M},b) \le \mu$
  $\sum_{i=2}^{n+1} M_{ii}=\sum_{j=n+2}^{2n+1} M_{jj}$
  if rank(**M**)=1 **then**
    break
  else
    $\sigma_2 \leftarrow$ second singular value of $\mathbf{M}$
    $\mathbf{W}=(\mathbf{M}+\sigma_2\mathbf{I})^{-1}$
  end if
end for
return **M**, b
$b_t \leftarrow b$, $\mathbf{w}_t \leftarrow$from elements of $\mathbf{M}$

Output: $(\mathbf{w}_t,b_t)$: SVM parameters

C. Further Computational Complexity Reduction

Further computational reduction can be achieved by noting that only the elements in the first row and column of $\mathbf{M}$ and its diagonal appear explicitly in (7), while the other elements are only needed to guarantee that $\mathbf{M}\succeq 0$ (block-arrow sparsity pattern). Moreover, from Theorem 1.5 in [20], there exists a rank 1 positive semi-definite completion of $\mathbf{M}$ iff $\text{rank}(\mathbf{M}_i)=1, i=1,..2n$, where $\mathbf{M}_i \doteq \begin{bmatrix}1 & M_{1i}\\ M_{1i} & M_{ii}\end{bmatrix}$. Thus, a computationally attractive alternative to Algorithm 1 can be obtained by replacing the objective with $J_s(\mathbf{M}_i) \doteq \sum \text{Trace}(\mathbf{W}_i\mathbf{M}_i)$ and the constraint $\mathbf{M}\succeq 0$ with $\mathbf{M}_i\succeq 0$, $i=1,..2n$. Note that this formulation has only $6n$ variables, compared to $(2n+1)(n+1)$ in (7). Since the computational complexity of interior point methods scales as (number of variables)^3, the reduction is substantial, even for moderately sized feature vectors.

[1] A lower bound on μ can be found by simply solving (7) without the rank constraint.

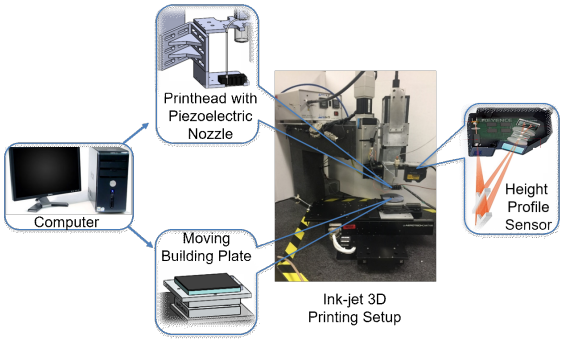
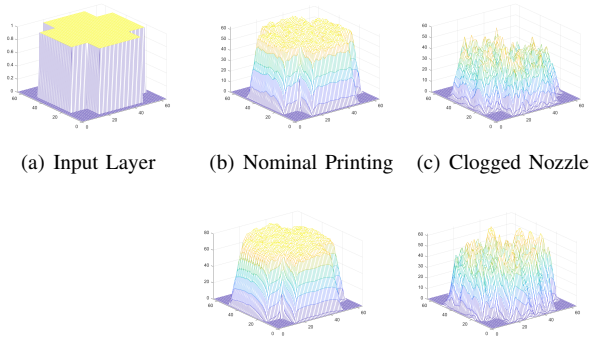


Fig. 2. Ink-jet 3-D Printing System. Printer builds parts in layers by ejecting droplets from a nozzle onto a moving substrate. Each ejected layer of droplets is cured and the resulting height profile is measured.



(a) Input Layer (b) Nominal Printing (c) Clogged Nozzle

(d) Nominal Printing (e) Clogged Nozzle

Fig. 3. Height Profile of a Data Point on Two Different Simulator Domains: (b), (c) correspond to Sim1; (d), (e) correspond to Sim2.

IV. Application: Fault Detection in Additive Manufacturing

In this section we illustrate the advantages of the proposed framework on a non trivial problem: fault detection on a layer-to-layer ink-jet 3-D printer. These printers (shown in Fig. 2) build the desired parts by depositing droplets on locations determined by the input layer droplet map [1], [2], [3], [4]. Printers typically have an open-loop control scheme assuming constant droplet density, and corrections are made by refining layers after deposition. Uncertainty in the building process (such as droplet spread and thermal differences) cause uneven height profiles and inhibit development of a reliable purely physics-based model relating the input droplet map and the output. As a result, physics-based simulators are not exact and fault detection algorithms designed using simulation data alone are not reliable. We show that domain adaptation can leverage a small amount of real data, together with simulations, to substantially improve the performance of data-driven fault detection algorithms. Using simulated training data obviates the need for generating the various kinds of faults necessary for diagnostics, which may necessitate operating under hazardous conditions that will generally degrade or destroy existing machines. This study focuses on the abnormal condition of a clogged printing nozzle. We aim to achieve early termination of printing when a clogged nozzle is detected on a layer using a fault detector designed using simulator data (source domain) adapted to the real printer (target). In the sequel, we consider two different experiments. In the first one, in order to explore fault detection accuracy of our method on a fairly large testing data set, we conduct our experiments on two different simulators. The simulators were developed using the graph-based model presented in [3]. Since both simulators produce faulty and nominal output layers that are statistically different from each other (see Fig. 3), we will treat one of these simulators as a surrogate for a real printer. In the second experiment we use one of these simulators together with nominal and faulty data generated by an actual ink-jet printer, where the number of faulty data points is substantially smaller than the one corresponding to nominal conditions.

A. Simulator-to-Simulator Adaptation

In this experiment, we have 2048 data points (output layers) from each simulator. While all label information is available in the source domain (Sim1), we assume that label information from the target domain is imbalanced: Only 5% of the faulty data labels are available in the second simulator. For experiments, 2/3 of label-rich class samples are used during training along with few labeled data from the faulty class, and the rest is used for testing. In principle, each data point $\mathbf{x} \in \mathbb{R}^{4096}$, corresponding to a vectorized 64x64 output layer profile. However, the high amount of redundancy in the data (due to the geometric constraints of the process) allows for reducing the data to dimension as low as 20 using PCA, without significant performance decrease. Table I shows classifier's performance using three different metrics: precision $= \frac{\text{tp}}{\text{tp+fp}}$, recall $= \frac{\text{tp}}{\text{p}}$, and F1 $= 2\frac{\text{precision} \cdot \text{recall}}{\text{precision+recall}}$, where p=(number of actual faults), tp=(number of actual faults found by the algorithm), and fp=(non-faulty data mislabeled as a fault). We compared our results against the baseline of a regular linear SVM trained on the source domain. Its performance on the source domain validation set (Sim1) and on the target test set (no adaptation case - NA) are given to illustrate the performance degradation caused by the difference in statistical properties across domains. In the third rows, performance of a linear SVM trained only on labeled target samples is given (Tgt). Finally, in the fourth rows, we report the performance of SVM with adaptation (WA). As shown there, accuracy of a regular SVM trained only on the source domain decreases with respect to all three metrics when it performs on the target. Training the classifier only on the target domain also results poorly - half of the faults cannot be recalled - since the number of data points available for training is not enough for a good generalization. On the other hand, the SVM-WA proposed here outperforms both baseline systems in terms of recall and f1-score metrics.

B. Simulator-to-Printer Adaptation

In this section we tested our algorithm in a more challenging real setting. Using Sim1 as the source domain, we use actual prints as the target domain data points. We have

497 nominal and 21 faulty data points available from the actual ink-jet 3D printer. Note that these faults are not only from clogged nozzle condition as we are simulating on Sim1 domain but from different abnormal cases that may occur during building (slanted base, omitted droplets, sparse/close spacing). We synthesized simulation outputs by using the same set of input maps used for real printing. We used 10 randomly picked faulty prints for training, and the rest is for testing. As shown in Table I, the non-adapted classifier is not able to detect any faults and results in a zero recall rate, while the domain adapted algorithm proposed here correctly recalls 9 of 11 faulty real printing layers even though fault types except clogged nozzle aren't available from Sim1. This is because our algorithm leverages nominal data points of the real printer, so the relation between domains can be linearly approximated by a mapping between nominal classes.

TABLE I
FAULT DETECTION RESULTS (SOURCE-TO-TARGET)

	Sim1-to-Sim2		Sim1-to-Printer	
F1Score	Sim1	0.9052	Sim1	0.8876
F1Score	Sim2 (NA)	0.8712	Printer (NA)	NaN
F1Score	Sim2 (Tgt)	0.6671	Printer (Tgt)	0.4706
F1Score	Sim2 (WA)	0.9535	Printer (WA)	0.7826
Precision	Sim1	0.8767	Sim1	0.8721
Precision	Sim2 (NA)	0.8804	Printer (NA)	NaN
Precision	Sim2 (Tgt)	0.9919	Printer (Tgt)	0.6667
Precision	Sim2 (WA)	0.9374	Printer (WA)	0.7500
Recall	Sim1	0.9357	Sim1	0.9036
Recall	Sim2 (NA)	0.8623	Printer (NA)	0.00
Recall	Sim2 (Tgt)	0.5026	Printer (Tgt)	0.3636
Recall	Sim2 (WA)	0.9702	Printer (WA)	0.8182

V. CONCLUSIONS

In this paper we presented a domain-adaptation based technique capable of leveraging a small amount of labeled data generated by the real system together with data generated by an untuned simulator. Our main result showed that an SVM based classifier capable of exploiting data from both the source and target domain can be obtained by solving an optimization problem subject to a single non-convex constraint. Further, this problem can be relaxed to a convex optimization and efficiently solved. These results were illustrated using as an example detection of a clogged nozzle on an ink-jet 3D printer, where the proposed method outperforms classifiers designed using only source or target data.

In principle, since we used a linear SVM, the results presented in the paper apply only to the case where the nominal and faulty data are linearly separable. Results presented here can be trivially extended to polynomial kernel SVMs (where the data is polynomially lifted to a space where the classes are linearly separable) by simply considering the lifting $\mathbf{x} \to \mathbf{v}(\mathbf{x})$, where the elements of $\mathbf{v}(\mathbf{x})$ are monomials of the form $x^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_m^{\alpha_m}$, and applying Algorithm 1 to the Veronese mapped $\mathbf{v}(\mathbf{x})$. Work currently in progress seeks to develop a computationally efficient implementations of Algorithm 1 allowing for applying the proposed technique to large data sets in scenarios where the dimension of the feature vector is not small and cannot be easily reduced without performance loss.

.

ACKNOWLEDGMENT

The authors are indebted to Dr. Biel Roig Solvas for many discussions related to the properties of Algorithm 1.

REFERENCES

[1] Y. Guo and S. Mishra, A predictive control algorithm for layer-to-layer ink-jet 3D printing. *2016 American Control Conference (ACC)*, 2016.
[2] L. Lu, J. Zheng, and S. Mishra. A layer-to-layer model and feedback control of ink-jet 3-d printing. *IEEE / ASME Transactions on Mechatronics* 20.3 : 1056–1068, 2015.
[3] Y. Guo, J. Peters, T. Oomen, and S. Mishra. Control-oriented models for ink-jet 3D printing. *Mechatronics*, Vol. 56, pp. 211–219, 2018.
[4] Y. Guo, J. Peters, T. Oomen, and S. Mishra. Distributed model predictive control for ink-jet 3D printing. *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2017.
[5] B. Sun, J. Feng, and K. Saenko. Return of Frustratingly Easy Domain Adaptation, *AAAI*, pp. 2058–2065, 2016.
[6] S. Herath, M. T. Harandi, and F. Porikli. Learning an invariant hilbert space for domain adaptation. *CVPR*. 2017.
[7] B. Gong, et al. "Geodesic flow kernel for unsupervised domain adaptation." *Computer Vision and Pattern Recognition (CVPR)*, 2012.
[8] K. Saenko, et al. Adapting visual category models to new domains. *European Conference on Computer Vision (ECCV)*, 2010.
[9] B. Fernando, et al. Unsupervised visual domain adaptation using subspace alignment. *IEEE International Conference on Computer Vision (ICCV)*, 2013.
[10] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. *AAAI*. Vol. 8. 2008.
[11] S. J. Pan, et al. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22.2: 199 – 210, 2011.
[12] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. *IEEE International Conference on Computer Vision (ICCV)*, 2011.
[13] R. L. Christensen, K. K. Sorensen, R. Wisniewski, and M. Sznaier. Unsupervised Fault Detection of Reefer Containers: A Moments-Based SDP Approach. *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018.
[14] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey *ACM Comput. Surv.*, 41(3):15:115:58, Jul 2009.
[15] J. B. Lasserre and E. Pauwels. Sorting out typicality with the inverse moment matrix SOS polynomial. *CoRR*, abs/1606.03858, 2016.
[16] J. A. Lopez, M. Sznaier, and O. Camps. Unsupervised fault detection using semidefinite programming. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 3798–3803, Dec 2015.
[17] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optimization*, 11:796–817, 2001.
[18] N. Boumal, B. Mishra, P. A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1), pp.1455–1459, 2014.
[19] M. Fazel, H. Hindi and S. P. Boyd, A rank minimization heuristic with application to minimum order system approximation. *Proceedings of the 2001 American Control Conference*, pp. 4734-4739, 2001.
[20] J. Dancis. Positive semidefinite completion of partial Hermitian matrix. *Linear Algebra and Its Applications*, Vol. 175, 97–114, October 1992.
[21] Z. Gao, C. Cecati, and S. X. Ding. A survey of fault diagnosis and fault-tolerant techniquesPart I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* 62.6: 3757–3767, 2015.
[22] C. Cortes, and V. Vapnik. Support-vector networks. *Machine learning 20.3*: 273–297, 1995.
[23] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM Press, 1992.
[24] M. Zhong, S. X. Ding, J. Lam, and H. Wang. An LMI approach to design robust fault detection filter for uncertain LTI systems. *Automatica*, 39(3), pp. 543–550, 2003.
[25] C. C. Chang and C. J. Lin. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.