Outliers Cleaning in Dynamic Systems

A Thesis Presented
by
Yuexi Zhang
to
The Department of Electrical and Computer Engineering
in partial fulfillment of the requirements
for the degree of
Master of Science
in
Electrical and Computer Engineering

Northeastern University
Boston, Massachusetts

April 2017
To my parents.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>iv</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>List of Acronyms</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>viii</td>
</tr>
<tr>
<td>Abstract of the Thesis</td>
<td>ix</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Related Work</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Thesis Organization</td>
<td>3</td>
</tr>
<tr>
<td><strong>2 Preliminaries</strong></td>
<td>4</td>
</tr>
<tr>
<td>2.1 Hankel Matrix Representations</td>
<td>4</td>
</tr>
<tr>
<td>2.2 The Gram Matrices</td>
<td>5</td>
</tr>
<tr>
<td>2.3 The JensenBregman Log-det Divergence (JBLD)</td>
<td>5</td>
</tr>
<tr>
<td><strong>3 Methods</strong></td>
<td>6</td>
</tr>
<tr>
<td>3.1 Proposed Framework for Cleaning Outliers</td>
<td>6</td>
</tr>
<tr>
<td>3.1.1 Overview of Computing JBLD Values</td>
<td>6</td>
</tr>
<tr>
<td>3.1.2 Steps of Computing JBLD</td>
<td>7</td>
</tr>
<tr>
<td>3.1.3 Data Interpolation and Matrix Completion</td>
<td>8</td>
</tr>
<tr>
<td>3.1.4 Outline of the Proposed Framework</td>
<td>10</td>
</tr>
<tr>
<td>3.2 Supervised Outlier Removal for Dynamic Systems (SORDS)</td>
<td>11</td>
</tr>
<tr>
<td>3.2.1 Optimization Problem</td>
<td>11</td>
</tr>
<tr>
<td>3.2.2 SORDS Algorithm</td>
<td>12</td>
</tr>
<tr>
<td><strong>4 Applications</strong></td>
<td>14</td>
</tr>
<tr>
<td>4.1 Video Experiments</td>
<td>14</td>
</tr>
<tr>
<td>4.1.1 Dataset</td>
<td>14</td>
</tr>
<tr>
<td>4.1.2 Evaluations</td>
<td>14</td>
</tr>
</tbody>
</table>
4.1.3 Comparison for UYDP Dataset .............................................. 15
4.2 Experiments for SORDS Algorithm ........................................ 17
4.2.1 Experiments with Synthetic Data ....................................... 18

5 Conclusions and Future Work .................................................. 24

Bibliography ............................................................................. 25
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Failure Cases. Some of misdetections and outliers may occur when there are occlusions and overlapping parts.</td>
<td>2</td>
</tr>
<tr>
<td>3.1</td>
<td>The flowchart of our framework</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>1st Example Video from UYDP Dataset. (a) Original poses from pose estimator, (b) Poses after proposed framework.</td>
<td>16</td>
</tr>
<tr>
<td>4.2</td>
<td>JBLD values of the joint with outliers in the 1st example. We denoted two joints as the right elbow and the right wrist based on the pose estimator [1]. The yellow curve is from results of the original detections, and other two are from the cleaned trajectory. From results, the curve indicated that after our proposed framework, outliers will be removed and then return a promising recovery.</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>JBLD values of the joint with outliers in the 2nd example. We denoted this joint as left wrist based on the pose estimator [1]. The yellow curve is from results of the original detections, and other two are from the cleaned trajectory. From the results, the curve indicated that after our proposed framework, outliers will be removed and then return a promising recovery.</td>
<td>18</td>
</tr>
<tr>
<td>4.4</td>
<td>2nd Example Video from UYDP Dataset. (a),(c) Original poses from pose estimator, (b) (d)Poses after proposed framework.</td>
<td>19</td>
</tr>
<tr>
<td>4.5</td>
<td>Estimation Results. The x axis represents the time sequence and the y axis represents the magnitude of the data. The red curve is the estimated data coming from SORDS algorithm; the cyan curve is the clean data coming from AR system and the blue curve is the noisy data with small noise and outliers. From this comparison, we see that the data can be recovered accurately.</td>
<td>21</td>
</tr>
<tr>
<td>4.6</td>
<td>Estimation of Errors. The red curve represents the error between estimation and clean data. The blue data represents the error between noisy and clean data. The result shows that our algorithm works significantly better for cleaning outliers and noises.</td>
<td>22</td>
</tr>
<tr>
<td>4.7</td>
<td>Accuracy v.s. Outliers. In this experiment, n = 100. The red curve represents our algorithm and the blue curve represents SRPCA. From the result, our algorithm has much higher accuracy than SRPCA as expected.</td>
<td>22</td>
</tr>
</tbody>
</table>
4.8 **Precision, Recall and Geometric Mean.** (a) Precision and Recall along with outliers; (b) Geometric Mean with outliers changing. In this experiments, by examining the precision, recall and geometric mean, we claimed that our algorithm has better performance than SRPCA.
List of Tables

4.1 Evaluation of accuracy for UYDP dataset. According to the notification from [2], there is a problem of annotating wrists from ground truth. Hence, wrists are not comparable for this dataset. By comparing with other joints, our proposed method does the best performance overall.
List of Acronyms

**JBLD**  JensenBregman Log-det Divergence. A distance-like function to represent the distance between two matrices which introduced in Chapter 2.3.

**IHSTLN**  Iterative Hankel Structure Total Least Square. An algorithm for matrix-completion.

**ALM**  Augmented Lagrangian Multipliers. An algorithm for matrix-completion which introduced in Chapter 3.1.3.

**SRPCA**  Structured Robust Principal Component Analysis. An algorithm to handle outliers.

**SORDS**  Supervised Outlier Remove for Dynamic Systems. An algorithm which is introduced in Chapter 3.2.
Acknowledgments

Here, I would like to express my deep thanks to all of the people who have helped me during the work for this thesis. Especially, I’m really appreciated for the support from my advisor, Dr. Octavia Camps. I have worked with her for three years since I was a senior student for undergraduate study. From knowing nothing about Computer Vision until knowing something right now; from couldn’t do anything until finishing my master thesis, she is always patient to help me with the research. In past three years, I spent very meaningful time with her in this lab which is Robust System Lab to do some research. I learned a lot other than I could learn from the class such as abilities to do the research. This kind of experience plays an important role in the process of getting a Ph.D. degree for the next goal of my study career. I always remembered the moment that she welcome me to join her lab three years ago. So, at this moment, I would like to deeply express my thanks to her for providing this valuable opportunity in the process of getting my degree.

Besides, please allow me to say ”thank you” to Dr. Mario Sznaier and Dr. Taskin Padir are on behalf during the process of getting my degree. For my research topic, Mario also provides a lot of help to make me understand problems that I was trying to solve. His knowledge and experiences were fully showed on analysis of problems, and his passions are always like motivations to encourage me to continue working on the problems when having challenges. Moreover, I got to know Dr. Padir from his interesting class of Assistive Robotics. From that class, I learned essential principles to build a simple robot which can help humans about solving problems for them.

I cannot forget members in RSL lab, they also helped me a lot for my research. They are Mengran Gou, Wenqian Liu, Xikang Zhang, Angels Rates-Borras, Bengisu Ozbay, Sadjad Asghari-Esfeden, Yin Dong, Tianyu Dai, Yongfang Chen, Yin Wang, Caglayan Dicle,and any others with whom I have interacted. In particular, I would like to acknowledge Xikang Zhang again for his great help in explaining related knowledge, coding, discussion, sharing ideas and experiences during the process of doing thesis work. Furthermore, for those friends who accompanied me through these years, I would like to say ”thanks so much”. They are Jiapei Guo, Xinyan Deng, Ke Chen, Linbin Chen, Ruijing Yi and people who encouraged me a lot. I spent a happy and expressive time with them.

Finally, I want to express my deepest love to my parents and thanks for their greasiest support to me for past six years. Six years ago, I became a ”Husky” in Northeastern. During these years, they are always on my side and encourage me a lot when I faced to challenges and difficulties, even if they are far from me. To get a better education means a lot to them so that they have done their best to provide me this opportunity to study here. I’m lucky and appreciated to have parents such like them. Thank you, my dad and mom!
Abstract of the Thesis

Outliers Cleaning in Dynamic Systems

by

Yuexi Zhang

Master of Science in Electrical and Computer Engineering

Northeastern University, April 2017

Dr. Octavia I. Camps, Advisor

In this thesis, we focused on solving the problem of detecting and cleaning outliers when estimating human poses in a temporal sequence. We addressed this challenge by formulating optimization problems that use properties of dynamic systems explaining the temporal data. In particular, we solved this problem using two different approaches. In the first approach, we assumed that the data was generated by an underlying, but unknown, dynamic system. So, we designed a framework which combined a distance-like function of the Hankel matrix constructed by measurements to detect sharp changes with a data Interpolation algorithm. In the second approach, we also incorporated apriori knowledge of the underlying dynamics. In this case, it was assumed that training data is available to learn the underlying dynamics. Then, trajectories were cleaned while respecting this apriori knowledge. We tested the proposed approaches using both synthetic and real data from video sequences. Experiments showed that our approaches had better performances.
Chapter 1

Introduction

1.1 Overviews

Human pose estimation is a popular topic in computer vision, which applications such as human activity recognition. In recent years, as deep is learning booming rapidly, the estimation becomes more accurate. However, misdetections and unexpected outliers can still occur especially in some complicated cases such as occlusions are present. Some failure cases are shown in Figure 1.1.

Our approaches focused on the sequences which allow us to do our analysis by using motion information from a dynamic system. In [3], it was showed that dynamics can be used as a feature to interpret activities and that Hankel Matrices constructed for trajectories capture the dynamic information. As proposed by [4], dynamic constrains can improve robustness to occlusions, video rectification and even to the dynamic changes without any requirement of camera calibration. Moreover, in the case of occlusion, missing measurements can be estimated by the fact that they must lie in the subspace spanned by previous measurements and also must satisfy epipolar constrains. In order to compare dynamic changes, [5] showed that dynamics can be efficiently compared using distance-like functions such as the Affine Invariant Riemannian Metric (AIM) [6] [7] and the Jensen-Bregman Log-det Divergence (JBLD) [8].

1.2 Related Work

Classical approaches to human pose estimation are based on the pictorial structure model as proposed in [9]. The model focuses on representing parts of the body as a tree-structured graphical model, together with geometric constrains which capture the spatial correlations on pairs of parts.
CHAPTER 1. INTRODUCTION

Figure 1.1: **Failure Cases.** Some of misdetections and outliers may occur when there are occlusions and overlapping parts.

Inspired by these pictorial structure models, other works \cite{10,11,12,13} have been done to improve the performance and accuracy. However, this model requires that all parts are visible so that the model can capture correlations among them. Therefore, this approach can fail due to occlusions or self-occlusions. To solve this problem, Ramakrishna *et al.* \cite{14} proposed an occlusion-aware algorithm by tracking the pose in an image sequence to use the spatial and temporal consistency. Also, \cite{15} proposed a graphical model for both self-occlusion and occlusion by other objects. They claimed that the model structure can learn occlusion coherence from data by capturing the interactions between human body parts and objects. In recent years, people tend to solve the problem by using deep learning networks. In \cite{1}, they implicitly modeled long-range dependencies between variables for a problem of articulated pose estimation. They achieved this by designing a sequential architecture together with *convolutional networks* to learn both image features and image-dependent spatial models. As proposed in \cite{16}, they presented a graphical model with novel pairwise relations by using Deep Convolutional Neural Networks (DCNNs) to learn conditional probabilities for the presence of parts and the spatial relationships between image measurements. \cite{17} also combined the DCNN with the expressive mixture of parts model to propose an end-to-end framework. As they claimed in the paper, their framework can reduce the parameter space when modeling the relationship between the spatial and appearance information over all parts. Another framework \cite{18}, proposed a feature learning structure for the correlations among body joints and used the ImageNet pre-trained VGG-16 as their base model.

Researchers have tried to improve the estimation for human poses using video sequences. \cite{19} proposed a framework of tracking-by-selection to solve estimation as a multi-target tracking problem. They claimed that each joint has a path in the sequence and that the location of the joint does not depend on its temporal neighbors. The basic idea of estimating poses over videos is trying to capture both temporal and appearance information across frames based on a promising
CHAPTER 1. INTRODUCTION

detection of each single image. In [20], they firstly used temporal information in deep Convolutional Networks (ConvNet) with optical flow as its input motion feature. Inspired by this work, [21] proposed a new approach for part localization in ConvNets. They also used optical flow as its input feature and pointed that an additional convolutional layer was able to learn a simplicity model of the spatial human layout. Later on, [2] improved this work to demonstrate that a few high-accuracy poses are given from ConvNets estimator at first few frames, by temporal information and optical flow, joint estimations can be propagated for the rest of frames in a video sequence.

1.3 Thesis Organization

This thesis is orginazed as following:

Chapter 2: Explains background of dynamic systems, Hankel Matrix, Gram Matrices, as well as distance-like functions.

Chapter 3: Explains details of the proposed approaches.

Chapter 4: Discussions of applications.

Chapter 5: Conclusion and Future Work.
Chapter 2

Preliminaries

2.1 Hankel Matrix Representations

Consider a sequence of observations of an Auto Regression (AR) model from [22],

\[ y_k = \sum_{i=1}^{n} a_i y_{k-i} \]  
\[ (2.1) \]

where \( a = [a_1, a_2...a_n] \) is a set of coefficients. The sequence is associated to a block Hankel Matrix with the form

\[
H_{y}^{s,r} = \begin{bmatrix}
    y_1 & y_2 & y_3 & \cdots & y_r \\
    y_2 & y_3 & y_4 & \cdots & y_{r+1} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    y_s & y_{s+1} & y_{s+2} & \cdots & y_{r+s+1}
\end{bmatrix}
\]  
\[ (2.2) \]

where \( r \) and \( s \) determine the dimension of the Hankel Matrix. As noted in [8], the columns of the Hankel Matrix are related to overlapping subsequences of the data, shifted by one and the block is anti-diagonal. Because of this special structure, it captures very usefully dynamic information of the system. Specifically, for selected \( r \) and \( s \), \( \text{rank}(H_{y}^{s,r}) = n < \min(r, rs) \). Furthermore, since the data comes from observations of an AR model when setting \( r = n \) in (2.2), it was showed that the last column of the Hankel Matrix is a linear combination of the previous ones and coefficients for this combination are the coefficients of the auto regressor. Namely,

\[
H_{y}^{s,r}[a^T - 1]^T = 0
\]  
\[ (2.3) \]
2.2 The Gram Matrices

When comparing two Hankel Matrices, we can simply compute the angle between the corresponding subspaces. However, in the real applications, observations will contain the noise so that the rank of matrix $H_y$ will be the full rank [5] and the angle between subspaces will become zero. One solution can be first to estimate its rank $\rho$ for the $H_y$ clean matrix. However, it is difficult to reliably estimate $\rho$ principal components. Alternatively, one can use Gram matrices as introduced in [5]:

$$\hat{G} = \frac{HH^T}{\|HH^T\|_F} \quad (2.4)$$

As pointed in [5], the Gram matrices capture the same properties of rank and invariance as the Hankel Matrices. However, Gram matrices are limited to Positive Semi-Definite (SPD) manifold, allowing us to use distance on this manifold.

2.3 The Jensen-Bregman Log-det Divergence (JBLD)

The difference between two dynamic systems can be captured by computing the distance between their associated Gram matrices. Here, we will introduce a distance-like function on the PD manifold named Jensen-Bregman Log-det Divergence (JBLD) [8] defined as:

$$\delta_{ld}^2(X, Y) = \log \left| \frac{X + Y}{2} \right| - \frac{1}{2} \log |XY| \quad (2.5)$$

Since JBLD itself is not a metric, one solution proposed by [23] is to compute $\delta_{ld}^2$ instead of computing $\delta_{ld}$. 

5
Chapter 3

Methods

3.1 Proposed Framework for Cleaning Outliers

The main goal of this thesis is to provide a simple and easy way of dealing with outliers for human pose estimation. Since, we focus on cleaning outliers in a trajectory, i.e., a time series, we can use temporal information from the past and the future to recover the present information through the underlying dynamic system. In real problems, poses which are coming from detectors always contain several joints. Therefore, it’s practical to deal with joint-wise outliers. Thus, in this thesis, we deal with one joint sequence at a time, as summarized in Figure 3.1.

3.1.1 Overview of Computing JBLD Values

First, we define a window size to chunk the entire trajectory into small tracklets and shift the window from the beginning until the end by $\Delta t$ frames at each time. Then, we fit each small tracklet of the window into a Hankel Matrix Chapter 2.1 to capture its dynamics. As discussions in Chapter 2.2, then we compute the Gram matrix for each Hankel matrix. Afterward, we compute the JBLD distance between the previous and current Gram matrices to compare the dynamic information of two windows. Since Gram matrices should be PD, we defined a small residue to regularize it.

For a human activity in a time sequence, poses should be coherent and consistent across frames. So, a joint trajectory which over time can be described by a low dynamic system. At this point, we claimed that if there are no outliers, the JBLD distance between two adjacent windows should not be too large. Therefore, we can determine where outliers occurred by examining the JBLD values between consecutive windows.
CHAPTER 3. METHODS

Figure 3.1: The flowchart shows our framework in details and present the entire process that how to get a clean trajectory.

3.1.2 Steps of Computing JBLD

Suppose we have a series of human poses from a video sequence with T frames and n joints. We define a shifting window of size L and shifted $\Delta t$ frames at each time. Let a trajectory of one joint in the sequence, $P_{1}^{i}, P_{2}^{i}, ..., P_{T}^{i}$, where $P_{t}^{i} = (x_{t}, y_{t})$, $i \in \{1, 2, ..., n\}$. For example, $P_{t}^{i}$ represents the $i^{th}$ joint at $t^{th}$ frame.

The trajectory can be chunked into several tracklets with the defined window size. There is a $\Delta t$ shift between two adjacent windows, namely, there are $L - \Delta t$ frames overlapping.

Window 1 : $P_{1}^{i}, P_{2}^{i}, ..., P_{L}^{i}$;

Window 2 : $P_{1+\Delta t}^{i}, P_{2+\Delta t}^{i}, ..., P_{L+\Delta t}^{i}$;

Window 3 : $P_{3+\Delta t}^{i}, P_{4+\Delta t}^{i}, ..., P_{L+2\Delta t}^{i}$;

...;

Then, computing the Gram matrix for each Hankel matrix of each window:

$$\hat{G}_{1} = \frac{H_{1}H_{1}^{T}}{\|H_{1}H_{1}^{T}\|_{F}} + \text{residue}$$

$$\hat{G}_{2} = \frac{H_{2}H_{2}^{T}}{\|H_{2}H_{2}^{T}\|_{F}} + \text{residue}$$

$$\hat{G}_{3} = \frac{H_{3}H_{3}^{T}}{\|H_{3}H_{3}^{T}\|_{F}} + \text{residue}$$
where \textit{residue} is a small number to force the Gram matrix to be PD. After this, we compute the JBLD distances between two adjacent windows.

\[
1^{st} \text{JBLD} = JBLD \left\{ \hat{G}_1, \hat{G}_2 \right\} \\
2^{nd} \text{JBLD} = JBLD \left\{ \hat{G}_2, \hat{G}_3 \right\} \\
\vdots
\]

The relationship between the frame where outliers occurred and the number where JBLD changes dramatically can be carried as below:

\[
Frame(t) = JBLD(j) \times \Delta t + L
\]

where \( j \in \{1, 2, \ldots, J\} \), \( t \in \{1 \times \Delta t + L, 2 \times \Delta t + L, \ldots, j \times \Delta t + L\} \)

### 3.1.3 Data Interpolation and Matrix Completion

Once outliers are removed, we can recover the missing data using Augmented Lagrangian Multipliers (ALM) method to interpolate the sequence such that the output sequence should be of low order dynamics while keeping fidelity to inliers.

We firstly form the problem as follows. Given a vector sequence \( y_t \) where \( t = 1, 2, \ldots, n \), we want to solve

\[
\begin{align*}
\min_x & \quad \|H_x\|_* + \frac{\lambda}{2} \|x_\Omega - y_\Omega\|^2_2 \\
\text{subject to} & \quad Z = \text{mat}(Sx)
\end{align*}
\]  

(3.2)

where \( H_x \) is block Hankel matrix of vector sequence \( \{x_t\} \) and \( \Omega \subset \{1, 2, \ldots, d \times n\} \) is the set of indices of inliers. \( x_\Omega \) (or \( y_\Omega \)) is a vector which contains concatenation of all \( x_i \) (or \( y_i \)) with \( i \in \Omega \). If we define \( P_\Omega = I_\Omega : \) where \( I \) is the identity matrix, then it is obvious that \( x_\Omega = P_\Omega x \) and \( y_\Omega = P_\Omega y \). Also we introduce the Hankel structure operator \( S \) as in [24]. The problem becomes

\[
\begin{align*}
\min_x & \quad \|\text{mat}(Sx)\|_* + \frac{\lambda}{2} \|P_\Omega x - P_\Omega y\|^2_2 \\
\text{subject to} & \quad Z = \text{mat}(Sx)
\end{align*}
\]

(3.3)

To make the problem easier to solve, we introduce a new variable \( Z \) and add an equality constraint. And the problem becomes

\[
\begin{align*}
\min_{x,Z} & \quad \|Z\|_* + \frac{\lambda}{2} \|P_\Omega x - P_\Omega y\|^2_2 \\
\text{subject to} & \quad Z = \text{mat}(Sx)
\end{align*}
\]

(3.4)

(3.5)

Now we are ready to use the ALM method to solve the problem. We first write the augmented Lagrangian function
CHAPTER 3. METHODS

\[ L = \|Z\|_1 + \frac{\lambda}{2} \|P_\Omega x - P_\Omega y\|_2^2 - \langle Y, Z - \text{mat}(Sx) \rangle + \frac{\mu}{2} \|Z - \text{mat}(Sx)\|_F^2 \] (3.6)

The function below has the same minimizer as that above

\[ \tilde{L} = \|Z\|_1 + \frac{\lambda}{2} \|P_\Omega x - P_\Omega y\|_2^2 + \frac{\mu}{2} \|Z - \text{mat}(Sx) - Y/\mu\|_F^2 \] (3.7)

We can minimize this objective function with respect to \(x\) and \(Z\) separately. Using proximity operator of \(Z\), there is

\[ Z^{(k+1)} = D_{1/\mu}(\text{mat}(Sx^{(k)}) + Y^{(k)}/\mu) \] (3.8)

where the soft thresholding operator \(D\) is defined as in [25]

\[ D_\tau(X) = UD_\tau(\Sigma)V, \quad D_\tau(\Sigma) = \text{diag}(\{\sigma_i - \tau\}_+) \] (3.9)

where \(y_+ = \max(0, y)\). Then, we solve for \(x\) in

\[ \nabla_x \tilde{L} = 0 \] (3.10)

we get the closed form solution

\[ x^{(k+1)} = (\mu S^T S + \lambda P_\Omega^T P_\Omega)^{-1} \left[ \lambda P_\Omega^T P_\Omega y - \mu S^T \text{vec} \left( Y^{(k)}/\mu - Z^{(k+1)} \right) \right] \] (3.11)

To speed up our method, we used inexact ALM instead of exact ALM, which means we update the Lagrangian multiplier immediately after each update of \(Z\) and \(x\). The Lagrangian multiplier is updated as follows

\[ Y^{(k+1)} = Y^{(k)} - \mu (Z - \text{mat}(Sx)) \] (3.12)

Then, we update \(\mu^{(k)}\)

\[ \mu^{(k+1)} = \rho \mu^{(k)}, \quad \rho > 1 \] (3.13)

Then, we start the next iteration. The whole algorithm is summarized in Algorithm 1.
CHAPTER 3. METHODS

Algorithm 1 Interpolation with inexact ALM algorithm
1: **Input:** Data $y$, set of inlier indices $\Omega$, convergence tolerance $\epsilon$, initial parameter $\mu_0$, penalty parameter $\lambda$, ALM parameter change rate $\rho$
2: **Initialize:** $Z^{(0)} \leftarrow 0$, $x^{(0)} \leftarrow 0$, $Y^{(0)} \leftarrow 0$, $\mu^{(0)} \leftarrow \mu_0$, $k \leftarrow 0$
3: Form Hankel structure operator $S$
4: $P_\Omega \leftarrow I_{\Omega}$.
5: While $\|Z - \text{mat}(Sx)\|_F > \epsilon$
   a. $Z^{(k+1)} \leftarrow D_{1/\mu} \text{mat}(Sx^{(k)}) + Y^{(k)}/\mu$
   b. $x^{(k+1)} \leftarrow (\mu S^T S + \lambda P_T P_\Omega)^{-1}$
      \[ [\lambda P_T P_\Omega Y - \mu S^T \text{vec}(Y^{(k)}/\mu - Z^{(k+1)})] \]
   c. $Y^{(k+1)} \leftarrow Y^{(k)} - \mu (Z - \text{mat}(Sx))$
   d. $\mu^{(k+1)} \leftarrow \rho \mu^{(k)}$
   e. $k \leftarrow k + 1$
   end while
6: $x^* \leftarrow x^{(k)}$
7: **Output:** $x^* = 0$

3.1.4 Outline of the Proposed Framework

The idea behind the proposed framework is to simply dealing with outliers for each joint at a time. The outline is shown in below.

Algorithm 2 Proposed Framework of Cleaning Outliers
1: **Data:** Detections from pose estimator with length of $T$ frames for each joint
2: **Results:** A set of clean trajectories for each joint over the entire sequence.
3: **Parameters:** $np =$ number of joints, $\Delta t =$ shift frames, $L =$ window size, $\lambda =$ penalty for optimization process.
4: for $i = 1 : np$
   a. $JbdValues = JBLD(Data \{1, i\})$
   b. $Jbd_{ith} = \text{find}(JbdValues > \text{threshold})$;
   c. $OutlierFrames = Jbd_{ith} \times \Delta t + L$
   d. $Data \{1, i\} = ALM(Data \{1, i\}, \lambda)$;
   end
5: A set of clean trajectories are returned
CHAPTER 3. METHODS

Since our algorithm can handle every joint trajectory at the same time, the output from the process is a set of clean trajectories over the entire video sequence. After cleaning, we can get more accurate pose estimations across the video sequence. Some examples will be showed in the next chapter.

3.2 Supervised Outlier Removal for Dynamic Systems (SORDS)

In this section, we will discuss how to do outlier cleaning when training data is available.

3.2.1 Optimization Problem

Suppose that observations are from an AR model in Chapter 2.1 so that we denote the clean data as $y_k$.

\[
y_{k+1} = \sum_{i=1}^{n} a_i y_{k-i}
\]  

(3.14)

If observations contain noises, we expressed the observation as:

\[
\tilde{y}_k = y_k + e_k
\]  

(3.15)

In most cases, $e$ is the sum of two independent terms. One is outlier noise and the other one is inlier noise. We defined $e = \eta + \varepsilon$ and the expression of the observation is given by:

\[
\tilde{y}_k = y_k + \eta_k + \varepsilon_k
\]  

(3.16)

where $\eta_k$ is denoted as outlier noise and $\varepsilon$ is small noise assumed to be Gaussian noise. In the data, we assume that outliers are fewer than inliers. Given the observations $\tilde{y}$, the problem becomes how to have a best estimate which we denote as $\hat{y}$ to recover the clean data from the noisy observations.

Firstly, if only dealing with outlier noise in the absence of $\varepsilon$, the problem can be written as:

\[
\min \| \eta \|_0 \text{ s.t. } \tilde{y} = \hat{y} + \eta
\]  

(3.17)

where $\| \eta \|_0$ counts the number of nonzero elements in $\eta$. Since $\eta$ is sparse, so we look a sparse solution. Since the $l_0$ norm is non-convex, we considere its efficient and convex relaxation $l_1$ norm, i.e.,

\[
\min \| \eta \|_1 \text{ s.t. } \tilde{y} = \hat{y} + \eta
\]  

(3.18)
CHAPTER 3. METHODS

This is can be solved by using convex program tools\cite{26} and it has a sparse solution \cite{27}. Afterwards, in real cases, we considered the combination of both outliers and small noises. Recalling the expression of Hankel Matrix in Chapter \ref{chapter:2} and that the observations are from and AR model, we have an expression such that

\[ H_{\hat{y}}[a^T - 1]^T = 0 \]  

(3.19)

where \( H_{\hat{y}} \) is the Hankel Matrix of the clean data \( \hat{y} \) and \( a \) is the regressor, we have another constrain for our optimization problem:

\[
\min \| \eta \|_1 + \lambda \| \varepsilon \|_2^2
\]

(3.20)

s.t \( \tilde{y} - \hat{y} - \eta - \varepsilon = 0 \), \( H_{\hat{y}}[a^T - 1]^T = 0 \)

where \( \lambda \) is a parameter to penalize the term of \( l_2 \) norm. We solved this problem by using CVX tool \cite{26}.

3.2.2 SORDS Algorithm

In Section \[3.2.1\] we discussed how we formed our idea into an optimization problem. We named our algorithm as Supervised, because we assumed that the regressors was already known from training data. Therefore, we expressed our algorithm as below:
CHAPTER 3. METHODS

Algorithm 3 Supervised Outlier Remove for Dynamic Systems

1: Testing Data: Detections from pose estimator with length of T frames for each joint from each cluster;
2: Regressors a: Trained from training data for each cluster;
3: Results: A set of clean trajectory for each joint over the entire sequence;
4: Parameters: \( \lambda \): to penalize the term of noise;
5: \( \hat{y} \): estimated data, \( H_{\hat{y}} \): Hankel Matrix for \( \hat{y} \);
6: \texttt{cvx_begin}
   \begin{align*}
   \text{variables} & \hat{y} \\
   \text{obj} & = \| \eta \|_1 + \lambda \| \varepsilon \|_2^2 \\
   H_{\hat{y}} [a^T - 1]^T & = 0 \\
   \hat{y} - \hat{y} - \eta - \varepsilon & = 0 \\
   \text{minimize} & (\text{obj}) \\
   \text{cvx_end}
   \end{align*}
7: A clean trajectory is returned

Our goal is to apply our algorithm on human pose estimation. In the next chapter, we showed experiments to test our SORDS algorithm.
Chapter 4

Applications

4.1 Video Experiments

In Section 3.1 which we presented a framework to deal with outliers. In this section, we will do evaluations on the Upper-body YouTube Dancing Pose (UYDP) dataset to evaluate the performance of our framework. As proposed in [28], an algorithm named Iterative Hankel Structure Total Least Square (IHSTLN) can also deal with missing data in a trajectory. Therefore, we considered it is an alternative data interpolation algorithm which can be applied to our framework. In the following experiments, we evaluated the performance by using both ALM and IHSTLN. Furthermore, we also compared it with the Structured Robust Principal Component Analysis (SRPCA) method proposed in [24] which also work for cleaning outliers.

4.1.1 Dataset

Upper-body YouTube Dancing Pose (UYDP). This dataset [29] contains 20 videos and there are 100 frames for each video with annotated ground truth. In each video, there is only one performer with dancing poses.

4.1.2 Evaluations

To be consistent with previous work, we evaluated the performance of our results on UYDP dataset by using Average Precision of Key Points (APK) [30] at a threshold 0.2.
CHAPTER 4. APPLICATIONS

4.1.3 Comparison for UYDP Dataset

Since our proposed method is based on the detections from [1], we not only compared with other methods but also compared with [1]. The model of detection from [1] we used in this thesis is trained from the combination of MPII and LSP dataset with 6 stages, which is denoted as MPII+LSP 6-stage CPM. Since the human pose is symmetric, we took the average of accuracy for those symmetric parts. For the entire UYDP dataset, results are shown in Table 4.1. The most challenging part for this dataset comes from estimating locations for elbows and wrists since videos here are all about dancing. Unlike smooth activities, poses for dancing are hard to estimate because bodies or arms will twist in complicated ways and sometimes activities will perform along with occlusions. However, by looking at results in Table 4.1, we see that our framework does perform better than any other algorithm. In the following, we presented some examples from the UYDP dataset to show the visual performance of our framework. According to the Table 4.1, visualizations showed were came from "JBLD+ALM" framework.

<table>
<thead>
<tr>
<th></th>
<th>Head</th>
<th>Wrist</th>
<th>Elbows</th>
<th>Shoulders</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pfister et al. [21]</td>
<td>78.7</td>
<td>-</td>
<td>35.2</td>
<td>63.3</td>
<td>-</td>
</tr>
<tr>
<td>Chen &amp; Yuille [16]</td>
<td>86.3</td>
<td>-</td>
<td>46.8</td>
<td>80.3</td>
<td>-</td>
</tr>
<tr>
<td>Yang &amp; Ramanan [30]</td>
<td>81.7</td>
<td>-</td>
<td>17.6</td>
<td>66.5</td>
<td>-</td>
</tr>
<tr>
<td>Shen et al. [29]</td>
<td>90.9</td>
<td>-</td>
<td>33.3</td>
<td>83.5</td>
<td>-</td>
</tr>
<tr>
<td>Charles et al. [2]</td>
<td>91.7</td>
<td>-</td>
<td>57.6</td>
<td>83.5</td>
<td>-</td>
</tr>
<tr>
<td>Wei et al. [1]</td>
<td>96.0</td>
<td>-</td>
<td>82.1</td>
<td>87.7</td>
<td>-</td>
</tr>
<tr>
<td>SRPCA [24]</td>
<td>96.4</td>
<td>-</td>
<td>82.8</td>
<td>88.1</td>
<td>-</td>
</tr>
<tr>
<td>JBLD+IHSTLN</td>
<td>96.6</td>
<td>-</td>
<td>82.73</td>
<td>88.76</td>
<td>-</td>
</tr>
<tr>
<td>JBLD+ALM</td>
<td>97.0</td>
<td>-</td>
<td>83.5</td>
<td>88.6</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.1: Evaluation of accuracy for UYDP dataset. According to the notification from [2], there is a problem of annotating wrists from ground truth. Hence, wrists are not comparable for this dataset. By comparing with other joints, our proposed method does the best performance overall.

Example 1: In this example, according to the pose estimator [1], the joints which contain outliers are the right elbow and the right wrist. Then, misdetections occurred around the 25th frame to the 32nd frame according to JBLD values over the entire sequence in Figure 4.2. Afterward,
CHAPTER 4. APPLICATIONS

(a) Original poses from pose estimator

(b) Poses after proposed framework

Figure 4.1: **1st Example Video from UYDP Dataset.** (a) Original poses from pose estimator, (b) Poses after proposed framework.
CHAPTER 4. APPLICATIONS

(a) JBLD distribution for the Right Elbow  (b) JBLD distribution for the Right Wrist

Figure 4.2: **JBLD values of the joint with outliers in the 1st example.** We denoted two joints as the right elbow and the right wrist based on the pose estimator [1]. The yellow curve is from results of the original detections, and other two are from the cleaned trajectory. From results, the curve indicated that after our proposed framework, outliers will be removed and then return a promising recovery.

we compared with the input detections from pose estimator [1], results showed that outliers were removed and significantly reduced values of JBLD. From Figure 4.1 results were presented visually before and after applying our proposed framework. By looking at visualizations, we concluded that estimations for poses were improved.

**Example 2:** In this example, we presented two clips cropped from the video at the location where outliers occurred according to the JBLD distribution showed in Figure 4.3. From Figure 4.3 outliers presented around the 30th frame first and then the 79th frame. We also compared JBLD curves from the proposed framework with the original detections from the pose estimator [1]. Afterward, we showed results of the cleaned trajectory along with the original detection as shown in Figure 4.4. Visualizations showed that outliers were efficiently removed and then a more clean trajectory was returned.

4.2 Experiments for SORDS Algorithm

In this section, we will do evaluations for our SORDS algorithm on Synthetic data. To examine the performance of our algorithms, we also compare with SRPCA algorithm.
CHAPTER 4. APPLICATIONS

Figure 4.3: **JBLD values of the joint with outliers in the 2nd example.** We denoted this joint as left wrist based on the pose estimator. The yellow curve is from results of the original detections, and other two are from the cleaned trajectory. From the results, the curve indicated that after our proposed framework, outliers will be removed and then return a promising recovery.

4.2.1 Experiments with Synthetic Data

At the beginning, we tested our idea starting with synthetic data. In experiments, we randomly generated a set of observations from AR model. Since the algorithm is supervised, to get regressors of the model, we sampled a set of poles including both real and complex numbers from a ring which is near the unit circle. To test our algorithm, we chose an order 2 system with 100 data points in this experiment. Afterward, we generated two random initial conditions and then using regressors to get the rest of the data. In the next step, we added some Gaussian noise with the noise level of 0.1 and manually set the different number of outliers with the value of 3 to get noisy data. Finally, we used Algorithm showed in Section 3.2.2 to solve the problem.

**Experiment1:** For the estimation results, we showed an example in Figure 4.5. In this example, we fixed systems and number of outliers. The results showed that the data was accurately recovered from the noisy data. In order to further present the results of the estimation, we also plotted the error curve showing in Figure 4.6. From the curve, the error between estimated and clean data is very close to zeros, hence, our algorithm works well for cleaning outliers and noise.

**Experiment2:** We evaluated the performance of our algorithm with increasing number of
CHAPTER 4. APPLICATIONS

(a) Clip1: Original poses from pose estimator
(b) Clip1: Poses after proposed framework
(c) Clip2: Original poses from pose estimator
(d) Clip2: Poses after proposed framework

Figure 4.4: 2nd Example Video from UYDP Dataset. (a),(c) Original poses from pose estimator, (b) (d) Poses after proposed framework.
outliers. In this test, we fixed regressors of the system but changed initial conditions \( n \) times. Then, within each iteration, we increased the number of outliers. Afterward, we defined a threshold to determine the accuracy. If the Euclidean distance between the estimation and clean data point is within the threshold, we set it as the corrected estimation. Otherwise, its the uncorrected one. For the accuracy of estimations, we computed the ratio which is between corrected estimations and the total number of estimations. Then, for each number of outliers, we took the average of results over \( n \) different initial conditions. The purpose of doing this is to eliminate the effect caused by initial conditions and make the result work for more general cases. In the next step, we compared our algorithm with SRPCA. Because SRPCA is the unsupervised, so we expected that our algorithm has better. The result is shown in Figure 4.7. In this test, we set \( n = 100 \) and evaluated the accuracy with outliers from 5 to 90%. The plots in the figure confirm that our algorithm has higher accuracy than SRPCA as expected.

**Experiment3:** In this experiment, we examined performances of Precision, Recall and their geometric mean for our algorithm and SRPCA, with the different number of outliers. Then, we assumed that if outliers were indeed cleaned out, the Euclidean distance between estimated data and the noisy data should be large at the location where originally contains the outlier in the noisy data. Hence, we defined another threshold to determine sets for Truth Positive (TP), False Positive (FP) and False Negative (FN) by comparing the Euclidean distance. Since we also followed the same outline as the experiment 2, so that we had results from 5 to 90% outliers over \( n = 100 \) different initial conditions. Then, we took the average over 100 iterations to get results. Results are shown in Figure 4.8. From the results, we can see that our algorithm has a significantly better performance than SRPCA.
CHAPTER 4. APPLICATIONS

Figure 4.5: Estimation Results. The x axis represents the time sequence and the y axis represents the magnitude of the data. The red curve is the estimated data coming from SORDS algorithm; the cyan curve is the clean data coming from AR system and the blue curve is the noisy data with small noise and outliers. From this comparison, we see that the data can be recovered accurately.
Figure 4.6: **Estimation of Errors.** The red curve represents the error between estimation and clean data. The blue data represents the error between noisy and clean data. The result shows that our algorithm works significantly better for cleaning outliers and noises.

Figure 4.7: **Accuracy v.s. Outliers.** In this experiment, n = 100. The red curve represents our algorithm and the blue curve represents SRPCA. From the result, our algorithm has much higher accuracy than SRPCA as expected.
Figure 4.8: **Precision, Recall and Geometric Mean.** (a) Precision and Recall along with outliers; (b) Geometric Mean with outliers changing. In this experiments, by examining the precision, recall and geometric mean, we claimed that our algorithm has better performance than SRPCA.
Chapter 5

Conclusions and Future Work

In this thesis, we introduced two approaches of outlier removing for human pose estimations. The first proposed approach is a simple but efficient framework that uses JBLD distance to detect outliers and then recovers missing data when by doing data interpolation and matrix completion [28]. JBLD is a distance to compare PD matrices. If there are no outliers, two adjacent matrices should be similar so that JBLD value is small at this point. Otherwise, we claimed that an outlier existed. To do the data interpolation, we form a big Hankel Matrix over an entire trajectory and then fill in gaps instead of outliers. Then, using temporal information to predict gaps as well as minimizing the rank of this Hankel Matrix until it gets to the minimum. A clean trajectory can be obtained after our framework. We tested our framework on several real examples in Chapter 4 and it beats the state-of-art as results shown in Table 4.1. We extended our problem to focus on solving an optimization problem which can remove outliers in one step if training data is available.

Future Work. The first approach is heavily based on the initial detection. One weakness of this approach is that if there are too many outliers, the alarm coming from the first step of this framework might not be accurate. In this case, we need to always make sure the outlier level of the detector. Also, if the sequences we are trying to work on are complex, we have to ensure that detections are promising enough to use. Therefore, one way of improving the performance of our framework is to look for a more accurate pose estimator. The other way is to manipulate detections after the estimator to make them more promising. For the second improvement, we have done some preliminary work that we treated pose estimation in a time series as a multi-targets tracking problem. Then, we can combine motion with appearance information to get more accurate trajectory, similarly in [31, 19]. For the second approach, since our work now is at the beginning, the next step is to apply this algorithm to real data.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


[27] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution,” *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.


BIBLIOGRAPHY
