INESC-ID TECHNICAL REPORT N^o /2019
FINAL GRANT REPORT

Distributed Estimation of Multiple Vector Fields from Target Trajectories

Author:
Ana PORTÊLO

Supervisors:
João MIRANDA LEMOS

This work has been performed in the framework of project:
SPARSIS – Sparse Modeling and Estimation of Motion Fields,
supported by FCT under contract: PTDC/EEIPRO/0426/2014.

April 30, 2019

Summary

This report summarises the work done during the 16 months research grant that started on January 2018. The work plan can be divided into milestones identified in this report as sections. The main objectives of the work consist in the development of algorithms and software for sparse and distributed estimation and its application to target tracking in a computer vision framework.

The first part of the work proposes three distributed estimation approaches to the problem of vector field estimation from pedestrian trajectory data. In many surveillance applications the area of interest is very large or includes alleys or corners. Thus, images from a single camera may not cover the entire area of interest, and the images of two or more cameras need to be combined. This setting motivates the use of distributed optimization algorithms. The proposed approaches are (1) a purely decentralized approach without communication between agents, (2) a distributed estimation approach based on a cooperative game, and (3) a distributed version of the Alternating Direction Method of Multipliers (ADMM). The performance of these approaches in the estimation of different types of vector fields is important to select the best approach for each scenario. Examples of vector field estimation using synthetic and real data showed that (a) the Decentralized approach was the most robust and cost efficient approach to estimate generating vector fields that do not have changes in direction, (b) the Cooperative game approach was very susceptible to changes in the direction of the generating vector field, and (c) the distributed ADMM approach was the most robust and reliable approach to estimate generating vector fields that change direction. The proposed approaches need to be adapted to simultaneously estimate multiple vector fields. Within the scope of the first part of the work, the grantee's participation on the Summer School on Control and Robotics: Fundamentals for advanced research (2018) was very important.

The second part of the work focuses on the problem of multiple vector field estimation. Far-field activities which can represented as time series or trajectories and be summarized in compact representations of frequent patterns. Popular representations such as clustering or probabilistic modelling of trajectories often do not inform about both velocity and direction of motion, which are by definition visually and quantitatively embedded in vector fields. However, a common use of vector fields may dismiss information about forbidden areas, or regions with concurrent activity patterns. To address this problem we present a non-iterative layered vector field estimation process that yields sparse vector field abstractions of activity patterns from groups of trajectories. The key feature of our approach is the estimate of the probability density function (PDF) of targets positions: it automatically tunes the cost function parameter, and serves as weights in the sparse estimation problem. Experiments in synthetic trajectory data show that the proposed estimation approach yields correctly sparse vector fields which are similar to known generating vector fields in both velocity and direction. We also propose a trajectory labeling algorithm that labels trajectories according to their activity patterns based on the vector field abstractions. Experiments on three real data sets yield higher labeling accuracy of test trajectories when compared to other generative models. The proposed approach abstracts frequently observed activity patterns in a scene and informs about

their physical and semantic meaning. Outlier trajectories are also detected. Within the scope of the second part of the work, the grantee's visit to the center for Intelligent Sensing at Queen Mary University of London, under the supervision of Professor Andrea Cavallaro, was very important to kickstart the approach to the multiple vector field estimation problem.

Contents

1. Distributed estimation of a Single Vector Field **5**
1.1. Introduction . . . 5
1.2. Dynamic Model and Vector Field Estimation . . . 6
1.2.1. Dynamic Model of Pedestrian Trajectories . . . 6
1.2.2. Vector Field Representation . . . 6
1.2.3. Vector Field Estimation . . . 7
1.3. Distributed Estimation Algorithms . . . 7
1.3.1. Notation . . . 7
1.3.2. Purely Decentralized Vector Field Estimation . . . 8
1.3.3. Distributed Vector Field Estimation Based on a Cooperative Game 9
1.3.4. Distributed Vector Field Estimation Based on ADMM . . . 11
1.4. Experimental Results . . . 14
1.4.1. Synthetic Pedestrian Trajectory Data . . . 15
1.4.2. Real Pedestrian Trajectory Data . . . 40
1.5. Conclusions . . . 44

2. Centralized Estimation of Multiple Vector Fields **45**
2.1. Introduction . . . 45
2.2. Estimation of Multiple Vector Fields . . . 47
2.3. Activity Pattern Labeling . . . 48
2.4. Experimental Results . . . 49
2.4.1. Synthetic data . . . 49
2.4.2. Real data . . . 51
2.5. Conclusion . . . 52

A. Distributed ADMM **57**
A.1. Distributed ADMM . . . 57
A.2. Toy Examples . . . 58
A.2.1. Example 1 . . . 58
A.2.2. Example 2 . . . 61
A.2.3. Example 3 . . . 63
A.2.4. Example 4 - from [1] . . . 65

1. Distributed estimation of a Single Vector Field

This chapter proposes three distributed estimation approaches to the problem of vector field estimation from pedestrian trajectory data. In many surveillance applications the area of interest is very large or includes alleys or corners. Thus, images from a single camera may not cover the entire area of interest, and the images of two or more cameras need to be combined. This setting motivates the use of distributed optimization algorithms. The proposed approaches are (1) a purely decentralized approach without communication between agents, (2) a distributed estimation approach based on a cooperative game, and (3) a distributed version of the Alternating Direction Method of Multipliers (ADMM). The performance of these approaches in the estimation of different types of vector fields is important to select the best approach for each scenario. Examples of vector field estimation using synthetic and real data showed that (a) the Decentralized approach was the most robust and cost efficient approach to estimate generating vector fields that do not have changes in direction, (b) the Cooperative game approach was very susceptible to changes in the direction of the generating vector field, and (c) the distributed ADMM approach was the most robust and reliable approach to estimate generating vector fields that change direction. The proposed approaches need to be adapted to simultaneously estimate multiple vector fields.[1]

1.1. Introduction

The estimation of vector fields based on pedestrian trajectories provides information about the usual motion flow of pedestrians in a scene [2, 3, 4]. This information can be used in surveillance systems to detect unusual trajectories [2], to plan better accessibility conditions according to frequent trajectories [5], or for crowd analysis [6]. More generally, vector field estimation can be used in segmentation [7, 8], autonomous navigation systems [9, 10], video compression or interpolation [11, 12], among others [13].

In many surveillance applications the area of interest is very large or includes alleys or corners. Thus, images from a single camera may not cover the entire area of interest, and the images of two or more cameras need to be combined. In this case, it is reasonable to assume there is some overlap between the images of the different cameras to make sure the whole area is covered and there are no blind spots. Multi-camera systems involve several constraints [14, 15, 16]. One of them, particularly interesting to this work, is to ensure coherence between the estimated vector fields using information from several cameras. This setting motivates the use of distributed optimization algorithms.

[1] Parts of this chapter were published on ACIVS'18.

In distributed optimization algorithms, several agents estimate a set of variables without the need for a central coordinating agent, nor widespread knowledge of every variable. Distributed algorithms usually involve two steps: the communication step and the computation step. In the communication step, each agent shares its local information (*e.g.*, the new estimates) with its neighbours. In the computation step, each agent minimizes its local cost function using information shared by its neighbours in the prior communication step. Some popular distributed estimation approaches are based on the Alternating Direction Method of Multipliers (ADMM) [17], or on game theory [18].

This work focuses on the estimation of a single vector field, not necessarily uniform in the image plane, that describes the observed pedestrian trajectories. The proposed distributed approaches to estimate vector fields from pedestrian trajectories are: (1) a purely decentralized approach without communication between agents, (2) a distributed estimation approach based on a cooperative game, and (3) a distributed version of ADMM. The performance of these approaches in the estimation of different types of vector fields, using different number of trajectories, and with different overlaps between camera images is important to select the best approach for each scenario. Moreover, it should be relevant to the set-up of multicamera surveillance systems according to the geometry of the scene, and to optimize existing set-ups.

The rest of this work is organized as follows. Section 1.2 describes the dynamic model of pedestrian trajectories, the parametric vector field representation, and the basic cost function for vector field estimation. Section 1.3 describes the three proposed distributed estimation approaches. Then, Section 1.4 presents experiments on synthetic and real pedestrian trajectory data, followed by some conclusions and references to future work in Section 1.5.

1.2. Dynamic Model and Vector Field Estimation

1.2.1. Dynamic Model of Pedestrian Trajectories

This work assumes that the pedestrian trajectories, $\boldsymbol{x}(t), t = 1, \ldots, L$, on the image plane are driven by the vector field, $\boldsymbol{T}(\boldsymbol{x})$, according to

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + \boldsymbol{T}(\boldsymbol{x}(t-1)) + \boldsymbol{w}(t)\,, \tag{1.1}$$

where $\boldsymbol{x}(t) \in [0,1]^2, \forall\, t$, $\boldsymbol{T} : [0,1]^2 \to \mathbb{R}^2$, and $\boldsymbol{w}(t) \sim \mathcal{N}(0, \sigma^2\,\mathbf{I}), \forall\, t$ is a white random perturbation.

1.2.2. Vector Field Representation

The vector field is defined only at the grid nodes of an over-imposed regular, uniform grid on the image plane, $\mathcal{G} = \{\boldsymbol{g}_i \in [0,1]^2, i = 1, \ldots, N\}$, that contains the open scene and the pedestrian trajectories. However, the pedestrian trajectories can be defined in any coordinate, even if they do not correspond to a grid node ($\boldsymbol{x}(t) \notin \mathcal{G}$). Therefore, it is necessary to use a bilinear interpolation to represent the vector field that drives the

trajectories on any coordinate of an image. The bilinear interpolation is given by

$$\boldsymbol{T}(\boldsymbol{x}_s(t)) = \sum_{n=1}^{N} \boldsymbol{\phi}_n(\boldsymbol{x}_s(t))\, \boldsymbol{t}_n \,, \tag{1.2}$$

where $\boldsymbol{\phi}_n(\boldsymbol{x}_s(t))$ are the interpolation coefficients of the velocity vectors, $\boldsymbol{t}_n$, at the grid nodes.

1.2.3. Vector Field Estimation

The vector field estimate $\boldsymbol{T}$ that drives a set of S collected trajectories $\mathcal{X} = \{\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_S(t)\}$ is the minimizer of the following cost function

$$f(\boldsymbol{T}) = \|\boldsymbol{V} - \boldsymbol{T}\,\boldsymbol{\Phi}\|_2^2 + \alpha\, \|\Delta\boldsymbol{T}\|_2^2 + \beta\, \|\boldsymbol{T}\|_1 \,, \tag{1.3}$$

where $\|.\|_p$, $p \in \{1, 2\}$ defines the pth norm of a vector, $\boldsymbol{T} \in \mathbb{R}^{2\times N}$, $\boldsymbol{V} \in \mathbb{R}^{2\times M}$, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$, $M = \sum_{s=1}^{S}(L_s - 1)$, are given by

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{t}_1 & \ldots & \boldsymbol{t}_N \end{bmatrix}, \tag{1.4}$$

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1(2) \ldots \boldsymbol{v}_1(L_1) & | & \ldots & | & \boldsymbol{v}_S(2) \ldots \boldsymbol{v}_S(L_S) \end{bmatrix}, \tag{1.5}$$

$$\boldsymbol{\Phi} = \left[\begin{array}{c|c|c} \boldsymbol{\phi}_1(\boldsymbol{x}_1(1)) \ldots \boldsymbol{\phi}_1(\boldsymbol{x}_1(L_1 - 1)) & & \boldsymbol{\phi}_1(\boldsymbol{x}_S(1)) \ldots \boldsymbol{\phi}_1(\boldsymbol{x}_S(L_S - 1)) \\ \vdots & \ldots & \vdots \\ \boldsymbol{\phi}_N(\boldsymbol{x}_1(1)) \ldots \boldsymbol{\phi}_N(\boldsymbol{x}_1(L_1 - 1)) & & \boldsymbol{\phi}_N(\boldsymbol{x}_S(1)) \ldots \boldsymbol{\phi}_N(\boldsymbol{x}_S(L_S - 1)) \end{array}\right], \tag{1.6}$$

with matrix $\boldsymbol{V}$ composed of the velocity vectors between consecutive target positions, *i.e.*, $\boldsymbol{v}_s(t) = \boldsymbol{x}_s(t) - \boldsymbol{x}_s(t-1)$. In (1.3), the first term is the usual data fidelity criterion, the second term refers to smoothness between pairs of neighbouring grid nodes, $(\boldsymbol{g}_1, \boldsymbol{g}_2) \in \mathcal{G}$, such that the velocity difference, $\Delta\boldsymbol{T} = \boldsymbol{t}_1 - \boldsymbol{t}_2$, should be small, and the thrid term refers to sparsity, such that in most grid nodes the velocity, $\boldsymbol{t}_n$, $n = \{1, \ldots, N\}$, should be small.

1.3. Distributed Estimation Algorithms

1.3.1. Notation

For each of the proposed vector field estimation approaches, there is a set of camera images corresponding to estimator agents $i \in \mathcal{A} = \{1, 2, \ldots, A\}$. An agent i has at least one neighbour, $j \in \mathcal{N}_i$. The combined image from all agents is composed of several regions divided by the agents margins, $\mathcal{R} = \{1, 2, \ldots, R\}$. Each agent is divided into sub-regions, $r \in \mathcal{R}_i \subset \mathcal{R}$, that may overlap with neighbour agents. An overlapping sub-region of an agent i with its neighbour j is defined as $o \in \mathcal{O}_j^{(i)} \cap \mathcal{R}_i$, $j \in \mathcal{N}_i$, where $\mathcal{O}_j^{(i)}$ is the set of overlapping sub-regions of agent i with each of its neighbours j.

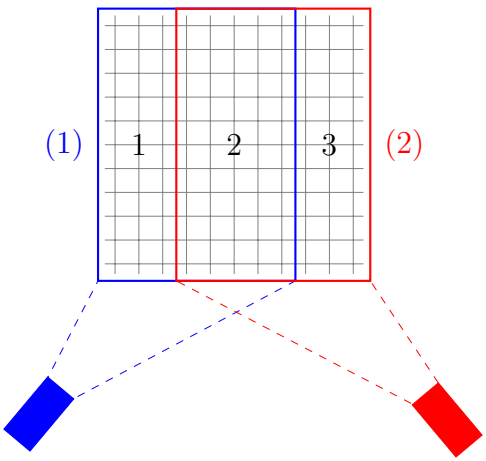


Figure 1.1.: Example of two estimator agents $\mathcal{A} = \{1, 2\}$ (blue and red), that are neighbours, *i.e.*, $\mathcal{N}_1 = \{2\}$ and vice versa.

In the example of Figure 1.1, the full set of regions is $\mathcal{R} = \{1, 2, 3\}$. Considering a regular grid of size $[11 \times 11]$ on top of the combined image, the set of sub-regions of agent 1, $\mathcal{R}_1 = \{1, 2\}$, includes the left-most nodes (*i.e.*, covering 11×3 nodes) and the center nodes (*i.e.*, covering 11×5 nodes). The set of sub-regions of agent 2, $\mathcal{R}_2 = \{2, 3\}$, includes the right-most nodes (*i.e.*, covering 11×3 nodes) and the center nodes (*i.e.*, covering 11×5 nodes). The overlapping regions of each agent are $\mathcal{O}_2^{(1)} = \mathcal{O}_1^{(2)} = \{2\}$, which include only the center nodes.

In this work, each camera and the respective field of view corresponds to an estimator agent, which yields its own vector field estimate. A vector field estimate is the set of estimated velocity vectors sitting on the grid nodes of the respective agent, *i.e.*, $\bar{\boldsymbol{T}}^{(i)} = \{\boldsymbol{t}_g\}, g \in \mathcal{R}_i$. Whenever applicable, $\bar{\boldsymbol{T}}^{(i)}$ are agent-specific vector field estimates, and $\boldsymbol{T}_r^{(i)}$, $r \in \mathcal{R}_i$ are sub-region-specific vector field estimates.

1.3.2. Purely Decentralized Vector Field Estimation

The first approach is a purely decentralized estimation problem in which communication between agents is not allowed. This approach aims to minimize the local cost functions, f_i defined as in (1.3). The problem to be solved is

$$\begin{aligned} \underset{\bar{\boldsymbol{T}}^{(i)}}{\text{minimize}} \quad & f_i(\bar{\boldsymbol{T}}^{(i)}) \\ \text{subject to} \quad & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t)\,, i \in \mathcal{A}\,. \end{aligned} \tag{1.7}$$

for each agent i, where $\boldsymbol{x}^{(i)}$ are the pedestrian trajectory data within each region $\mathcal{R}_i$. Because there is no communication between agents, (1.7) can be solved in parallel. The global cost function is the sum over all estimator agents: $F = \sum_{i=1}^{A} f_i(\bar{\boldsymbol{T}}^{(i)})$.

1.3.3. Distributed Vector Field Estimation Based on a Cooperative Game

The second approach is based on a distributed cooperative game, which aims to minimize a global cost function. There are three actions, *i.e.* strategies, each agent can follow to solve the vector field estimation problem, in each game iteration: keep the previous strategy and share its estimate of the last stable solution that yielded the global minimizer; cooperate with the neighbours and share an altruist estimate that solves their vector field estimation problems; or defect and selfishly ask its neighbours to share what it believes to be the best input for its own vector field estimation problem. Thus, in this approach, each agent estimates its vector field, $\bar{\boldsymbol{T}}^{(i)}$, assuming the neighbour agents share either (a) the last global minimizer (*i.e.*, the stable solution, $\bar{\boldsymbol{T}}_s^{(j)}$); (b) the wanted local minimizer of agent i (*i.e.*, altruist, $\bar{\boldsymbol{T}}_a^{(j)}$); or (c) their local minimizer (*i.e.*, selfish, $\bar{\boldsymbol{T}}_*^{(j)}$) [19].

The cooperative game algorithm has 2 computation instances, that can be done in parallel, and 2 communication instances. Each agent computes local vector field estimates and cost function values, and shares local vector field estimates with its neighbours and local cost function values with all agents. The following sub-sections formulate the problem of vector field estimation for the case of two cameras with overlap and then generalize it to the case of multiple agents.

Problem Formulation – The case of 2 agents

Consider the example of an open scene where the images of 2 cameras overlap (Figure 1.1). The problem to be solved is

$$\begin{aligned}
\underset{\bar{\boldsymbol{T}}^{(1)},\bar{\boldsymbol{T}}^{(2)}}{\text{minimize}} \quad & f_1(\bar{\boldsymbol{T}}^{(1)}|\boldsymbol{v}_1^{(1)},\boldsymbol{T}_2^{(2)}) + f_2(\bar{\boldsymbol{T}}^{(2)}|\boldsymbol{v}_3^{(2)},\boldsymbol{T}_2^{(1)}) \\
\text{subject to} \quad & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t)\,, i \in \{1,2\}\,, \\
& \boldsymbol{v}_r^{(i)}(t) = \boldsymbol{x}_r^{(i)}(t) - \boldsymbol{x}_r^{(i)}(t-1)\,, r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}\,, \\
& \boldsymbol{T}_o^{(j)} = \{\boldsymbol{T}_s^{(j)}, \boldsymbol{T}_*^{(j)}, \boldsymbol{T}_a^{(j)}\},\, o \in \mathcal{O}_i^{(j)},\, j \in \mathcal{N}_i\,,
\end{aligned} \tag{1.8}$$

where $\boldsymbol{x}_r^{(i)}$ and $\boldsymbol{v}_r^{(i)}$ are the pedestrian trajectory data and respective velocities within agent i sub-region $r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}$ (where $A \setminus B$ represents the set of elements of set A that are not in set B), and with the local cost functions defined as in (1.3).

Each agent i computes the selfish vector field estimate according to

$$\bar{\boldsymbol{T}}_*^{(i),k+1} = \underset{\bar{\boldsymbol{T}}^{(i)}}{\arg\min}\; f_i(\bar{\boldsymbol{T}}^{(i)}|\boldsymbol{v}_r^{(i)},\boldsymbol{T}_s^{(j),k})\,, \tag{1.9}$$

which considers local data, $\boldsymbol{v}_r^{(i)}$, $r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}$, and the shared stable vector field estimates from the neighbour regarding the overlapping region, $\boldsymbol{T}_s^{(j),k}$. Each agent i also computes the altruist vector field estimate it wishes to receive from its neighbour according to

$$\bar{\boldsymbol{T}}_a^{(j),k+1} = \underset{\bar{\boldsymbol{T}}^{(j)}}{\arg\min}\; f_i(\bar{\boldsymbol{T}}^{(j)}|\boldsymbol{v}_o^{(i)},\boldsymbol{T}_*^{(i),k+1})\,, \tag{1.10}$$

which considers the data from the overlapping region, $\boldsymbol{v}_o^{(i)}$, $o = \{2\} \in \mathcal{O}_i^{(j)}$, $j \in \mathcal{N}_i$, and its own selfish vector field estimate from its local sub-region, $\boldsymbol{T}_*^{(i),k+1}$.

After computing their new estimates, agent 1 shares $\bar{\boldsymbol{T}}_*^{(1),k+1}$ and $\bar{\boldsymbol{T}}_a^{(2),k+1}$ with agent 2, and gets $\bar{\boldsymbol{T}}_*^{(2),k+1}$ and $\bar{\boldsymbol{T}}_a^{(1),k+1}$. These estimates are then used to compute each agent local cost function values for each of the 9 combinations of the possible 3 estimates. Finally, the agents share the computed local cost function values, which are summed to yield the global cost function used for decision making. The stable estimates for the next iteration are the minimizers of Table 1.1.

Table 1.1.: Global cost functions for each of the 9 decision making scenarios involving 2 estimator agents. The pair of the collective best estimates for each agent yields the minimum of this table.

	$\boldsymbol{T}_s^{(2)}$	$\boldsymbol{T}_*^{(2)}$	$\boldsymbol{T}_a^{(2)}$
$\boldsymbol{T}_s^{(1)}$	$f_1(\bar{\boldsymbol{T}}_s^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_s^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_s^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_s^{(1)})$	$f_1(\bar{\boldsymbol{T}}_s^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_*^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_*^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_s^{(1)})$	$f_1(\bar{\boldsymbol{T}}_s^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_a^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_a^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_s^{(1)})$
$\boldsymbol{T}_*^{(1)}$	$f_1(\bar{\boldsymbol{T}}_*^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_s^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_s^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_*^{(1)})$	$f_1(\bar{\boldsymbol{T}}_*^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_*^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_*^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_*^{(1)})$	$f_1(\bar{\boldsymbol{T}}_*^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_a^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_a^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_*^{(1)})$
$\boldsymbol{T}_a^{(1)}$	$f_1(\bar{\boldsymbol{T}}_a^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_s^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_s^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_a^{(1)})$	$f_1(\bar{\boldsymbol{T}}_a^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_*^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_*^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_a^{(1)})$	$f_1(\bar{\boldsymbol{T}}_a^{(1)}\|\boldsymbol{v}_1^{(1)}, \boldsymbol{T}_a^{(2)})+$ $f_2(\bar{\boldsymbol{T}}_a^{(2)}\|\boldsymbol{v}_3^{(2)}, \boldsymbol{T}_a^{(1)})$

Problem Formulation – The case of multiple agents

Now, generalize the problem formulation to multiple agents with possibly more than 1 neighbour agent and overlapping regions. Note that Table 1.1 becomes multi-dimensional. Thus, the generalized problem formulation focuses on the concepts described above. The problem to be solved is

$$\begin{aligned}
\underset{\bar{\boldsymbol{T}}^{(1)},\ldots,\bar{\boldsymbol{T}}^{(A)}}{\text{minimize}} \quad & \sum_{i=1}^{A} f_i\big(\bar{\boldsymbol{T}}^{(i)}|\boldsymbol{v}_r^{(i)},\, \boldsymbol{T}_o^{(j)}\big) \\
\text{subject to} \quad & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t),\, i \in \mathcal{A}\,, \\
& \boldsymbol{v}_r^{(i)}(t) = \boldsymbol{x}_r^{(i)}(t) - \boldsymbol{x}_r^{(i)}(t-1)\,, r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}\,, \\
& \boldsymbol{T}_o^{(j)} \in \{\boldsymbol{T}_s^{(j)},\, \boldsymbol{T}_*^{(j)},\, \boldsymbol{T}_a^{(j)}\},\, o \in \mathcal{O}_i^{(j)},\, j \in \mathcal{N}_i\,,
\end{aligned} \tag{1.11}$$

where $\boldsymbol{x}_r^{(i)}$ and $\boldsymbol{v}_r^{(i)}$ are the pedestrian trajectory data and respective velocities within agent i sub-region $r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}$, $j \in \mathcal{N}_i$.

Each agent i computes the selfish vector field estimate according to

$$\bar{\boldsymbol{T}}_*^{(i),k+1} = \underset{\bar{\boldsymbol{T}}^{(i)}}{\arg\min}\, f_i(\bar{\boldsymbol{T}}^{(i)}|\boldsymbol{v}_r^{(i)}, \boldsymbol{T}_s^{(j),k})\,, \tag{1.12}$$

which considers local data, $\boldsymbol{v}_r^{(i)}$, $r \in \mathcal{R}_i \setminus \mathcal{O}_j^{(i)}$, and the shared stable vector field estimates from its neighbours regarding the overlapping regions, $\boldsymbol{T}_s^{(j),k}$. Each agent i also computes

the altruist vector field estimate it wishes to receive from its neighbours according to

$$\bar{\boldsymbol{T}}_a^{(j),k+1} = \arg\min_{\bar{\boldsymbol{T}}^{(j)}} f_i(\bar{\boldsymbol{T}}^{(j)} | \boldsymbol{v}_o^{(i)}, \boldsymbol{T}_*^{(i),k+1}) \,, \tag{1.13}$$

which considers the data from the overlapping regions, $\boldsymbol{v}_o^{(i)}$, $o \in \mathcal{O}_i^{(j)}$, $j \in \mathcal{N}_i$, and its own selfish vector field estimate from its local sub-region, $\boldsymbol{T}_*^{(i),k+1}$.

Then, the agents share their new estimates with their neighbours. This way, each agent can compute its local cost function, defined as in (1.3), where $\boldsymbol{V}^{(i)} = \left[\boldsymbol{v}_r^{(i)} \quad \{\boldsymbol{T}_o^{(j)}\}_{j\in\mathcal{N}_i}\right]$ considers the neighbours possible new estimates regarding the overlapping regions. The updated local cost function values are also shared. Finally, the decision about which estimate each agent should select is made considering the sums of the local cost functions over all agents, $F = \sum_{i=1}^{A} f_i\big(\bar{\boldsymbol{T}}^{(i)} | \boldsymbol{v}_r^{(i)}, \boldsymbol{T}_o^{(j)}\big)$, for all the possible combinations of pairs of neighbour agents estimates, $\big(\bar{\boldsymbol{T}}^{(i)}, \{\boldsymbol{T}_o^{(j)}\}_{j\in\mathcal{N}_i}\big) \in \big\{\boldsymbol{T}_s^{(\sqcup)}, \boldsymbol{T}_*^{(\sqcup)}, \boldsymbol{T}_a^{(\sqcup)}\big\}^{|\mathcal{N}_i|+1}$.

1.3.4. Distributed Vector Field Estimation Based on ADMM

The third approach is based on the Alternating Direction Method of Multipliers (ADMM). The following sub-sections introduce the general formulation of ADMM, formulate the problem of vector field estimation for the case of two cameras with overlap, and then generalize it to the case of multiple agents, which exploits the distributed architecture.

ADMM General Formulation

The ADMM algorithm is most useful in applications that have a non-differentiable cost function, such as the current vector field estimation problem (with cost function defined as in (1.3)). The ADMM can be applied in such cases because it belongs to the class of proximal gradient methods [20]. More specifically, ADMM is a proximal minimization algorithm applied to the dual problem, which iteratively applies the Douglas-Rachford operator to the sub-differential of the dual function [21].

The general problem formulation is as follows [17]

$$\begin{array}{ll} \underset{x_1,x_2}{\text{minimize}} & f_1(x_1) + f_2(x_2) \\ \text{subject to} & A_1\, x_1 + A_2\, x_2 = b \end{array}, \tag{1.14}$$

where $f_1 : \mathbb{R}^{n_1} \to \mathbb{R} \cup \{+\infty\}$ and $f_2 : \mathbb{R}^{n_2} \to \mathbb{R} \cup \{+\infty\}$, are closed and convex functions; $x_1 \in \mathbb{R}^n$, $x_2 \in \mathbb{R}^m$, $A_1 \in \mathbb{R}^{p\times n}$, $A_2 \in \mathbb{R}^{p\times m}$, and $b \in \mathbb{R}^p$, where x_1, and x_2 are copies of some variable x. The augmented Lagrangian of (1.14) is

$$\mathcal{L}_\rho(x_1, x_2; \lambda) = f_1(x_1) + f_2(x_2) + \lambda^\top (A_1\, x_1 + A_2\, x_2 - b) + \frac{\rho}{2}\|A_1\, x_1 + A_2\, x_2 - b\|^2 \,, \tag{1.15}$$

with penalty parameter, $\rho > 0$. The general ADMM algorithm updates are

$$x_1^{k+1} = \arg\min_{x_1} f_1(x_1) + (\lambda^k)^\top A_1\, x_1 + \frac{\rho}{2}\, \|A_1\, x_1 + A_2\, x_2^k - b\|^2\,, \tag{1.16}$$

$$x_2^{k+1} = \arg\min_{x_2} f_2(x_2) + (\lambda^k)^\top A_2\, x_2 + \frac{\rho}{2}\, \|A_1\, x_1^{k+1} + A_2\, x_2 - b\|^2\,, \tag{1.17}$$

$$\lambda^{k+1} = \lambda^k + \rho\,(A_1\, x_1^{k+1} + A_2\, x_2^{k+1} - b)\,, \tag{1.18}$$

where x_2 and λ represent the algorithm state, and x_1 and x_2 can be computed in sequence or in alternation.

Problem Formulation – The case of 2 agents

Consider the example of an open scene where the images of 2 cameras overlap (Figure 1.1). In this approach, the vector field estimates of each agent in the overlapping region must be such that $\boldsymbol{T}_2^{(1)}$ and $\boldsymbol{T}_2^{(2)}$ are equal. The problem to be solved is

$$\begin{aligned} \underset{\bar{\boldsymbol{T}}^{(1)},\bar{\boldsymbol{T}}^{(2)}}{\text{minimize}} \quad & f_1(\bar{\boldsymbol{T}}^{(1)}) + f_2(\bar{\boldsymbol{T}}^{(2)}) \\ \text{subject to} \quad & \boldsymbol{T}_2^{(1)} = \boldsymbol{T}_2^{(2)}\,, \\ & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t)\,, i \in \{1,2\}\,, \end{aligned} \tag{1.19}$$

where $\boldsymbol{x}^{(i)}$ are the pedestrian trajectory data within each agent region $\mathcal{R}_i$, and the local cost functions are defined as in (1.3). The augmented Lagrangian of (1.19) is defined as

$$\mathcal{L}_\rho(\bar{\boldsymbol{T}}^{(1)}, \bar{\boldsymbol{T}}^{(2)}; \lambda) = f_1(\bar{\boldsymbol{T}}^{(1)}) + f_2(\bar{\boldsymbol{T}}^{(2)}) + \lambda^\top(\boldsymbol{T}_2^{(1)} - \boldsymbol{T}_2^{(2)}) + \frac{\rho}{2}\, \|\boldsymbol{T}_2^{(1)} - \boldsymbol{T}_2^{(2)}\|^2\,, \tag{1.20}$$

with penalty parameter, $\rho > 0$. The penalty parameter is a tuning parameter that provides a balance between accurate vector field estimates and consensus between neighbour agent estimates in the overlapping region. The ADMM algorithm updates are

$$\bar{\boldsymbol{T}}^{(1),k+1} = \arg\min_{\bar{\boldsymbol{T}}^{(1)}} f_1(\bar{\boldsymbol{T}}^{(1)}) + \lambda^{k\top}\boldsymbol{T}_2^{(1)} + \frac{\rho}{2}\, \|\boldsymbol{T}_2^{(1)} - \boldsymbol{T}_2^{(2),k}\|^2\,, \tag{1.21}$$

$$\bar{\boldsymbol{T}}^{(2),k+1} = \arg\min_{\bar{\boldsymbol{T}}^{(2)}} f_2(\bar{\boldsymbol{T}}^{(2)}) - \lambda^{k\top}\boldsymbol{T}_2^{(2)} + \frac{\rho}{2}\, \|\boldsymbol{T}_2^{(1),k+1} - \boldsymbol{T}_2^{(2)}\|^2\,, \tag{1.22}$$

$$\lambda^{k+1} = \lambda^k + \rho\,(\boldsymbol{T}_2^{(1),k+1} - \boldsymbol{T}_2^{(2),k+1})\,. \tag{1.23}$$

Thus, there are only 2 communication instances in each iteration: (a) agent 1 shares its new estimate with agent 2, (b) both agents get the new estimates and the updated Lagrange multipliers.

Problem Formulation – The case of multiple agents

Now, generalize the problem formulation to multiple agents with possibly more than 1 neighbour agent and overlapping regions. This work follows the distributed ADMM algorithm [1], adapted to the vector field estimation problem.

As before, each agent i only knows its own cost function, f_i defined as in (1.3), and the shared vector field estimates from its neighbours, $\boldsymbol{T}_o^{(j)}$, $o \in \mathcal{O}_i^{(j)} \cap \mathcal{R}_i$, $j \in \mathcal{N}_i$. The problem to be solved is

$$\begin{aligned} \underset{\bar{\boldsymbol{T}}^{(1)},\ldots,\bar{\boldsymbol{T}}^{(A)}}{\text{minimize}} \quad & \sum_{i=1}^{A} f_i(\bar{\boldsymbol{T}}^{(i)}) \\ \text{subject to} \quad & \boldsymbol{T}_o^{(i)} = \boldsymbol{T}_o^{(j)},\, o \in \mathcal{O}_i^{(j)} \cap \mathcal{R}_i\,,\, j \in \mathcal{N}_i\,, \\ & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t)\,, i \in \mathcal{A}\,, \end{aligned} \tag{1.24}$$

where $\boldsymbol{x}^{(i)}$ are the pedestrian trajectory data within each agent region $\mathcal{R}_i$. The constraints translate in each agent having a copy of the vector field estimates at its overlapping regions. The constraints on (1.24) can alternatively be written as

$$\mathcal{M}_o \tilde{\boldsymbol{T}}_o = 0,\, o \in \mathcal{O}_i^{(j)} \cap \mathcal{R}_i\,,\, j \in \mathcal{N}_i\,, \tag{1.25}$$

where $\mathcal{M}_o$ is the transpose of the neighbour-overlap (*i.e.*, node-arc) incidence matrix of the set of overlapping regions o, for all pairs of neighbour agents.

The current problem formulation is not yet a distributed optimization problem. A colouring scheme similar to [1] allows the formulation of a distributed version of ADMM by defining

$$\tilde{\boldsymbol{T}}^c = \begin{cases} \{\bar{\boldsymbol{T}}^{(i)}\}_{i \in \mathcal{A} \cap \mathcal{C}_c}, & \text{if } \mathcal{O}_j^{(i)} \cap \mathcal{R}_i = \emptyset\,, (i,j) \in \mathcal{A} \\ \emptyset, & \text{if } \mathcal{O}_j^{(i)} \cap \mathcal{R}_i \neq \emptyset\,, (i,j) \in \mathcal{A} \end{cases}, \tag{1.26}$$

where $\tilde{\boldsymbol{T}}^c$ is the set of $\bar{\boldsymbol{T}}^{(i)}$ from the agents with colour c. This same notation applied to constraint (1.25) becomes $\mathcal{M}_o^1 \tilde{\boldsymbol{T}}_o^1 + \cdots + \mathcal{M}_o^C \tilde{\boldsymbol{T}}_o^C$, $\forall_o$. Moreover, applying it to the constraints of (1.24), allows their separation into C coupled constraints, such that it is rewritten as

$$\begin{aligned} \underset{\tilde{\boldsymbol{T}}^1,\ldots,\tilde{\boldsymbol{T}}^C}{\text{minimize}} \quad & \sum_{i \in \mathcal{C}_1} f_i(\bar{\boldsymbol{T}}^{(i)}) + \cdots + \sum_{i \in \mathcal{C}_C} f_i(\bar{\boldsymbol{T}}^{(i)}) \\ \text{subject to} \quad & \tilde{\mathcal{M}}^1 \tilde{\boldsymbol{T}}^1 + \cdots + \tilde{\mathcal{M}}^C \tilde{\boldsymbol{T}}^C = 0\,, \\ & \boldsymbol{x}^{(i)}(t) = \boldsymbol{x}^{(i)}(t-1) + \bar{\boldsymbol{T}}^{(i)}(\boldsymbol{x}^{(i)}(t-1)) + \boldsymbol{w}(t)\,, i \in \mathcal{A}\,, \end{aligned} \tag{1.27}$$

where $\tilde{\mathcal{M}}^c$ is the diagonal concatenation of matrices $\mathcal{M}_1^c$, $\mathcal{M}_2^c$, $\ldots$, $\mathcal{M}_O^c$,

$$\tilde{\mathcal{M}}^c = \text{diag}(\mathcal{M}_1^c,\, \mathcal{M}_2^c,\, \ldots,\, \mathcal{M}_O^c)\,. \tag{1.28}$$

This problem can be solved using the multi-block/distributed ADMM [1], where λ_o^{ij} is the dual variable associated to $\boldsymbol{T}_o^{(i)} = \boldsymbol{T}_o^{(j)}$, $o \in \mathcal{O}_i^{(j)} \cap \mathcal{R}_i$, $j \in \mathcal{N}_i$, $\gamma = \sum_{i \in \mathcal{C}_c} \sum_{j \in \mathcal{N}_i} \lambda_o^{ij}$. The augmented Lagrangian of (1.27) is

$$\mathcal{L}_\rho(\bar{\boldsymbol{T}}^1, \ldots, \bar{\boldsymbol{T}}^c; \gamma) = \sum_{c=1}^{C} \sum_{i \in \mathcal{C}_c} f_i(\bar{\boldsymbol{T}}^{(i)}) + \sum_{c=1}^{C} \gamma^\top \bar{\mathcal{M}}^c \bar{\boldsymbol{T}}^c + \frac{\rho}{2} \Big\| \sum_{c=1}^{C} \bar{\mathcal{M}}^c \bar{\boldsymbol{T}}^c \Big\|^2\,, \tag{1.29}$$

with penalty parameter, $\rho > 0$.

The problem to be solved consists of a sequence of C sub-problems, obtained by minimizing (1.29) with respect to each block $\tilde{\boldsymbol{T}}^c$, and of updates of the dual variable γ. The resulting Distributed-ADMM (D-ADMM) algorithm updates are

$$\tilde{\boldsymbol{T}}^{1,\,k+1} = \arg\min_{\tilde{\boldsymbol{T}}^1} \sum_{i\in\mathcal{C}_1} f_i(\bar{\boldsymbol{T}}^{(i)}) + \gamma^{k\,\top}\,\tilde{\mathcal{M}}^1\,\tilde{\boldsymbol{T}}^1 + \frac{\rho}{2}\Big\|\tilde{\mathcal{M}}^1\,\tilde{\boldsymbol{T}}^1 + \sum_{c=2}^{C}\tilde{\mathcal{M}}^c\,\tilde{\boldsymbol{T}}^{c,k}\Big\|^2, \tag{1.30}$$

$$\vdots$$

$$\tilde{\boldsymbol{T}}^{C,\,k+1} = \arg\min_{\tilde{\boldsymbol{T}}^C} \sum_{i\in\mathcal{C}_C} f_i(\bar{\boldsymbol{T}}^{(i)}) + \gamma^{k\,\top}\,\tilde{\mathcal{M}}^C\,\tilde{\boldsymbol{T}}^C + \frac{\rho}{2}\Big\|\sum_{c=1}^{C-1}\tilde{\mathcal{M}}^c\,\tilde{\boldsymbol{T}}^{c,k+1} + \tilde{\mathcal{M}}^C\,\tilde{\boldsymbol{T}}^C\Big\|^2, \tag{1.31}$$

$$\gamma^{k+1} = \gamma^k + \rho\sum_{c=1}^{C}\tilde{\mathcal{M}}^c\,\tilde{\boldsymbol{T}}^{c,k+1}\,. \tag{1.32}$$

Thus, there are $(C-1)+1$ communication instances in each iteration: (a) the agents of colour c share their new estimates with neighbour agents of colour $c+1$, and (b) all neighbour agents get the new estimates and the updated Lagrange multipliers. Moreover, each update $\tilde{\boldsymbol{T}}^c$ can be decomposed into $|\mathcal{C}_c|$ problems that can be solved in parallel, given that estimator agents of the same colour cannot be neighbours. Therefore, for each $i \in \mathcal{C}_c$,

$$\bar{\boldsymbol{T}}^{(i),\,k+1} = \arg\min_{\bar{\boldsymbol{T}}^{(i)}} f_i(\bar{\boldsymbol{T}}^{(i)}) + \sum_{\substack{j\in\mathcal{N}_i\\ o\in\mathcal{O}_i^{(j)}\cap\mathcal{R}_i}}\Big(\mathrm{sign}(j-i)\,\lambda_o^{ij,k} - \rho\,\boldsymbol{T}_o^{(j),k}\Big)^{\top}\boldsymbol{T}_o^{(i)} + \frac{\rho}{2}\sum_{\substack{j\in\mathcal{N}_i\\ o\in\mathcal{O}_i^{(j)}\cap\mathcal{R}_i}}\big\|\boldsymbol{T}_o^{(i)}\big\|^2, \tag{1.33}$$

can be computed in parallel. (For toy examples please see Appendix A.)

1.4. Experimental Results

This section presents vector field estimation examples from synthetic and real trajectory data, using the three proposed distributed estimation approaches described above. These examples include only 2 estimator agents (except in one example which includes 3 agents). The synthetic data examples depict which approach is better for which scenario and provide some ground truth for comparison with the real data example (*i.e.*, when estimating the V-shaped upwards vector field).

In the examples, the vector field estimation performance is assessed both regarding the magnitude and the relative angle of the estimated vectors of pairs of fields. This assessment is done *via* a vector field evaluation (VFE) diagram [22], which considers the vector similarity coefficient (R_v), the root mean square length (RMSL) and the root

mean square vector difference (RMSVD), defined as,

$$R_v = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{A}_i^* \cdot \boldsymbol{B}_i^* \,, \tag{1.34}$$

$$\text{RMSL} = L_V^2 = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{V}_i\|_2^2 \,, \tag{1.35}$$

$$\text{RMSVD} = \left[\frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{A}_i - \boldsymbol{B}_i\|_2^2\right]^{\frac{1}{2}} \,, \tag{1.36}$$

where "·" is the inner product between two normalized vector fields $\boldsymbol{A}_i^*$ and $\boldsymbol{B}_i^*$, with $\boldsymbol{V}_i^* = \frac{\boldsymbol{V}_i}{L_V}$ given $\boldsymbol{V}_i = (x_{vi}, y_{vi})$, $i = 1, 2, \ldots, N$, and $\|\cdot\|_2$ represents the l_2-norm of a vector. These metrics respectively assess the mean of the inner product of normalized vector pairs (1.34), the systematic difference in the mean vector length (1.35), and the overall difference between two vector fields (1.36). To facilitate interpretation of the results, RMSL and RMSVD are normalized with respect to L_V, where $\boldsymbol{V}$ is either the known generating vector field (in the synthetic case) or the full image estimated vector field (in the real case) when the focus is accuracy assessment, or the estimated vector field from agent 1 when the focus is consensus assessment.

The method to select the cost function and penalty parameter values for the examples using a 11×11 grid considered three steps: select α for $\beta = 0$ and $\rho = 0$, select β given the selected α and $\rho = 0$, and select ρ given the selected α and β. The optimal range (minimum of (1.3)) for α was $[0, 0.6]$. Given $\alpha = 0.2$, β induced sparsity for values larger than 10. However, this yielded vector field estimates that were not very smooth between neighbour grid nodes, which was not desirable. A subsequent study of the dependence between the values of β and ρ showed that for $\beta > 30$ there was no effect of ρ in the D-ADMM results, which was undesirable. Thus, the selection of $\beta = 0.2$ yielded a compromise between smoothness, sparsity and accuracy/consensus trade-off in D-ADMM. Finally, given $\alpha = 0.2$ and $\beta = 0.2$ the selected ρ value was 1.

The method to select the cost function and penalty parameter values for the examples using a 21×21 grid considered two steps: select α and β for $\rho = 0$ considering a single estimator agent for the full image plane, and select ρ given the selected α and β considering three estimator agents. The optimal values (minimum of (1.3)) for α and β were $(0.8, 12.17)$. This set of cost function values correctly induced sparsity in the regions where there were no available trajectories. (The smoothness of estimated vector fields was not a problem in the 21×21 example, as opposed to the 11×11 example described above.) Given these values, (1.3) for three estimator agents had low sensitivity to ρ, thus the selected ρ value was 1. The noise variance of the synthetic data was $\sigma^2 = 10^{-4}$.

1.4.1. Synthetic Pedestrian Trajectory Data

Synthetic data consisted on pedestrian trajectories of length L generated with different types of vector fields (see Figure 1.2):

- a hat vector field (drastically changing motion direction – upwards diagonal followed by downwards diagonal);
- a circular vector field (changing motion direction);
- a vertical vector field upwards;
- a V-shaped upwards vector field (composed of two upwards diagonal vector fields – to the left or to the right);
- a flow vector field (two-times changing motion direction – upwards diagonal followed by horizontal followed by a different upwards diagonal vector field).

Vector field estimation depended on:

- the number of available trajectories ($S = 5, 20, 80$);
- the width of the overlapping region, $|o| = 1, 3, 5, 7$ within an over-imposed grid of 11×11 nodes, or $|o| = 2, 4, 6$ within an over-imposed grid of 21×21 nodes.

We expected that:

- The hat vector field would demonstrate that the D-ADMM approach is more robust for estimating vector fields in the overlapping region when transitions between vector fields are not easily extrapolated between neighbour agents;
- The hat and the circular vector field would demonstrate that the Cooperative game approach is more susceptible to local optima when estimating vector fields which change direction within the overlapping regions of neighbour agents;
- All three approaches would yield good results when estimating the vertical vector field;
- The V-shape vector field would provide ground truth results for comparison with the results from vector field estimation using real pedestrian trajectory data;
- The flow vector field would resemble a wide open scene with long trajectories, thus enabling further tests of sparsity and consensus of the vector field estimates;
- More available trajectories would improve accuracy and consensus error of the estimates;
- A wider overlapping region would favour the D-ADMM approach and a narrower overlapping region would favour the Decentralized and Cooperative game approaches.

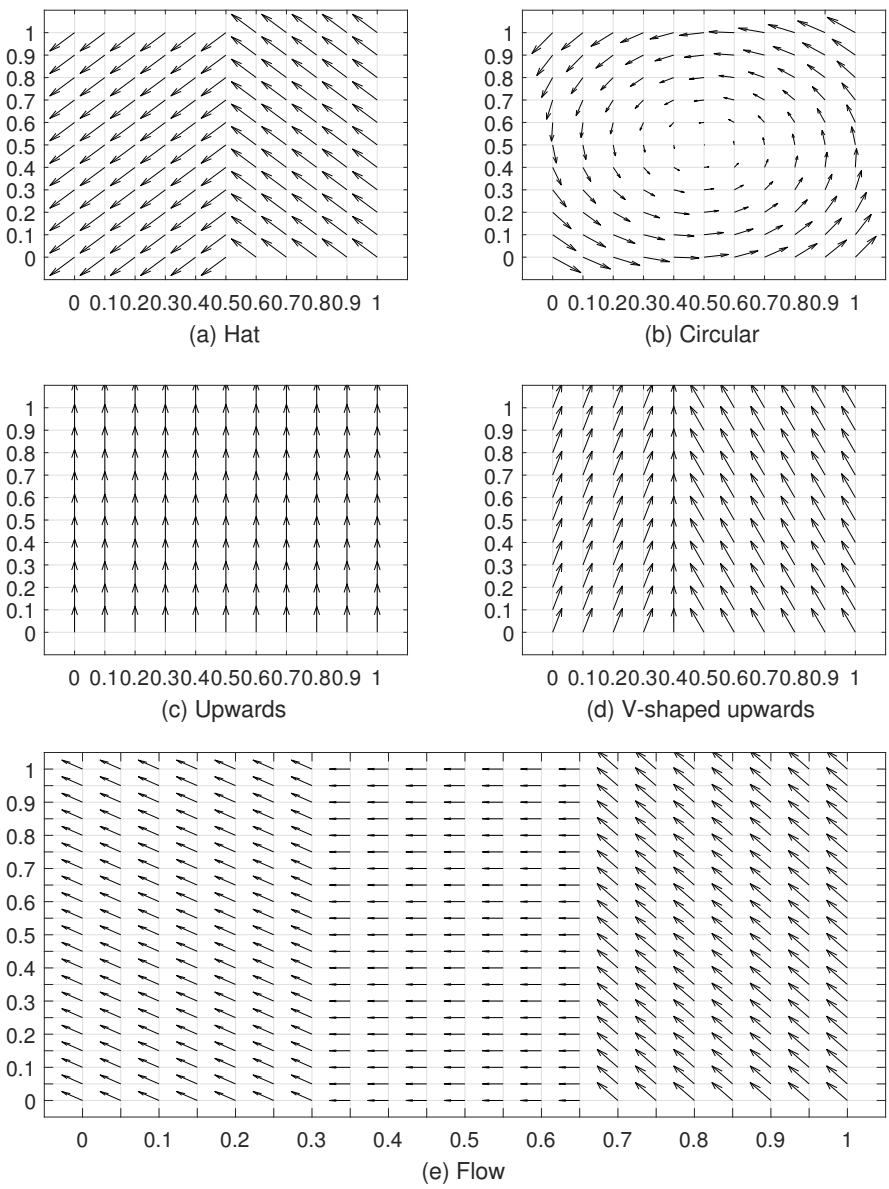


Figure 1.2.: Synthetic vector fields.

Hat vector field

The top row of Figure 1.3 shows the VFE diagram to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Higher $|o|$ yields higher angle similarity between pairs of vectors for all approaches. The

length of vectors (*i.e.*, normalized RMSL) is almost not affected by changes in $|o|$. The D-ADMM is the approach that is least affected by changes in $|o|$ (see Figure 1.4). The lack of consensus between neighbour agent estimates of the Decentralized and Cooperative game approaches is due to differences in the relative angle between the estimated vectors for low $|o|$. This effect is due to a transition between generating vector fields in the overlapping region, which is not easy to extrapolate between neighbour agents, given such short trajectory data (see Figure 1.4). Moreover, fewer available trajectories yield lower consensus between pairs of estimated vectors for low $|o|$. The D-ADMM approach is the least affected by changes in S (see Figure 1.5). Consensus assessment by the normalized $\mathrm{RMSVD_c}$ is not as clear as through the VFE diagram because the differences between vector fields are in the relative angle between pairs of estimated vectors (see Figure 1.6).

The bottom row of Figure 1.3 shows the VFE diagram to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More accurate estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors).

Lower $|o|$ yields less accurate vector field estimates (see Figure 1.6). Regarding the Decentralized and the D-ADMM approaches, larger $|o|$ improves accuracy of the estimates vector fields overall, as assessed by $\mathrm{RMSVD_{ref}}$. On the contrary, the accuracy of the estimates of the Cooperative game approach does not improve with changes in $|o|$, it even worsens for larger $|o|$, as assessed by RMSL (see Figure 1.3). Moreover, the effect of S is small for all approaches (lower-right panel of Figure 1.6). Accuracy assessment by the normalized $\mathrm{RMSVD_{ref}}$ complements the interpretation of the VFE diagram because the differences between vector fields are almost restricted to vector length within a very small interval.

Therefore, when estimating the hat vector field, the D-ADMM approach yields similar accuracy performance when compared to the Decentralized approach, but higher consensus performance than the other two approaches, as expected. This type of vector field drastically changes its direction within the overlapping region between neighbour agents and only the D-ADMM approach considers an equality constraint between neighbour agents estimates. Moreover, the Cooperative game approach is susceptible to local optima, which compromise its estimates.

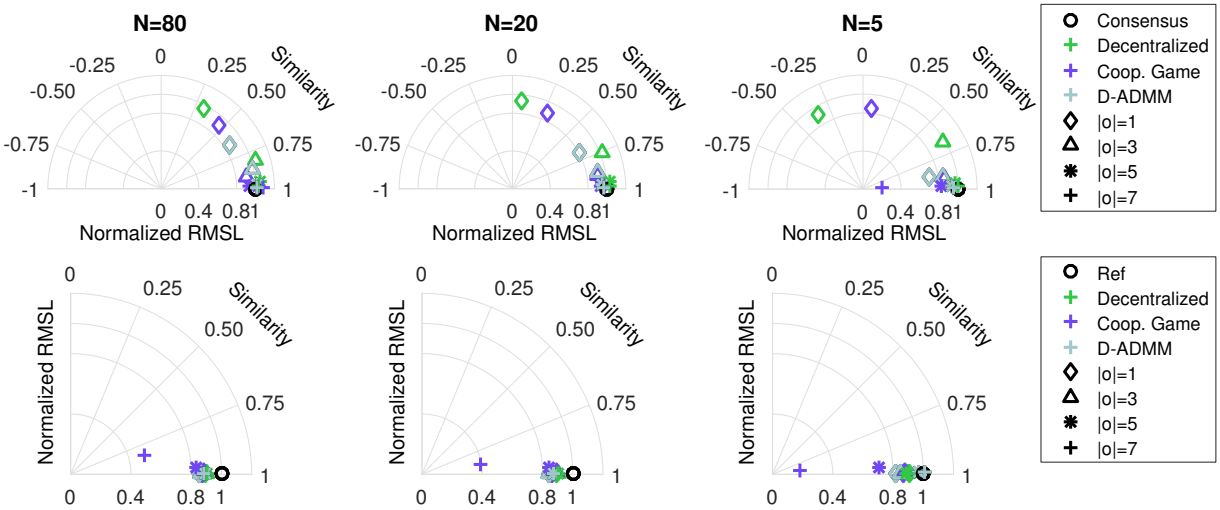
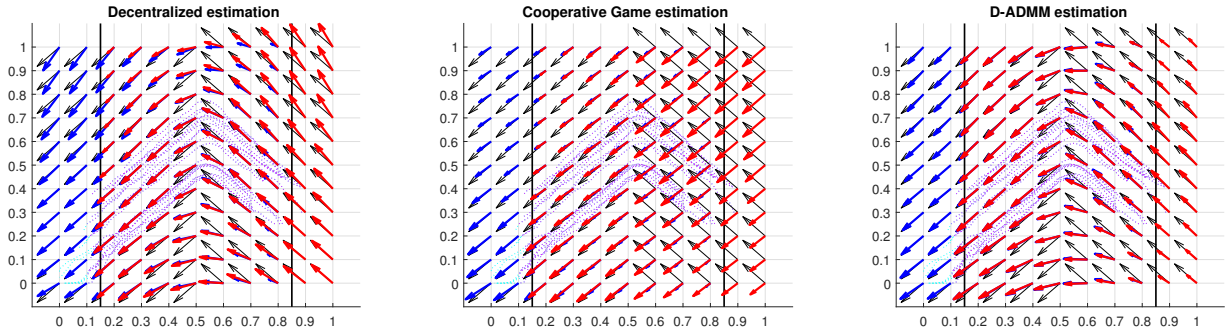
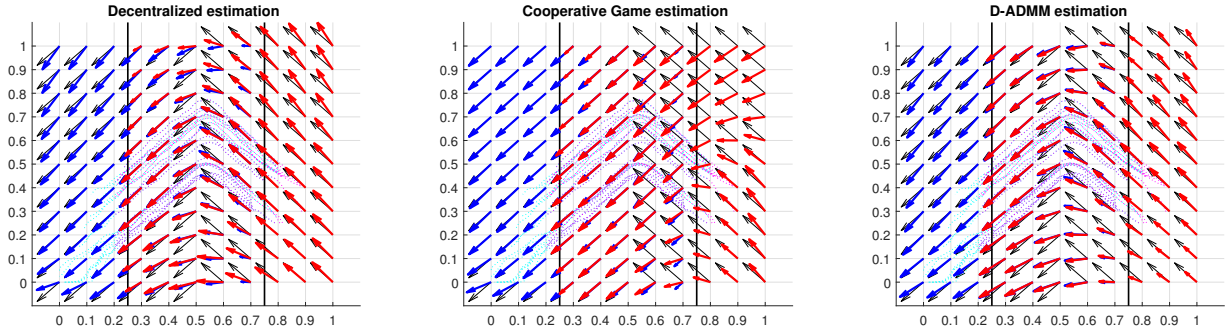


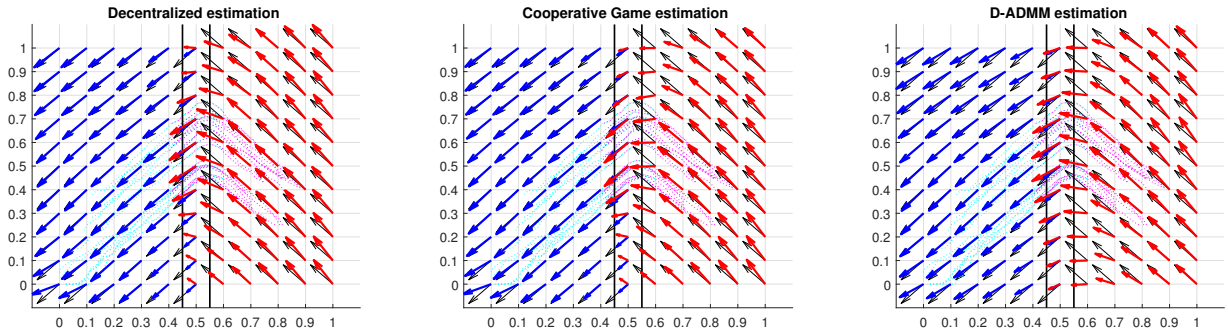
Figure 1.3.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The different symbols represent several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.



(a) $|o| = 7$ nodes

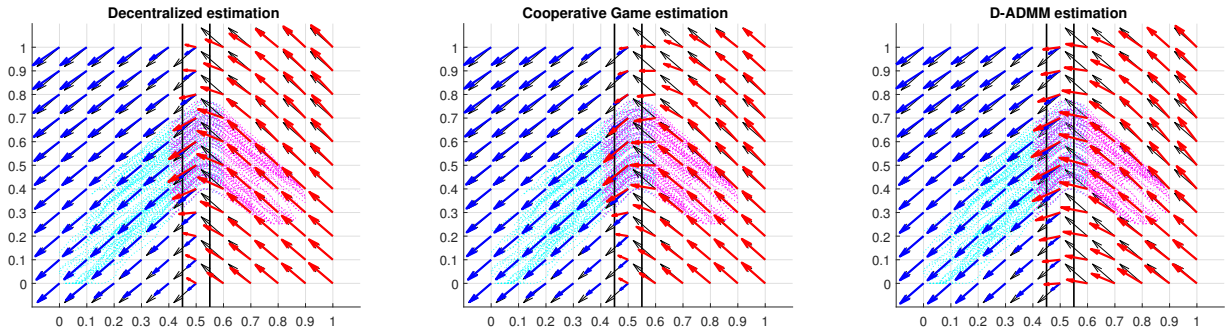


(b) $|o| = 5$ nodes

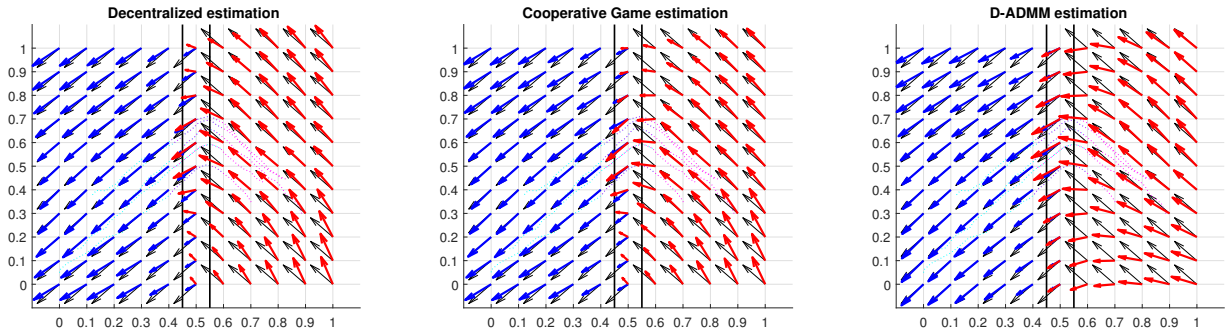


(c) $|o| = 1$ node

Figure 1.4.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 20$ trajectories. There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.



(a) $S = 80$ trajectories



(b) $S = 5$ trajectories

Figure 1.5.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using a narrow overlapping region of $|o| = 1$ node (between vertical black lines). There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.

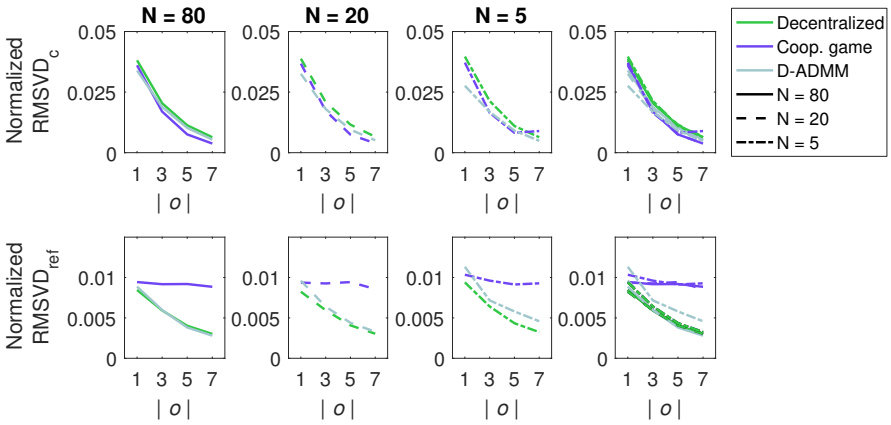


Figure 1.6.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The right-most column combines the different number of available trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Lower normalized RMSVD values represent better performance.

Circular vector field

The top row of Figure 1.7 shows the VFE diagram to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Higher $|o|$ yields higher angle similarity between pairs of vectors for all approaches. The length of vectors is almost not affected by changes in $|o|$. The Cooperative game is the approach that is most affected by changes in $|o|$ (see Figure 1.8). The lack of consensus between neighbour agent estimates is mostly due to differences in the relative angle of the estimated vectors. This effect is due to changes in the direction of the generating vector field in the overlapping region, which are not easy to extrapolate between neighbour agents. Moreover, fewer available trajectories yield lower consensus between pairs of estimated vectors for lower $|o|$. The D-ADMM approach is the least affected by changes in S (see Figure 1.9). Consensus assessment by the normalized $\mathrm{RMSVD_c}$ is not as clear as through the VFE diagram because the differences between vector fields are in the relative angle between pairs of estimated vectors (see Figure 1.10).

The bottom row of Figure 1.7 shows the VFE diagram to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More accurate estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors).

Lower $|o|$ yields less accurate vector field estimates (see Figure 1.10). Regarding the Decentralized and the D-ADMM approaches, larger $|o|$ improves accuracy of the estimates vector fields overall, as assessed by $\mathrm{RMSVD_{ref}}$. On the contrary, the accuracy of the estimates of the Cooperative game approach does not improve with changes in $|o|$, it even worsens for larger $|o|$. This effect is especially observed in the relative angle of the estimated vectors (see Figure 1.7 and Figure 1.8). Moreover, the effect of S is observed especially for the Cooperative game and the D-ADMM approaches (see Figure 1.9, and the lower-right panel of Figure 1.10). Regions where there are no available trajectories are the most affected, as expected. Accuracy assessment by the normalized $\mathrm{RMSVD_{ref}}$ complements the interpretation of the VFE diagram because the differences between vector fields are almost restricted to vector length within a very small interval.

Therefore, the Cooperative game and D-ADMM approaches are susceptible to the lack of trajectory data, yielding inaccurate estimates in those regions, as expected. However, better consensus is achieved by the D-ADMM approach. The Cooperative game approach also shows susceptibility to local optima, even for vector fields that gradually change direction, such as the circular vector field.

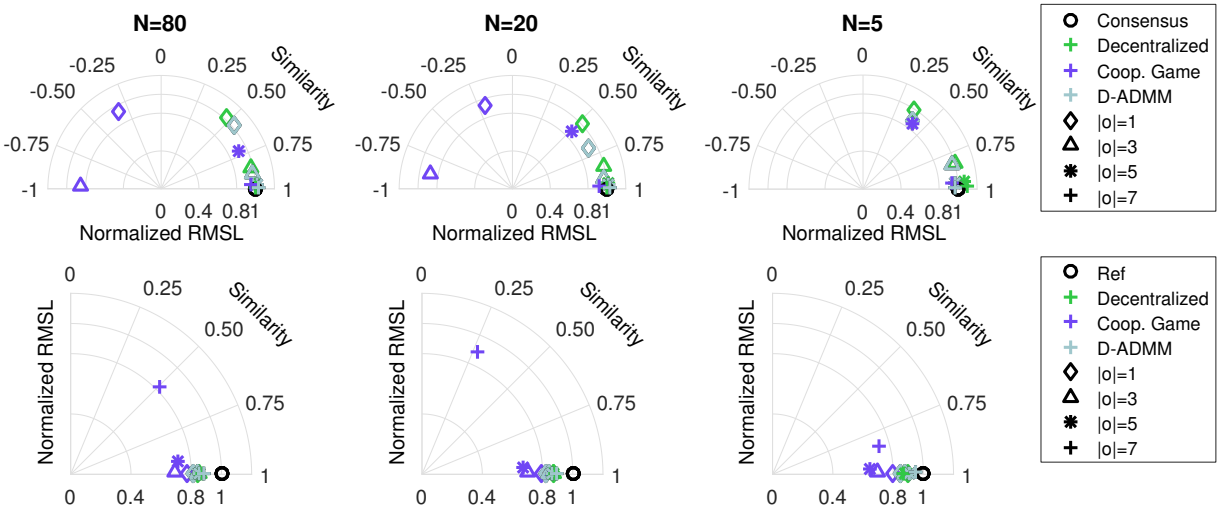
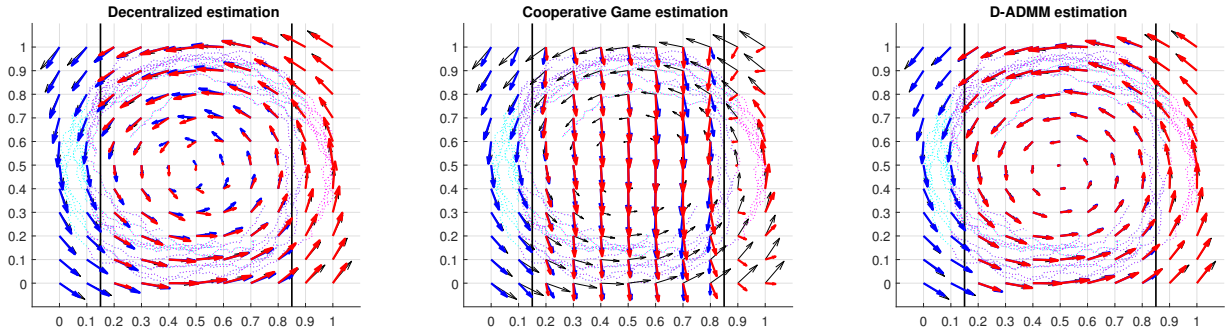
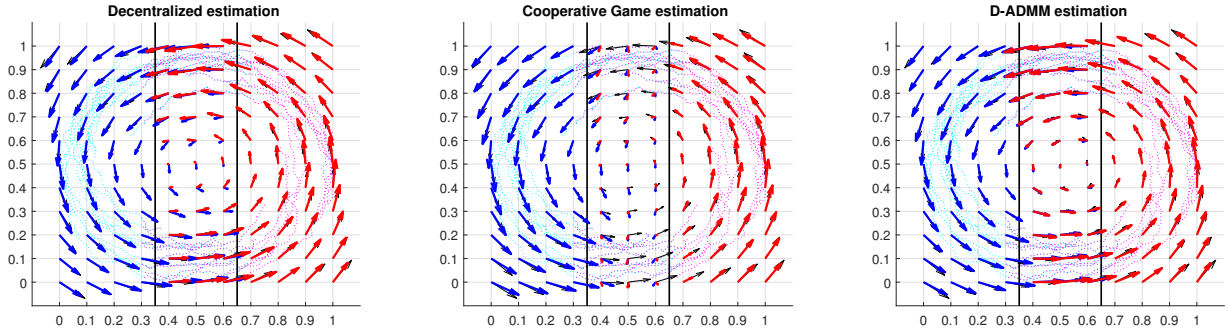


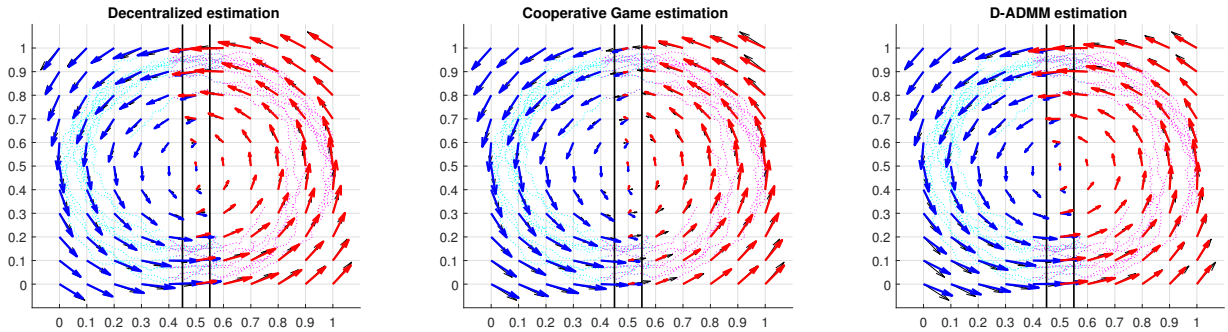
Figure 1.7.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The different symbols represent several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.



(a) $|o| = 7$ nodes

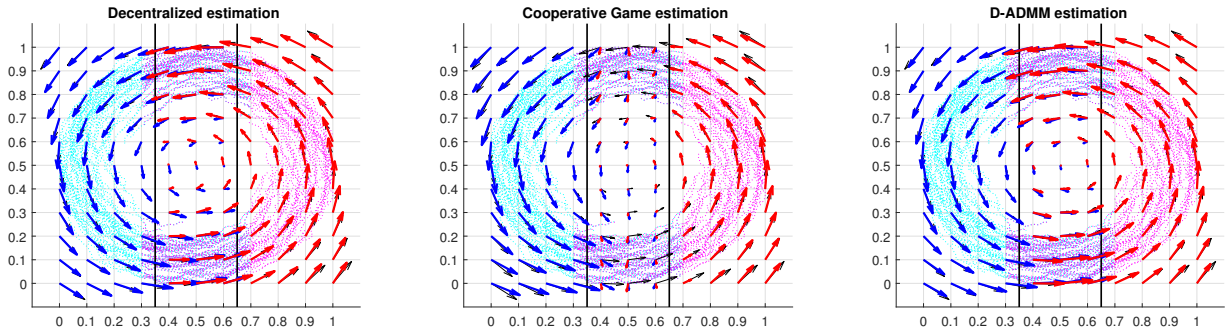


(b) $|o| = 3$ nodes

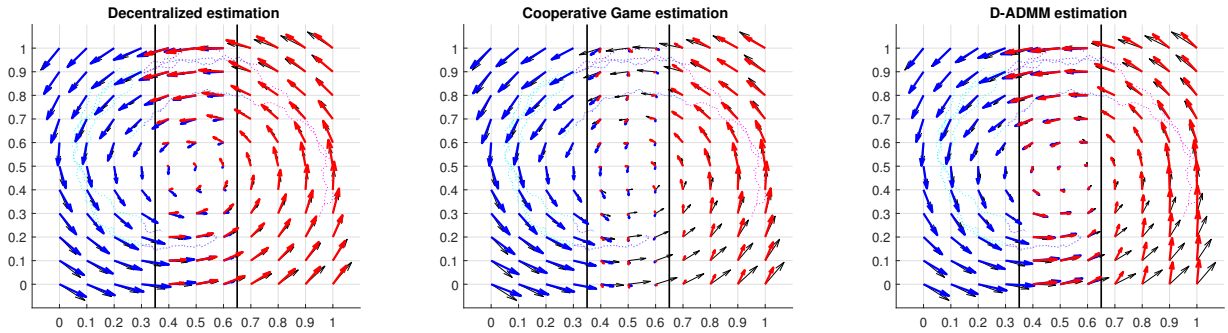


(c) $|o| = 1$ node

Figure 1.8.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 20$ trajectories. There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.



(a) $S = 80$ trajectories



(b) $S = 5$ trajectories

Figure 1.9.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using a medium width overlapping region of $|o| = 3$ nodes (between vertical black lines). There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.

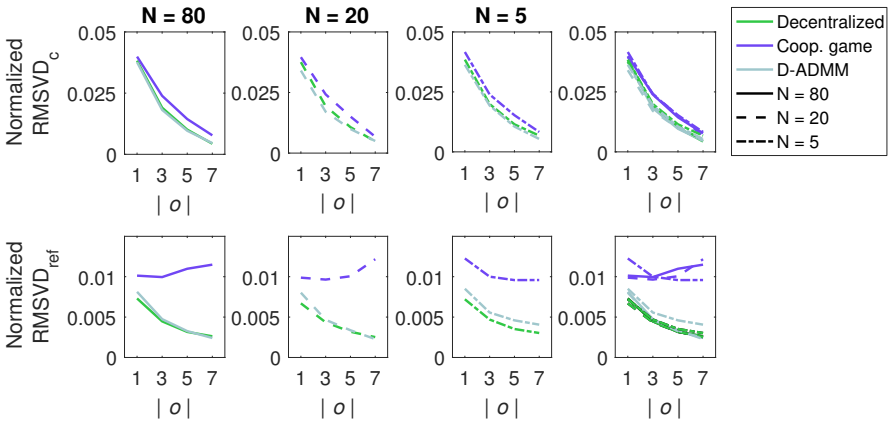


Figure 1.10.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The right-most column combines the different number of available trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Lower normalized RMSVD values represent better performance.

Upwards vector field

The top row of Figure 1.11 shows the VFE diagram to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Higher $|o|$ yields less consensual vector field estimates as assessed by $\mathrm{RMSVD_c}$, for all approaches (see Figure 1.12). However, this effect is not due to differences in the relative angle between estimated vector pairs but to differences in vector length (see Figure 1.13). This result is due to the presence of a single generating vector field, which does not change its direction. Moreover, the effect of S is small for all approaches (lower-right panel of Figure 1.12). However, fewer available trajectories yield lower consensus for the D-ADMM approach when $|o|$ is moderately high (see Figure 1.14). Consensus assessment by the normalized $\mathrm{RMSVD_c}$ complements the interpretation of the VFE diagram because the differences between vector fields are almost restricted to vector length within a very small interval.

The bottom row of Figure 1.11 shows the VFE diagram to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, N, and widths of overlap, $|o|$. More accurate estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors).

Lower $|o|$ yields less accurate vector field estimates as assessed by $\mathrm{RMSVD_{ref}}$ (see Figure 1.12). This effect is mostly observed in the length of the estimated vectors (see Figure 1.13). The D-ADMM is the approach that is least affected by changes in $|o|$. Moreover, the effect of S is observed especially for the Cooperative game (vector length) and the D-ADMM (overall) approaches for lower $|o|$. The Decentralized approach is the least affected by changes in S (lower-right panel of Figure 1.12 and Figure 1.14). Accuracy assessment by the normalized $\mathrm{RMSVD_{ref}}$ complements the interpretation of the VFE diagram because the differences between vector fields are almost restricted to vector length.

Therefore, when the generating vector field does not change its direction, it is better to use less computationally expensive approaches such as the Decentralized approach. This approach yields a more robust performance to different number of available trajectories and widths of overlap.

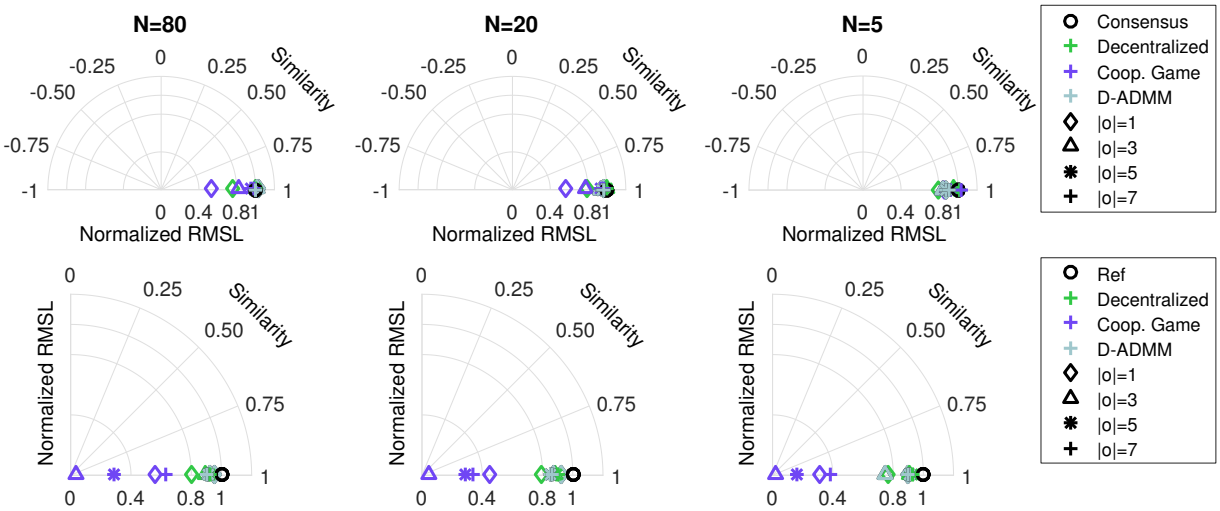


Figure 1.11.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The different symbols represent several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.

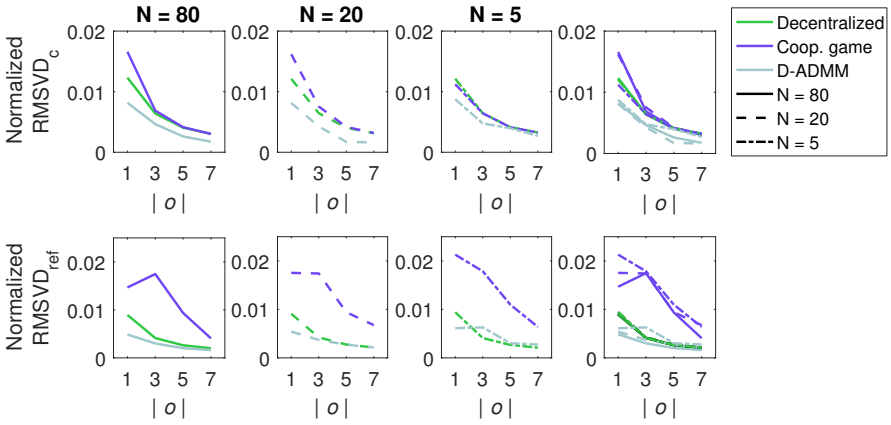
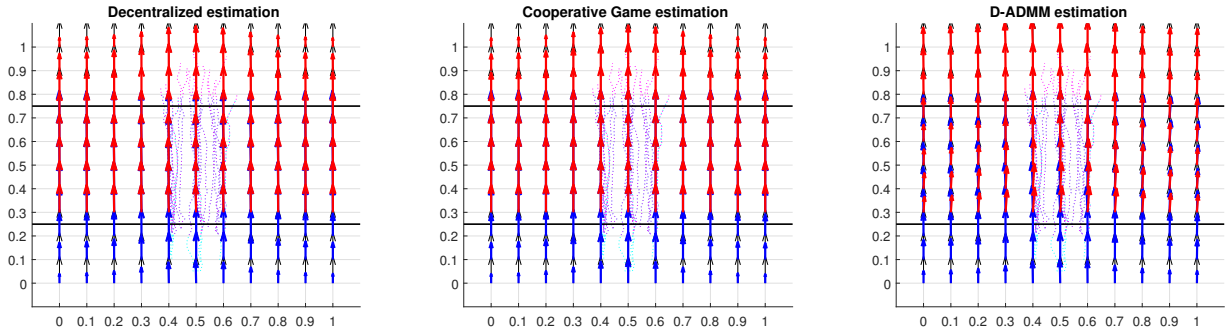
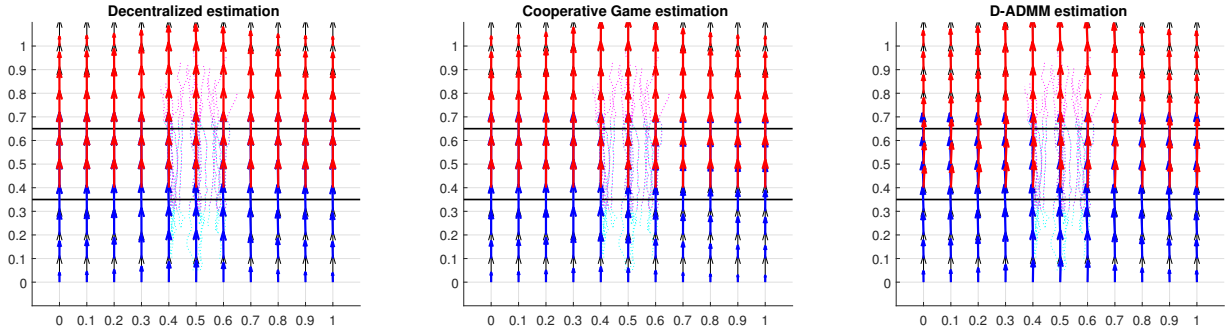


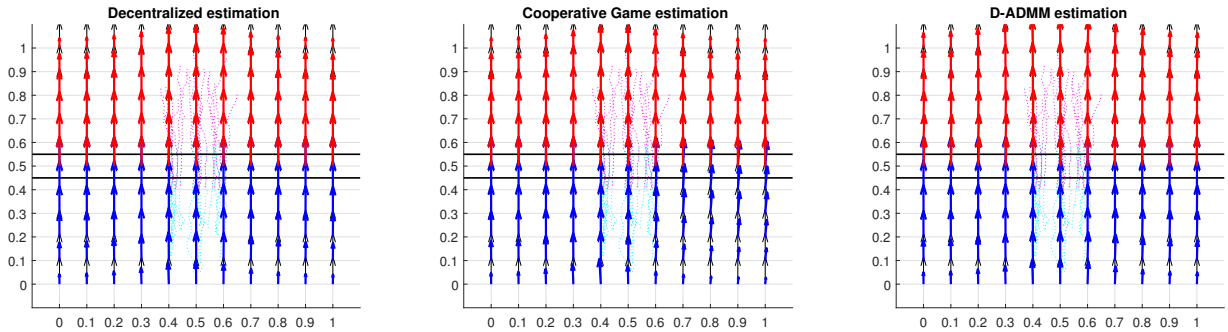
Figure 1.12.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The right-most column combines the different number of available trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Lower normalized RMSVD values represent better performance.



(a) $|o| = 5$ nodes

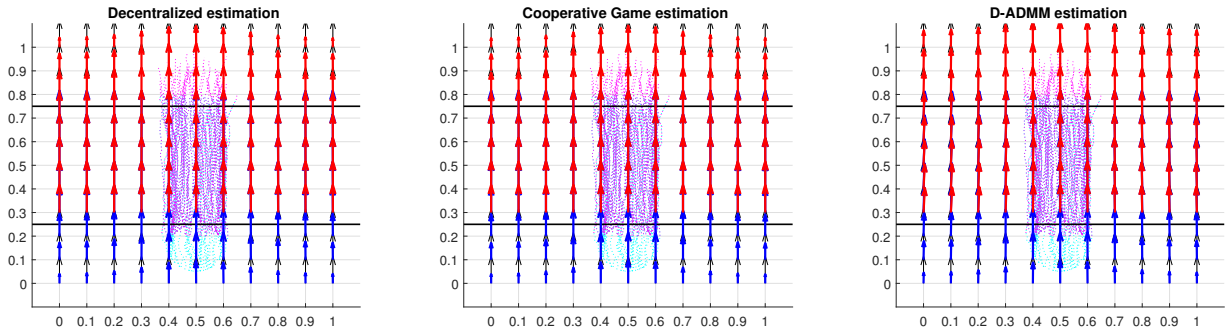


(b) $|o| = 3$ nodes

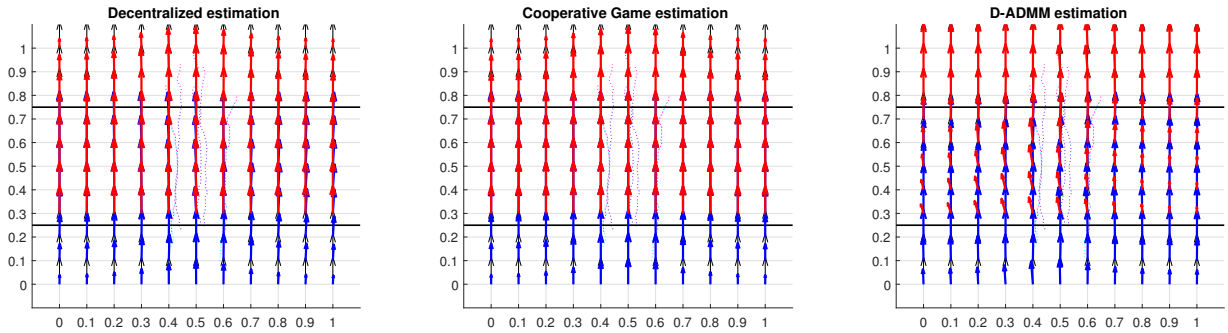


(c) $|o| = 1$ node

Figure 1.13.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 20$ trajectories. There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.



(a) $S = 80$ trajectories



(b) $S = 5$ trajectories

Figure 1.14.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using a medium width overlapping region of $|o| = 5$ nodes (between vertical black lines). There are two estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, and the other on the right, in red, which has access to the magenta portion of the trajectories.

Flow vector field

The top row of Figure 1.15 shows the VFE diagram to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Higher $|o|$ yields vector fields with more similar lengths (*i.e.*, normalized RMSL) for the Decentralized and D-ADMM approaches. The angle similarity between pairs of vectors is almost not affected by changes in $|o|$. The Cooperative game approach is the approach that yields the least consensual vector field estimates. The D-ADMM is the approach that is most affected by changes in $|o|$. However, this effect is mainly observed in the length of the estimated vectors, and not the relative angle. On the contrary, the lack of consensus between neighbour agent estimates of the Decentralized and Cooperative game approaches is due to differences in the relative angle between the estimated vectors. This effect is due to a transition between generating vector fields in the overlapping region, which is not easy to extrapolate between neighbour agents, given such short trajectory data (see Figure 1.16). Moreover, fewer available trajectories yield lower consensus between pairs of estimated vectors for low $|o|$. The D-ADMM approach is the most affected by changes in S (see Figure 1.17). Consensus assessment by the normalized $\mathrm{RMSVD_c}$ does not introduce new information when compared to assessment by the VFE diagram (see Figure 1.18).

The bottom row of Figure 1.15 shows the VFE diagram to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More accurate estimates yield larger normalized RMSL values and similarity closer to 1 (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors). Note that normalized RMSL values are very small (10^{-9}) because the estimated vector fields are sparse and thus have much smaller average length than the synthetic generator vector field, which is non-zero everywhere.

Lower $|o|$ yields less accurate vector field estimates with respect to vector length (see Figure 1.15). The angle similarity between pairs of vectors is almost not affected by changes in $|o|$. Regarding the Decentralized and the D-ADMM approaches, larger $|o|$ improves accuracy of the estimates vector fields overall. On the contrary, the accuracy of the estimates of the Cooperative game approach does not improve with changes in $|o|$, it even worsens for larger $|o|$ (see Figure 1.15). Moreover, the effect of S is small for all approaches (lower-right panel of Figure 1.18). Accuracy assessment by the normalized $\mathrm{RMSVD_c}$ does not introduce new information when compared to assessment by the VFE diagram (see Figure 1.18).

Therefore, when estimating the flow vector field, the D-ADMM approach yields similar accuracy performance when compared to the Decentralized approach, but higher consensus performance than the other two approaches, as expected. This example provided an opportunity to tune the cost function parameter values to induce sparsity of the estimates,

which hampered VFE diagram interpretation regarding the reference generating vector field. Nevertheless, this example provided a typical real case scenario with long trajectories generated by a type of vector field that changes its direction within the overlapping region between neighbour agents. Only the D-ADMM approach considers an equality constraint between neighbour agents estimates, thus favouring this approach when dealing with this kind of problems.

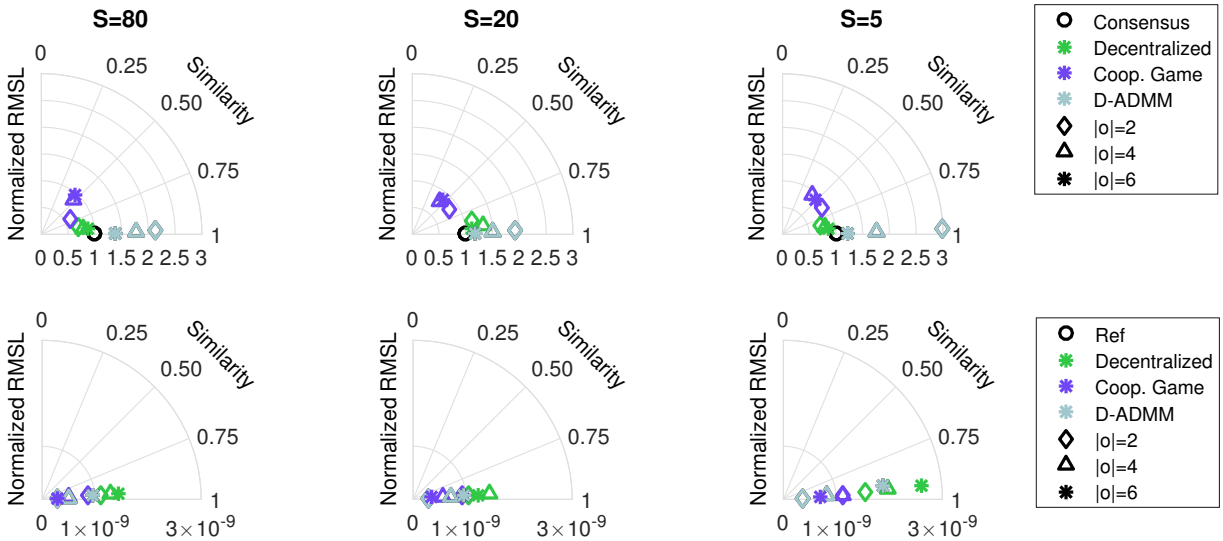
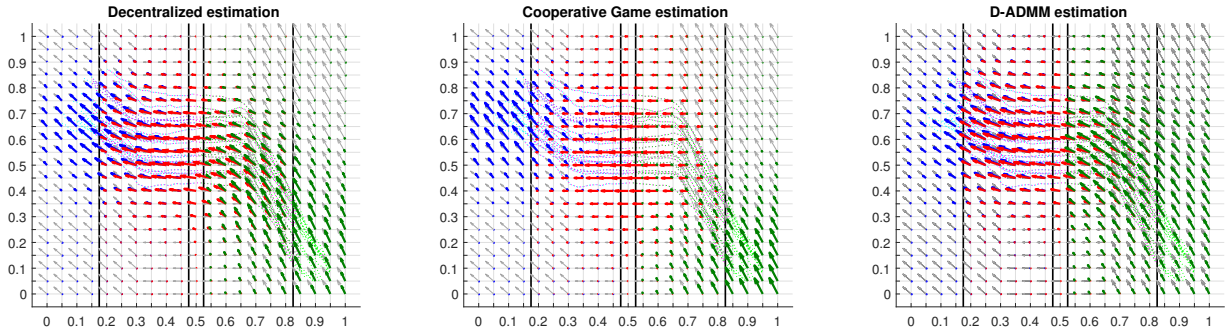
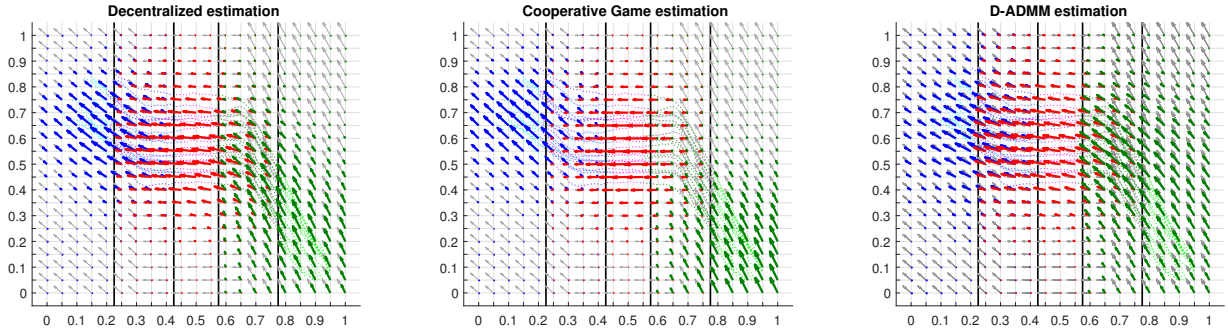


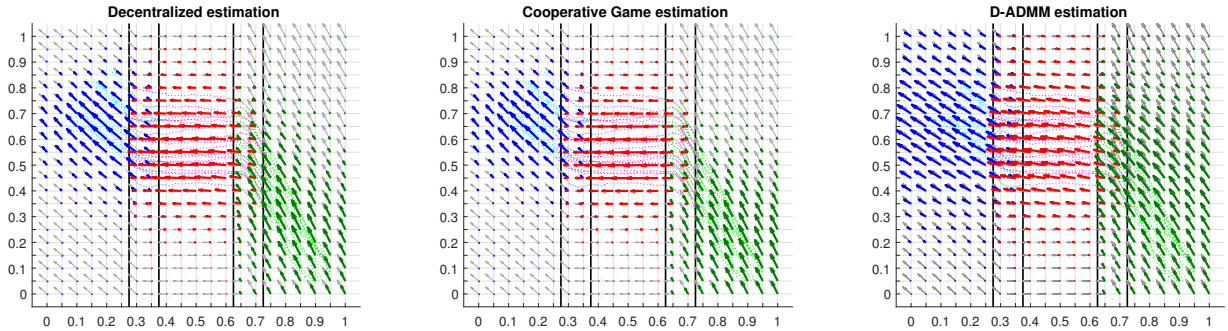
Figure 1.15.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The different symbols represent several widths of the overlapping region ($|o| = 2, 4, 6$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.



(a) $|o| = 6$ nodes

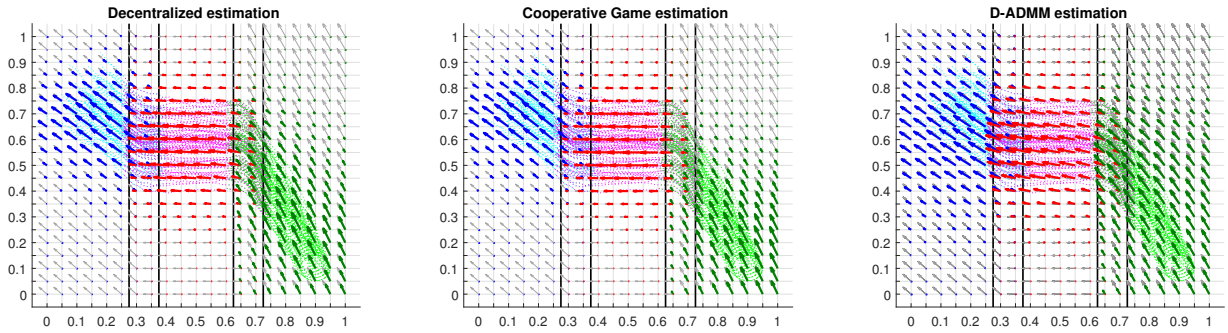


(b) $|o| = 4$ nodes

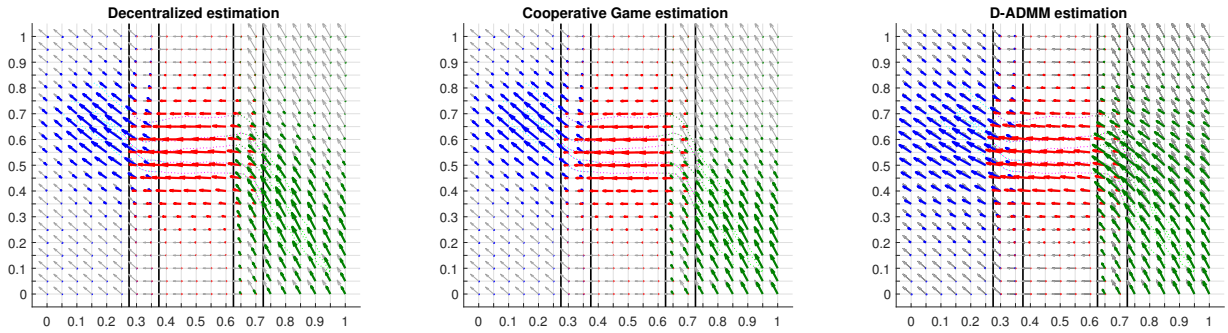


(c) $|o| = 2$ nodes

Figure 1.16.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 20$ trajectories. There are three estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, another on the center, in red, which has access to the magenta portion of the trajectories, and the third on the right, in green, which has access to the light green portion of the trajectories.



(a) $S = 80$ trajectories



(b) $S = 5$ trajectories

Figure 1.17.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using a narrow overlapping region of $|o| = 2$ nodes (between vertical black lines). There are three estimator agents: one on the left, in blue, which has access to the cyan portion of the trajectories, another on the center, in red, which has access to the magenta portion of the trajectories, and the third on the right, in green, which has access to the light green portion of the trajectories.

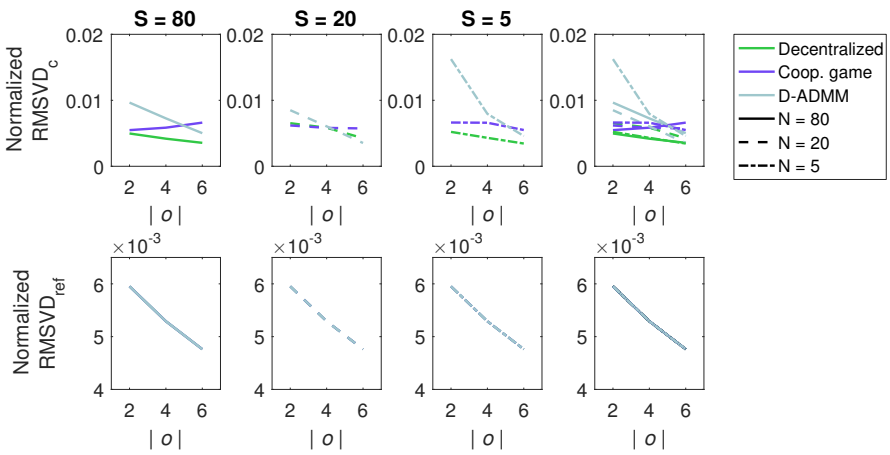


Figure 1.18.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. The different columns show results for different number of available trajectories ($S = 80, 20, 5$). The right-most column combines the different number of available trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 2, 4, 6$). Lower normalized RMSVD values represent better performance.

V-shaped upwards vector field

The top row of Figure 1.19 shows the VFE diagram (with the zoom in of the rectangle region on the left) to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Medium $|o|$ yields better matching vector fields overall (top zoom in panel of Figure 1.19). The relative angle between vector pairs is almost not affected by changes in $|o|$. The Cooperative game is the approach that yields better consensus between estimated vector pairs, *i.e.*, values closest to 1, for the different $|o|$ values (see Figure 1.21).

The bottom panel of Figure 1.19 shows the VFE diagram (with the zoom in of the rectangle region on the left) to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More accurate estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors).

Medium $|o|$ yields more accurate vector field estimates for all approaches. The D-ADMM approach is the least affected by changes in $|o|$, and the one that yields more accurate vector field estimates (see Figure 1.21).

For both consensus and accuracy, assessment by the normalized $\mathrm{RMSVD_c}$ and $\mathrm{RMSVD_{ref}}$ respectively are not as clear as through the VFE diagram because the differences between vector fields are almost reduced to vector length within a very small interval. Moreover, it was not necessary to study the effect of S because this example serves as ground truth for comparison with the real pedestrian trajectory data. Instead, S was fixed as 50, which is similar to the S of the real data (*i.e.*, $S = 47$).

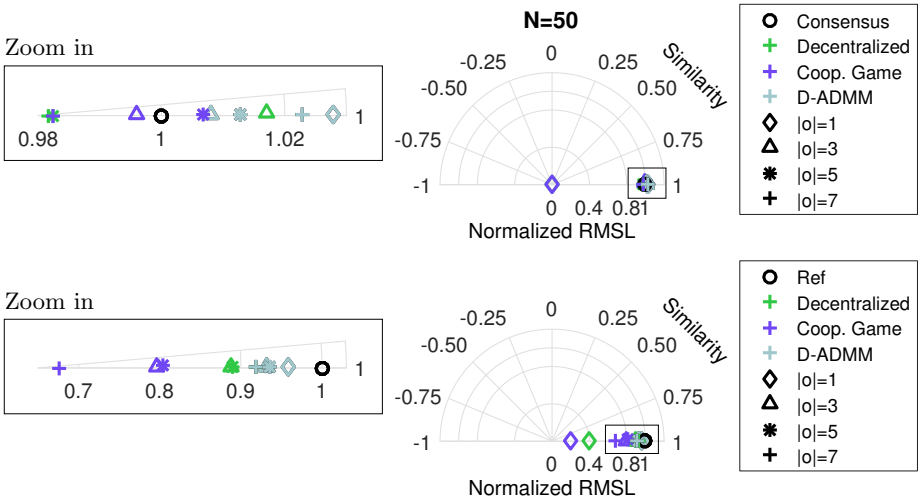


Figure 1.19.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different symbols represent several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.

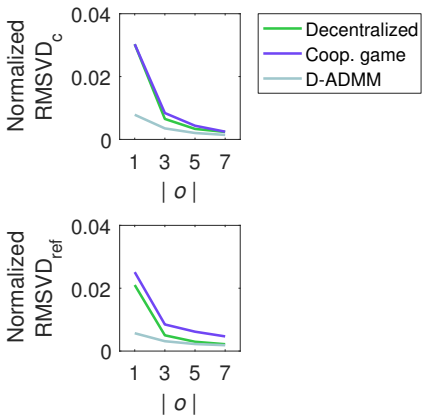
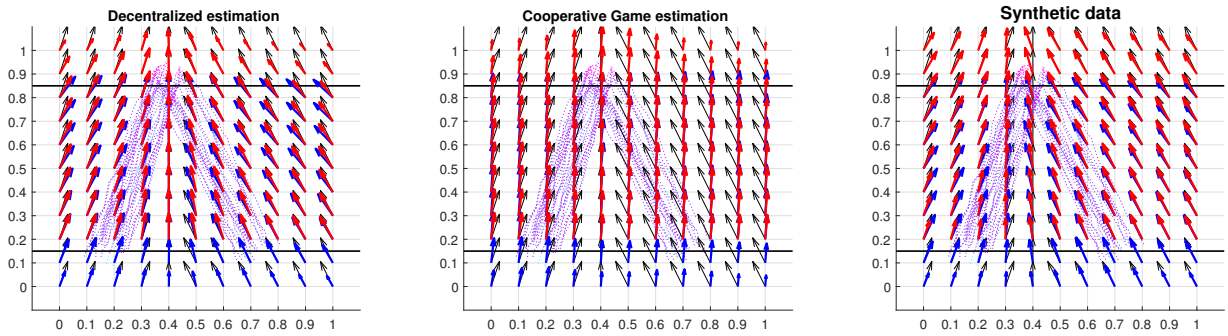
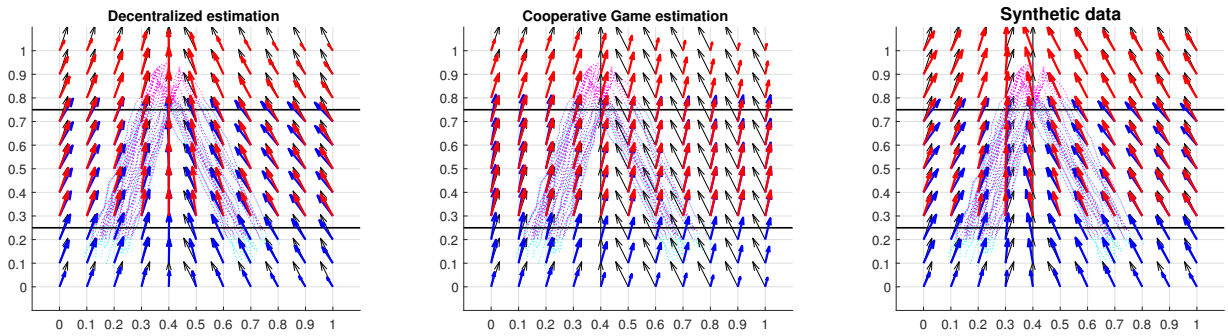


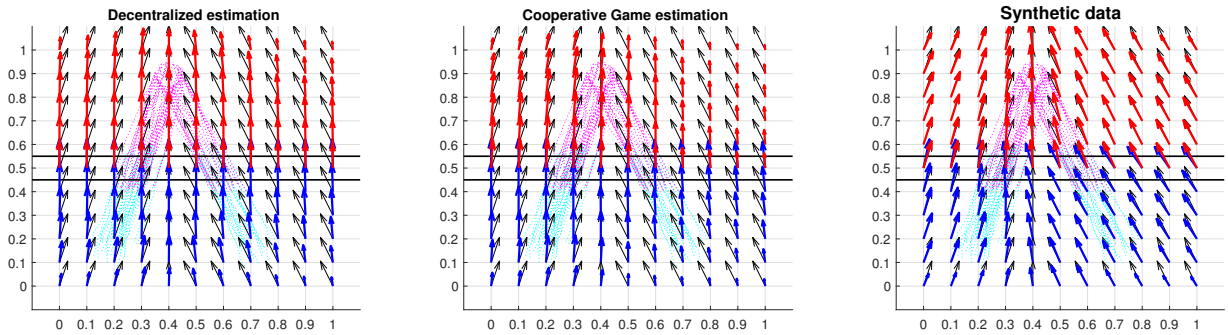
Figure 1.20.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. Results obtained with $S = 50$ trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Lower normalized RMSVD values represent better performance.



(a) $|o| = 7$ nodes



(b) $|o| = 5$ nodes



(c) $|o| = 1$ node

Figure 1.21.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 50$ trajectories. There are two estimator agents: one on the bottom, in blue, which has access to the cyan portion of the trajectories, and the other on the top, in red, which has access to the magenta portion of the trajectories.

1.4.2. Real Pedestrian Trajectory Data

The video signal was acquired using a Sony HDR-CX260 video camera with a resolution of 8.9 megapixels per frame and working at a frame rate of 30 frames per second. The pedestrians trajectories were extracted using a tracking algorithm. This procedure uses a background subtraction algorithm for the detection of active regions, followed by matching active regions at consecutive frames [23]. The trajectories were then sub-sampled at a frame rate of 1 frame per second, and the association errors were corrected. There were $S = 47$ available trajectories.

As before, vector field estimation depended on the width of the overlapping region ($|o| = 1, 3, 5, 7$), within an over-imposed grid of 11×11 nodes. We expected to use the results obtained using real pedestrian trajectory data as ground truth for comparison with the ones obtained in the estimation of the V-shaped upwards vector field using synthetic data.

V-shaped upwards vector field

The top row of Figure 1.22 shows the VFE diagram (with the zoom in of the rectangle region on the left) to assess consensus between neighbour agents estimated vector fields for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More consensual estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vector pairs).

Higher $|o|$ yields better matching vector fields overall (see Figure 1.23). The relative angle of the estimated vector pairs is almost not affected by changes in $|o|$. The D-ADMM approach is the least affected by changes in $|o|$. However, both the Cooperative game and the D-ADMM approaches yield the best consensus between estimated vector pairs, *i.e.*, values closest to 1, for $|o| = 7$ and $|o| = 5$, respectively.

The bottom panel of Figure 1.22 shows the VFE diagram (with the zoom in of the rectangle region on the left) to assess the accuracy of the estimated vector fields in reference to the generating vector field for each of the 3 proposed estimation approaches over different number of available trajectories, S, and widths of overlap, $|o|$. More accurate estimates yield values close to the black circle (*i.e.*, same length and maximum angle similarity between the estimated vectors and the generating field vectors).

Medium to high $|o|$ yields more accurate vector field estimates for all approaches. Accuracy of the estimated fields is lowest for the Cooperative game approach, followed by the Decentralized and the D-ADMM approaches. This effect is observed both in relative angle, in the case of the Cooperative game approach, and in the length of the estimated vectors, comparing the Decentralized with the D-ADMM approach, for medium to higher $|o|$ (see Figure 1.24). There is almost no effect of $|o|$ in the accuracy of the vector field estimates of the D-ADMM approach.

For both consensus and accuracy, assessment by the normalized $\mathrm{RMSVD_c}$ and $\mathrm{RMSVD_{ref}}$ respectively are not as clear as through the VFE diagram. This happens because the differences between vector fields are either reduced to vector length within a very small

interval, in the case of consensus, or shared between angle and length differences between vector pairs, in the case of accuracy.

In comparison with the V-shape upwards vector field estimation using synthetic data, these results yield similar trends in consensus and accuracy assessment. Regarding the consensus assessment, the effect of $|o|$ is larger for the estimated vector length of the real data example than in the synthetic data example, especially for the Decentralized and the Cooperative game approaches. Regarding the accuracy assessment, the effect of $|o|$ is larger for both the relative angle and vector length of the real data example than of the synthetic date example. This difference can be due to using low σ^2 for generating synthetic data, when compared to the real data.

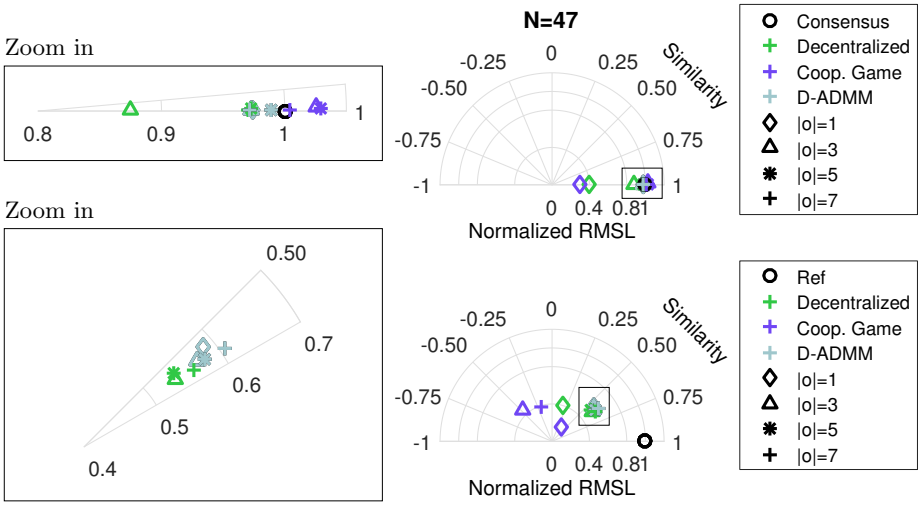


Figure 1.22.: Vector field diagrams to assess consensus between estimated vector fields of neighbouring agents (top row) and estimation accuracy with reference to the generating vector field (bottom row) for the three proposed approaches. The different symbols represent several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Values closer to the black circle (*i.e.*, the estimated vector field from agent 1 for consensus assessment, and the generating vector field for accuracy assessment) represent better performance.

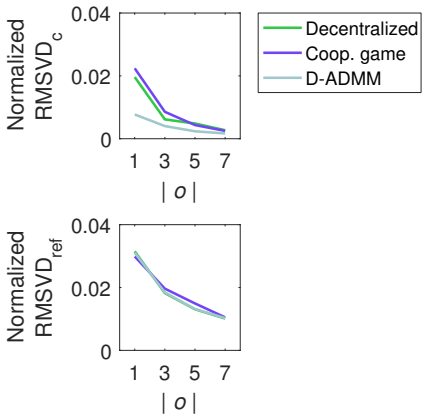
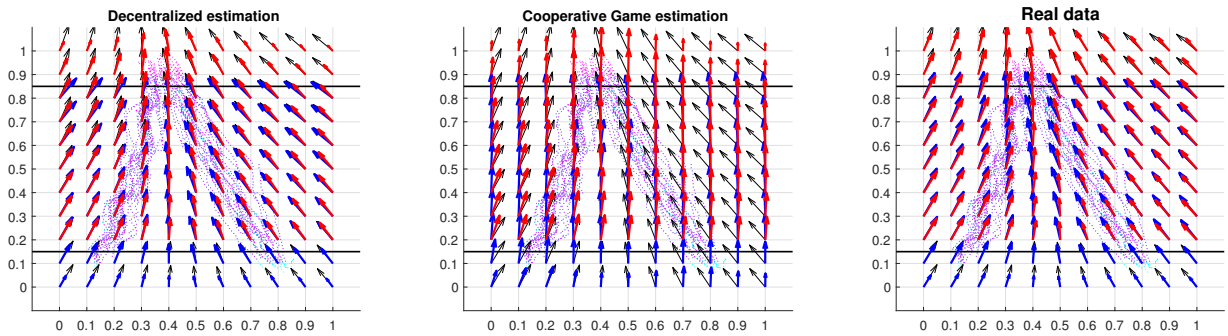
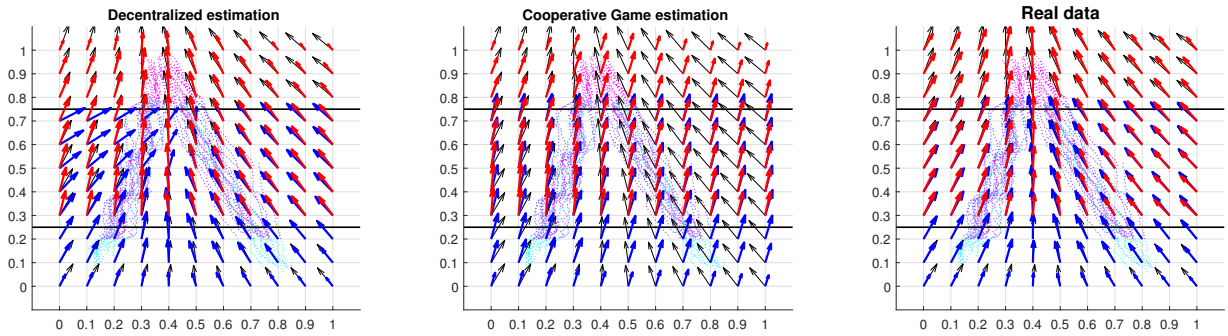


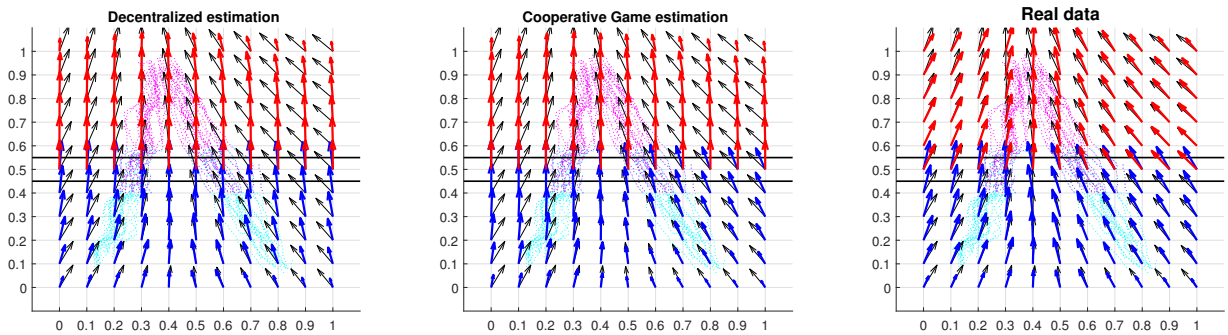
Figure 1.23.: Normalized root mean square vector difference (RMSVD) to assess overall difference between the two estimated vector fields of neighbouring agents (RMSVD_c – top row) and the estimated vector fields and the generating vector field ($\text{RMSVD}_{\text{ref}}$ – bottom row) for the three proposed approaches. Results obtained with $S = 47$ trajectories. The horizontal axis represents the several widths of the overlapping region ($|o| = 1, 3, 5, 7$). Lower normalized RMSVD values represent better performance.



(a) $|o| = 7$ nodes



(b) $|o| = 5$ nodes



(c) $|o| = 1$ node

Figure 1.24.: Vector field estimates for Decentralized estimation, distributed estimation based on a Cooperative game, and Distributed ADMM estimation using $S = 47$ trajectories. There are two estimator agents: one on the bottom, in blue, which has access to the cyan portion of the trajectories, and the other on the top, in red, which has access to the magenta portion of the trajectories.

1.5. Conclusions

This section proposes three distributed approaches to estimate vector fields from pedestrian trajectories in an open scene. The vector fields are assumed to rule the pedestrian trajectories similarly to what was described before [2, 3]. This work focuses on distributed estimation approaches that apply to the surveillance of very large spaces or scenes including alleys or corners, in which cases the images of a single camera are insufficient to cover the whole area of interest.

The first approach was a purely decentralized approach, where each agent estimates its own vector field with no knowledge of the neighbour agents estimates. The second approach was a distributed estimation approach based on a Cooperative game, where each agent estimates its own vector field with the objective of the collective best. In this scenario, the agents estimates do not necessarily converge to the optimum. The third approach was a distributed version of the ADMM algorithm, where each agent estimates its own vector field subject to equality constraints on the overlapping region. In this scenario, the goals are to achieve both accurate estimates and consensus between the neighbour agents estimates in the overlapping regions.

The experiments using synthetic target trajectory data show that the D-ADMM approach is the most robust and reliable approach to estimate generating vector fields with changes in direction within the overlapping regions of neighbour agents because only the D-ADMM approach considers a flexible equality constraint among neighbour agents estimates. Contrarily, the Cooperative game approach is very susceptible to changes in motion direction in the overlapping region of neighbour agents.

Regarding the number of available trajectories, the experiments show that fewer available trajectories hamper vector field estimation, as expected. Regarding the overlapping region, the D-ADMM approach is the most robust to different overlapping widths. Finally, the results on the V-shaped upwards vector field estimation showed that, from the proposed approaches, the D-ADMM yields more accurate and consensual estimates.

2. Centralized Estimation of Multiple Vector Fields

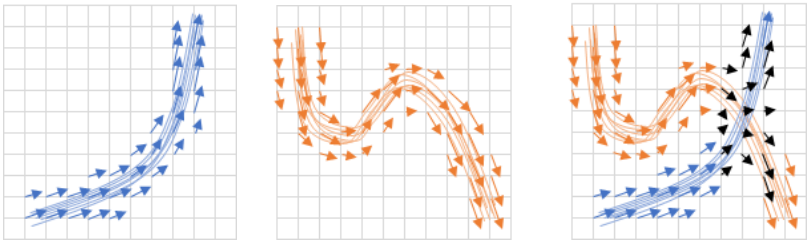
Far-field activities represented as time series or trajectories can be summarized in compact representations of frequent patterns. Popular representations such as clustering or probabilistic modelling of trajectories often do not inform about both velocity and direction of motion, which are by definition visually and quantitatively embedded in vector fields. However, a common use of vector fields may dismiss information about forbidden areas, or regions with concurrent activity patterns. To address this problem we present a non-iterative layered vector field estimation process that yields sparse vector field abstractions of activity patterns from groups of trajectories. The key feature of our approach is the estimate of the probability density function (PDF) of targets positions: it automatically tunes the cost function parameter, and serves as weights in the sparse estimation problem. Experiments in synthetic trajectory data show that the proposed estimation approach yields correctly sparse vector fields which are similar to known generating vector fields in both velocity and direction. We also propose a trajectory labeling algorithm that labels trajectories according to their activity patterns based on the vector field abstractions. Experiments on three real data sets yield higher labeling accuracy of test trajectories when compared to other generative models. The proposed approach abstracts frequently observed activity patterns in a scene and informs about their physical and semantic meaning. Outlier trajectories are also detected.[1]

2.1. Introduction

Far-field activities can be described by trajectories [24, 25], their spatial and angular features [26, 27], or by generative models [4, 28, 29, 30]. A high-level description of the spatial distribution of frequent activity patterns can support anomaly detection [2] and accessibility planning [5], and encode semantic regions of the scene [31].

In the literature, both diffusion and probabilistic models have been used to describe frequent activity patterns and to label trajectories according to activity patterns. On the one hand, *diffusion* models such as optical flow models have been used to describe coherent motions and semantic regions [32], and heat maps based on thermal diffusion processes have been used to capture the temporal motion information of activities [33]. On the other hand, *probabilistic* models such as Hidden Markov Models have been used to detect the points of interest in a scene [30], and Dirichlet Process Mixture Models have been used to robustly label new trajectories according to their activity patterns [29]. These studies either focused on attributing a single semantic meaning to specific spatial regions [32, 30],

[1] Parts of this chapter were submitted to IbPRIA'19.



(a) (b)

Figure 2.1.: Example of the usefulness of multiple vector fields. 2.1(a) Multiple vector fields considered independently allow for correct motion description of the different motion patterns of the two sets of trajectories. 2.1(b) A single vector field describing two sets of trajectories with very different motion patterns needs to consider them simultaneously and the true motion description is concealed (black arrows).

or on labelling trajectories according to their activity patterns [33, 29]. However, neither of them provide straightforward information about velocity and direction of the activity patterns, and only the latter are able to describe different global activity patterns in the same region of the scene. Typically, this type of motion information is associated to vector fields which embed both the physical meaning of motion (*i.e.*, velocity and direction) and the semantic interpretation of different regions of the scene [28, 5].

We propose to (i) describe multiple, complex activity patterns using layered vector fields (see Figure 2.1) and (ii) label test trajectories according to activity patterns using the estimated vector fields. Our approach extends that of previous studies in that: (a) it imposes data-driven sparsity to the vector field abstractions to prevent erroneous extrapolations in regions with no target data (contrarily to [5, 28]); (b) it is sensitive to concurrent activity patterns (contrarily to [32, 30]); and (c) it provides information on the velocity and direction of activity patterns (contrarily to [33, 29]).

The proposed vector field estimation uses a cost function specifically designed to yield sparse estimates. Contrarily to other studies, which induce sparsity of the vector field estimates through the l_1-norm [34], this work does so through statistical conditioning on available data – the estimated spatial Probability Density Function (PDF) of the targets positions restricts vector field estimates to the regions where targets are observed. The proposed cost function further benefits from automatic parameter tuning using targets positions and trajectory features. Layered vector field abstractions can be obtained if the proposed approach is applied on pre-clustered trajectories with similar activity patterns. We assess the accuracy of the vector field abstractions of synthetic trajectories by comparing the estimated and generating vector fields, and the correct sparsity by comparing the estimated vector fields in regions with no target data with the null vector field. Vector field comparisons focus on the mean vector length (RMSL) and the vector similarity coefficient (R) [22].

Moreover, we propose a trajectory labeling algorithm according to activity patterns. The displacement error between test trajectories and generated trajectories using the

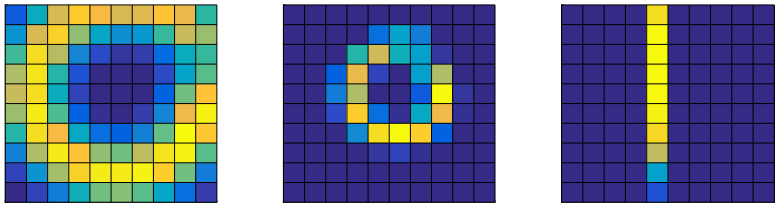


Figure 2.2.: Probability density functions of the targets positions for 3 different activity patterns from the synthetic data set (D_0). Gradient colours of ascending density of data points from dark blue to yellow.

estimated vector fields is the measure for classification. This way, test trajectories are sorted according to activity patterns or detected as outliers. We asses the accuracy of the trajectory labeling algorithm by comparing the attributed and the observed activity pattern labels of test trajectories.

2.2. Estimation of Multiple Vector Fields

We first aim to estimate the vector field, $\boldsymbol{T}$, that describes the activity patterns of a set of S trajectories, $\mathcal{X} = \{\boldsymbol{x}_1(t), \ldots, \boldsymbol{x}_S(t)\}$. Let $t = 1, \ldots, L_s$ and the target position, $\boldsymbol{x}_s(t)$, in the image plane of a camera be *driven* by $\boldsymbol{T}$ according to

$$\boldsymbol{x}_s(t) = \boldsymbol{x}_s(t-1) + \boldsymbol{T}(\boldsymbol{x}_s(t-1)) + \boldsymbol{w}_s(t)\,, \tag{2.1}$$

where $\boldsymbol{w}_s(t) \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\,\mathbf{I}), \forall\, t$, is a white random perturbation. Let the image plane be normalized, thus $\boldsymbol{x}_s(t) \in [0,1]^2\,\forall\, t$.

The vector field, $\boldsymbol{T} : [0,1]^2 \to \mathbb{R}^2$, is defined only at the grid nodes of an over-imposed regular, uniform grid, $\mathcal{G} = \{\boldsymbol{g}_n \in [0,1]^2, n = 1, 2, \ldots, N\}$, on the image plane. As the target trajectories can be defined in any image coordinate, even if it does not correspond to a grid node ($\boldsymbol{x}_s(t) \notin \mathcal{G}$), we bilinearly interpolate to represent the vector field that drives the target position on any coordinate of the image plane, $\boldsymbol{x}_s(t) \notin \mathcal{G}$:

$$\boldsymbol{T}(\boldsymbol{x}_s(t)) = \sum_{n=1}^{N} \boldsymbol{\phi}_n(\boldsymbol{x}_s(t))\,\boldsymbol{t}_n\,, \tag{2.2}$$

where $\boldsymbol{\phi}_n(\boldsymbol{x}_s(t))$ are the interpolation coefficients of the velocity vectors, $\boldsymbol{t}_n$, at the grid nodes. The matrix of interpolation coefficients for set $\mathcal{X}$ is $\boldsymbol{\Phi}$.

Vector field estimation corresponds to an optimization problem where $\boldsymbol{T}$ is the minimizer of a given cost function that has to induce data-driven sparsity. To impose *sparsity* of the vector field estimates in the regions where target data does not exist, the velocity vectors in $\boldsymbol{T}$ are weighted by 1 minus the spatial probability density function (PDF) of targets positions, *i.e.*, $\mathcal{D} = \mathbb{1} - \Gamma_p$, $\Gamma_p \in \mathbb{R}^N$. Γ_p is the estimated PDF of the targets positions using the Parzen window algorithm over set $\mathcal{X}$. Then, to get its value at the grid nodes, we discretize at the desired image coordinates (Figure 2.2).

The cost function is therefore defined as

$$f(\boldsymbol{T}) = \|\boldsymbol{V} - \boldsymbol{T}\,\boldsymbol{\Phi}\|_2^2 + \alpha\,\|\boldsymbol{T} \circ \mathbb{1}\mathcal{D}^\top\|_2^2\,, \tag{2.3}$$

where $\mathbb{1}$ is of size $[2\times 1]$, $\|\,.\,\|_2$ defines the l_2-norm of a vector, "$\circ$" represents the Hadamard product, and $\boldsymbol{T} \in \mathbb{R}^{2\times N}$, $\boldsymbol{V} \in \mathbb{R}^{2\times M}$, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times M}$, $M = \sum_{s=1}^{S}(L_s - 1)$, are given by

$$\boldsymbol{T} = \begin{bmatrix}\boldsymbol{t}_1 & \dots & \boldsymbol{t}_N\end{bmatrix}, \tag{2.4}$$

$$\boldsymbol{V} = \begin{bmatrix}\boldsymbol{v}_1(2)\dots\boldsymbol{v}_1(L_1) & | & \dots & | & \boldsymbol{v}_S(2)\dots\boldsymbol{v}_S(L_S)\end{bmatrix}, \tag{2.5}$$

$$\boldsymbol{\Phi} = \left[\begin{array}{c|c|c} \boldsymbol{\phi}_1(\boldsymbol{x}_1(1))\dots\boldsymbol{\phi}_1(\boldsymbol{x}_1(L_1-1)) & & \boldsymbol{\phi}_1(\boldsymbol{x}_S(1))\dots\boldsymbol{\phi}_1(\boldsymbol{x}_S(L_S-1)) \\ \vdots & \dots & \vdots \\ \boldsymbol{\phi}_N(\boldsymbol{x}_1(1))\dots\boldsymbol{\phi}_N(\boldsymbol{x}_1(L_1-1)) & & \boldsymbol{\phi}_N(\boldsymbol{x}_S(1))\dots\boldsymbol{\phi}_N(\boldsymbol{x}_S(L_S-1)) \end{array}\right], \tag{2.6}$$

with matrix $\boldsymbol{V}$ composed of the velocity vectors between consecutive target positions, *i.e.*, $\boldsymbol{v}_s(t) = \boldsymbol{x}_s(t) - \boldsymbol{x}_s(t-1)$.

In (2.3), α is correlated with the grid resolution N. To avoid manual parameter input in every estimation procedure, we propose an *automatic tuning* based on the cardinality of the non-zero elements (*i.e.*, $|\cdot|_{\neq 0}$), expected value (*i.e.*, $\mathbb{E}[\,\cdot\,]$), and standard deviation [*i.e.*, $\sigma(\,\cdot\,)$] of the estimated PDFs of target and trajectory features,

$$\alpha = 1 - \frac{|\Gamma_p|_{\neq 0}}{N}\,, \tag{2.7}$$

$$N = \max\left\{N_{\min}, |\Gamma_c|_{\neq 0} > \mathbb{E}[\Gamma_c] + 1.5\,\sigma(\Gamma_c)\right\}, \tag{2.8}$$

where Γ_p is the spatial PDF of the target positions as before; $\Gamma_c \in \mathbb{R}^N$ is the average curvature of the trajectories at the grid nodes, which is estimated using the velocity angles $\theta(t) = \tan^{-1}\left(\frac{y(t)-y(t-1)}{x(t)-x(t-1)}\right)$ [2, 27]; and $N_{\min}$ is the minimum grid resolution selected by the user. In (2.8), very curly trajectories (*i.e.*, extreme values of the distribution of Γ_c) define the grid resolution.

Multiple vector fields can be estimated using (2.3) if it is applied to each set of pre-clustered trajectories ($\mathcal{X}_k$) with similar activity patterns, *e.g.* using multiple features [27]. In the following, we assume that the pre-clustering step has taken place and that we have access to the sets $\mathcal{X}_k$.

2.3. Activity Pattern Labeling

Our second aim is to label trajectories according to their activity patterns. To achieve this aim, we first estimate the $\boldsymbol{T}_k$ following the above approach and using only trajectories from training sets $\mathcal{X}_k$. Then, we propose the following labeling algorithm:

1. Trajectory labeling:
   a) Generate trajectories from the starting point of a given test trajectory using the estimated $\boldsymbol{T}_k$ and (2.1);

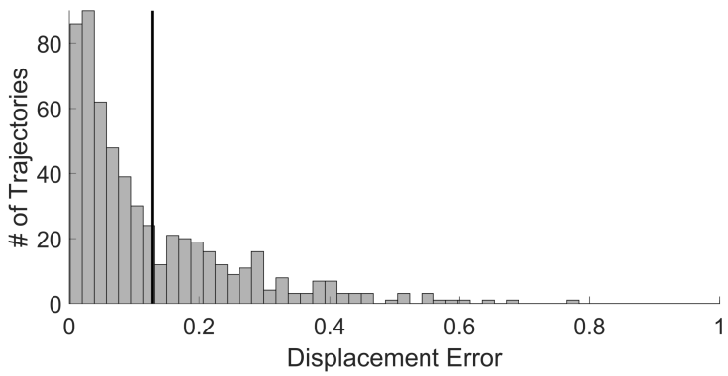


Figure 2.3.: Histogram of displacement errors between validation trajectories and generated trajectories using the known labels of activity patterns (vector fields). The cutoff threshold for outlier detection is the sum of the median and median absolute deviation of the displacement errors. Note that we use the median and its absolute deviation instead of the mean and standard deviation given that the distribution of displacement errors is right skewed. The validation trajectories used in this example come from the real data set D_3.

b) Compute the displacement error as the euclidean distance between the generated and the test trajectories;

c) Label each test trajectory with the activity pattern (vector field abstraction) that yields the smallest displacement error.

2. Outlier detection using threshold:

   a) Compute the cutoff threshold as the sum of the median and the median absolute deviation (MAD) of the displacement errors obtained from steps 1.(a) and 1.(b) applied on a set of validation trajectories (Figure 2.3);

   b) Label test trajectories as outliers of the labeled activity pattern from step 1.(c) if their displacement error is above the threshold.

2.4. Experimental Results

2.4.1. Synthetic data

Assessment measures.

Estimates of vector fields using synthetic data ($\boldsymbol{T}_{\text{est}}$) are assessed regarding both the *accuracy* when compared to the known generating vector field ($\boldsymbol{T}_{\text{ref}}$) and the correct *sparsity* compared to the null vector field ($\boldsymbol{T}_0$) in regions where no target data is observed. Let each node on the over-imposed grid be labeled according to its proximity to a given trajectory as an *active node*, if it belongs to a square of nodes containing part of a given trajectory, or a *non-active node*, if it does not belong to such a square of nodes (see

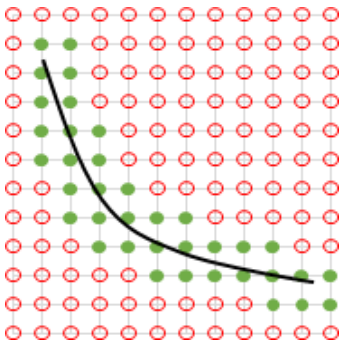


Figure 2.4.: Illustration of non-active nodes (red) *vs.* active nodes (green) around observed target positions (*i.e.*, trajectory; black).

Figure 2.4). Thus, the region where no target data is observed is defined as the set of non-active nodes in the image plane, $\mathcal{Z}$, with respect to a given trajectory set.

The assessment measures compare pairs of vectors regarding the vector similarity coefficient (R), *i.e.*, the mean of the inner product of normalized vector pairs from 2 vector fields A and B, defined as [22],

$$R = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \hat{\boldsymbol{t}}_i^A \cdot \hat{\boldsymbol{t}}_i^B \,; \tag{2.9}$$

and the vector root mean square length (RMSL), *i.e.*, the systematic difference in the mean vector length, defined as [22],

$$\text{RMSL} = L_V^2 = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left\| \boldsymbol{t}_i^A - \boldsymbol{t}_i^B \right\|_2^2 ; \tag{2.10}$$

where "$\cdot$" is the inner product, $\hat{\boldsymbol{t}} = \frac{\boldsymbol{t}}{\sqrt{\|\boldsymbol{t}\|_2^2}}$, $\| \,.\, \|_2$ represents the l_2-norm of a vector, and $|\,.\,|$ represents the cardinality of a set. In the case of *accuracy* assessment, $A = \boldsymbol{T}_{\text{ref}}$, $B = \boldsymbol{T}_{\text{est}}$, and $\mathcal{P} = \mathcal{G}$, the set of grid nodes. In the case of correct *sparsity* assessment, $A = \boldsymbol{T}_0$, $B = \boldsymbol{T}_{\text{est}}$, and $\mathcal{P} = \mathcal{Z}$, the set of non-active nodes. The optimal values for these measures are (R, RMSL)$= (1, 1)$ and RMSL$= 0$, respectively for accuracy and sparsity assessments.

Data set.

The synthetic data set (D_0) has 300 trajectories generated using 6 different activity patterns. We use D_0 as a proof of concept for the assessment of accuracy and correct sparsity of the estimated vector fields.

Results.

Figure 2.5 shows that the vector field estimates are very similar to the generating vector fields not only in terms of magnitude (RMSL) and direction (R) but also regarding

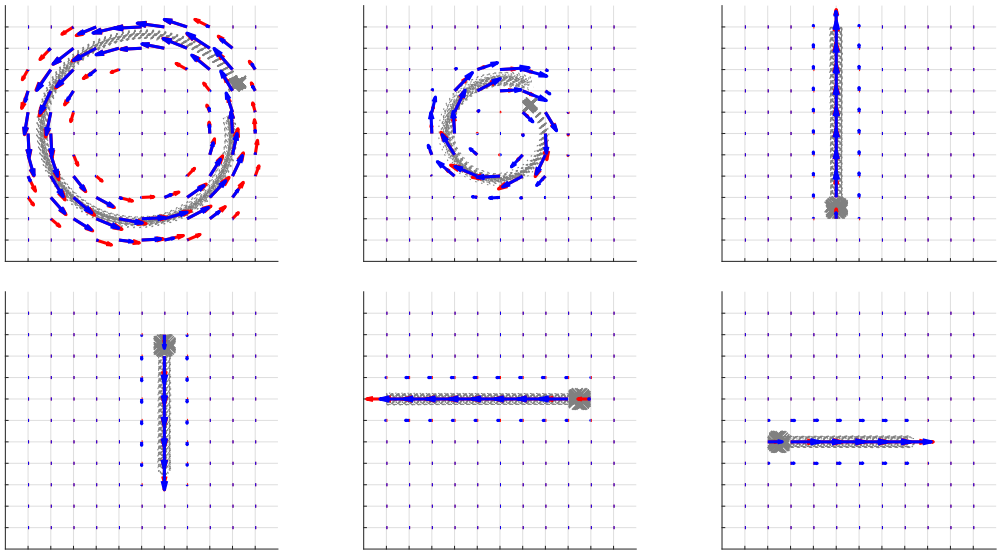


Figure 2.5.: Vector field estimates for the 6 activity patterns of the synthetic data set (D_0). Activity patterns are ordered across rows from left to right (top: 1, 2, 3; bottom: 4, 5, 6). Generating vector fields are shown in red, estimated vector fields in blue, generated trajectories in gray.

sparsity in the regions with no data, as expected. The accuracy of the estimated vector fields (activity patterns) is respectively (R, RMSL) **1:** (0.920, 0.663); **2:** (0.952, 2.305); **3:** (0.982, 0.979); **4:** (0.973, 0.723); **5:** (0.954, 9.787); **6:** (0.968, 0.763), and the RMSL of the vector field estimates corresponds to sparse vector fields, *i.e.*, all bellow 2.94e-04.

2.4.2. Real data

Assessment measures.

Activity pattern labeling accuracy is computed by comparing attributed and known trajectory labels taking into account the cutoff threshold as

$$Acc = \frac{\sum \mathrm{diag}(\mathcal{M})}{\sum_{ij} \mathcal{M}_{ij}}, \tag{2.11}$$

where $\mathcal{M}$ is the confusion matrix in a problem with multiple activity patterns.

Data set.

The real data sets we used are: D_1 (Hu), containing 1500 trajectories with 15 activity patterns [29]; D_2 (Wang), containing 220 trajectories with 11 activity patterns [32]; D_3 (Morris), containing 1900 trajectories with 19 activity patterns [30]. We use these data sets to assess activity pattern labeling and outlier detection.

Table 2.1.: Overall accuracy of trajectory labeling for the proposed approach and comparison with literature results.

Data Set	activity patterns	traj. per pattern	Overall Accuracy (%) Ours	Literature
D_0	6	50	**100.00**	–
D_1	15	18–278	**87.81**	[82.00, 86.70]
D_2	11	20	**95.45**	[86.30, 91.80]
D_3	19	100	**99.47**	[95.00, 98.00]

Results.

Table 2.1 shows that overall the proposed algorithm correctly labels trajectories according to their activity patterns with an accuracy above that described in the literature. More specifically with higher accuracy than Heat-map, HMM, and DPMM, which are comparable generative models used to describe activity patterns [29, 32, 30, 35].

Regarding outlier detection, note that the proposed algorithm always assigns an activity pattern to a trajectory – the attributed activity pattern is the one that generates trajectories with the smallest displacement error relative to the test trajectory. However, if the displacement error is above the threshold the respective trajectory is plotted in a different colour than the others and tagged as an outlier. Figure 2.6 shows examples of 2 activity patterns for each real data set, which have similar motion patterns but different semantic meanings. The proposed approach yields vector field abstractions that can distinguish between similar activity patterns with different underlying semantics, given that for each data set, trajectories which were wrongly labelled as having one activity pattern were correctly detected as outliers of that activity pattern. For example, the green outlier trajectories from D_1 (Figure 2.6 bottom left panel) are detected as outliers of that activity pattern – whereas the vector field of interest describes a left turn into the primary road, the outlier trajectories correspond to targets that instead performed a left turn into a secondary road. Similar examples for the other two data sets are shown in the bottom row of Figure 2.6. Thus, the proposed approach correctly labels new trajectories according to their activity patterns and semantic meanings.

2.5. Conclusion

We proposed a vector field estimation approach that copes with dense trajectory data and yields compact abstractions of frequent activity patterns. The proposed approach embeds data-driven sparsity, whilst providing physical and semantic meaning of observed activity patterns. Finally, the estimated vector fields can be used to label new trajectories and detect outliers according to their activity pattern.

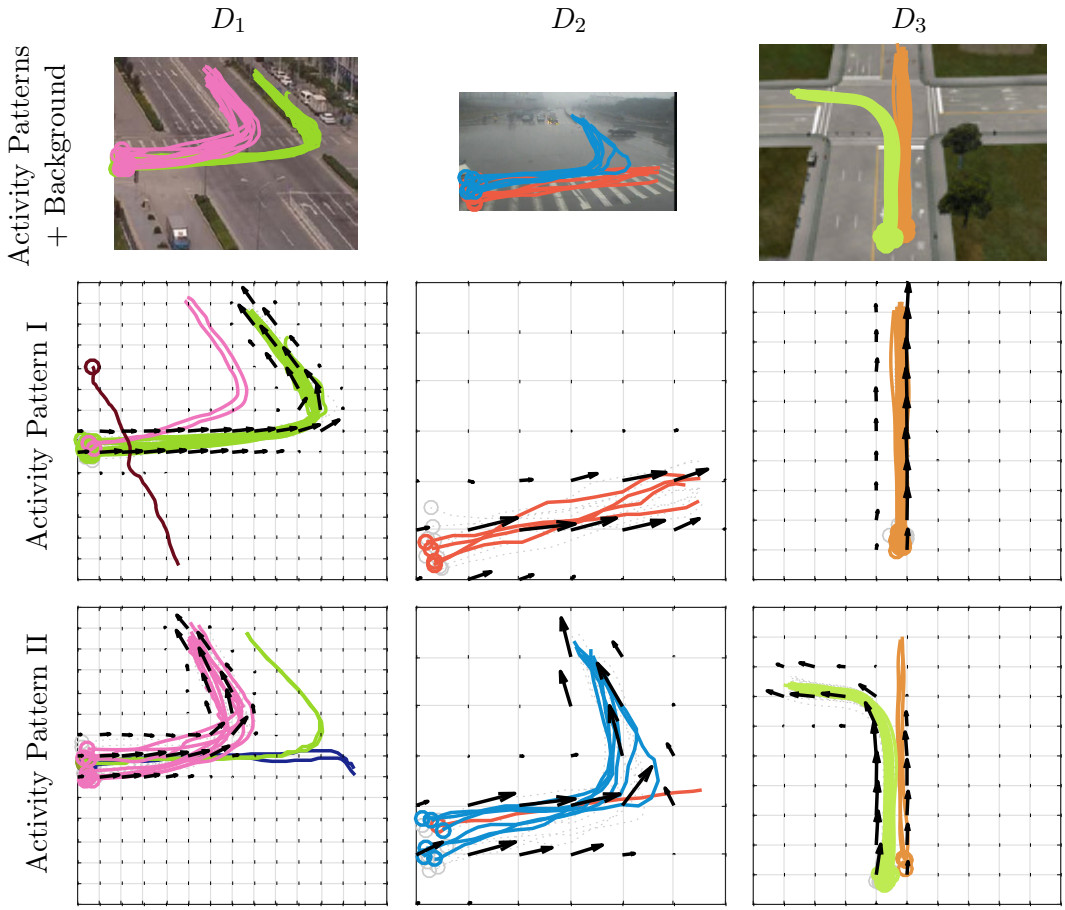


Figure 2.6.: Examples of trajectory labeling using the real data sets (D_1, D_2, and D_3), only two example activity patterns are shown for each data set. *Top row.* Overview of background image and test trajectories from two activity patterns. *Middle row.* Correctly labelled trajectories and over-imposed estimated vector fields (black arrows) of Activity Pattern I for each data set. Only D_1 has outlier trajectories from two other activity patterns (shown in different colors). *Bottom row.* Correctly labelled trajectories, over-imposed estimated vector fields (black arrows), and outliers (shown in different colors) of Activity Pattern II for each data set. All the data sets have outlier trajectories from Activity Pattern I shown in the row above. D_1 also has outlier trajectories form one additional activity pattern.

Bibliography

[1] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "Distributed Optimization with Local Domains: Applications in MPC and Network Flows," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, 2015.

[2] J. S. Marques and M. A. T. Figueiredo, "Fast Estimation of Multiple Vector Fields: Application to Video Surveillance," in *7th Int. Symposium on Image and Signal Processing and Analysis (ISPA)*, 2011.

[3] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, "Trajectory analysis in natural images using mixtures of vector fields," in *ICIP*, 2009, pp. 4353–4356.

[4] M. Barão and J. S. Marques, "Gaussian Random Vector Fields in Trajectory Modelling," in *Proc. of the 19th Irish Machine Vision and Image Processing Conf. (IMVIP)*, 2017, pp. 211–216.

[5] N. Ferreira, J. T. Klosowski, C. E. Scheidegger, and C. T. Silva, "Vector Field k-Means: Clustering Trajectories by Fitting Multiple Vector Fields," in *Eurographics Conf. on Visualization (EuroVis)*, vol. 32, 2013.

[6] H. Yao, A. Cavallaro, T. Bouwmans, and Z. Zhang, "Guest Editorial Introduction to the Special Issue on Group and Crowd Behavior Analysis for Intelligent Multicamera Video Surveillance," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 27, no. 3, 2017.

[7] J. Konrad and E. Dubois, "Bayesian Estimation of Motion Vector Fields," *IEEE Transitions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, 1992.

[8] C. Stiller, "Object-based estimation of dense motion fields," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 234–250, 1997.

[9] J.-W. Kampoon, W. Okolo, S. A. Erturk, O. Daskiran, and A. Dogan, "Wind Field Estimation and Its Utilization in Trajectory Prediction," in *AIAA Atmospheric Flight Mechanics Conference*, 2015.

[10] M. J. Tribou, A. Harmat, D. W. Wang, I. Sharf, and S. L. Waslander, "Multi-camera parallel tracking and mapping with non-overlapping fields of view," *International Journal of Robotics Research*, pp. 1–21, 2015.

[11] Y. Su, M.-T. Sun, and V. Hsu, "Global Motion Estimation From Coarsely Sampled Motion Vector Field and the Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 2, 2005.

[12] L.-C. Liu, J.-C. Chien, H. Y.-H. Chuang, and C. C. Li, "A Frame-Level FSBM Motion Estimation Architecture with Large Search Range," in *Proceedings of the*

IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03), 2003.

[13] P. Vijaykumar, A. Kumar, and S. Bhatia, "Latest Trends, Applications and Innovations in Motion Estimation Research," *International Journal of Scientific & Engineering Research*, vol. 2, no. 7, 2011.

[14] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606–1624, 2008.

[15] M. A. Patricio, J. Carbó, O. Pérez, J. García, and J. M. Molina, "Multi-agent framework in visual sensor networks," *Eurasip Journal on Advances in Signal Processing*, 2007.

[16] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, 2011.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[18] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal on Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, 2013.

[19] J. M. Maestre, F. J. Muros, F. Fele, D. Muñoz de la Peña, and E. F. Camacho, "Distributed MPC based on a team game," in *Distributed Model Predictive Control Made Easy*, J. M. Maestre and R. R. Negenborn, Eds. Springer Publishing Company, Incorporated, 2013, ch. 25, pp. 407–419.

[20] N. Parikh, "Proximal Algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[21] J. F. C. Mota, "Communication-Efficient Algorithms For Distributed Optimization," Ph.D. dissertation, Carnegie Mellon University, 2013.

[22] Z. Xu, Z. Hou, Y. Han, and W. Guo, "A diagram for evaluating multiple aspects of model performance in simulating vector fields," *Geoscientific Model Development*, vol. 9, no. 12, pp. 4365–4380, 2016.

[23] C. J. Veenman, M. J. T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 54–72, 2001.

[24] B. T. Morris and M. M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1114–1127, 2008.

[25] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[26] V. Mirge, K. Verma, and S. Gupta, "Dense traffic flow patterns mining in bi-directional road networks using density based trajectory clustering," *Advances in Data Analysis and Classification*, vol. 11, no. 3, pp. 547–561, 2017.

[27] N. Anjum and A. Cavallaro, "Multifeature object trajectory clustering for video analysis," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1555–1564, 2008.

[28] J. C. Nascimento, M. A. T. Figueiredo, and J. S. Marques, "Activity Recognition Using a Mixture of Vector Fields," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1712–1725, 2013.

[29] W. Hu, X. Li, G. Tian, S. Maybank, and Z. Zhang, "An Incremental DPMM-Based Method for Trajectory Clustering, Modeling, and Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1051–1065, 2013.

[30] B. T. Morris and M. M. Trivedi, "Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2287–2301, 2011.

[31] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes," in *European Conf. on Computer Vision (ECCV)*, 2016.

[32] W. Wang, W. Lin, Y. Chen, J. Wu, J. Wang, and B. Sheng, "Finding Coherent Motions and Semantic Regions in Crowd Scenes: A Diffusion and Clustering Approach," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 756–771.

[33] W. Lin, H. Chu, J. Wu, B. Sheng, and Z. Chen, "A Heat-Map-based Algorithm for Recognizing Group Activities in Videos," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1980–1992, 2013.

[34] C. Barata, J. C. Nascimento, and J. S. Marques, "A sparse approach to pedestrian trajectory modeling using multiple motion fields," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2017, pp. 2538–2542.

[35] H. Xu, Y. Zhou, W. Lin, and H. Zha, "Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage," in *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

A. Distributed ADMM

A.1. Distributed ADMM

The ADMM algorithm is most useful in applications that have a non-differentiable cost function, such as the current vector field estimation problem (with cost function defined as in (1.3)). The ADMM can be applied in such cases because it belongs to the class of proximal gradient methods [20]. More specifically, ADMM is a proximal minimization algorithm applied to the dual problem, which iteratively applies the Douglas-Rachford operator to the sub-differential of the dual function [21].

The following toy examples deal with the problem of distributed optimization, where each graph node (p) knows only his own cost function ($\boldsymbol{f}_p$) and the shared estimates of the optimization variable ($\boldsymbol{x}$) from his neighbors ($\mathcal{N}_p$). The nodes cost functions are defined as

$$\boldsymbol{f}_p\big(\boldsymbol{x}^{(p)}\big) = \boldsymbol{x}^{(p)\top}\,\boldsymbol{M}_p\,\boldsymbol{x}^{(p)} + \boldsymbol{v}_p^\top\,\boldsymbol{x}^{(p)}\,, \tag{A.1}$$

where $\boldsymbol{M}_p$ and $\boldsymbol{v}_p$ are specific for node p. The general problem we want to solve is

$$\begin{aligned} \underset{\{\bar{\boldsymbol{x}}_l\}_{l=1}^n}{\text{minimize}} \quad & \boldsymbol{f}_1(x_{S_1}^{(1)}) + \boldsymbol{f}_2(x_{S_2}^{(2)}) + \cdots + \boldsymbol{f}_P(x_{S_P}^{(P)}) \\ \text{subject to} \quad & x_l^{(i)} = x_l^{(j)}, (i,j) \in \mathcal{E}_l, l = 1,\ldots,n\,, \end{aligned} \tag{A.2}$$

with $S_p \subseteq \{1,\ldots,n\}$ the set of components of interest for node p, l the components of the optimization variable, and $\mathcal{E}_l$ the edges of shared components. The constraints on the optimization variable thus translate in each node having a copy of the components of interest of the optimization variable. The constraints on (A.2) can alternatively be written as

$$x_l^{(i)} = x_l^{(j)}, (i,j) \in \mathcal{E}_l \Leftrightarrow \boldsymbol{A}_l\,\bar{\boldsymbol{x}}_l = \boldsymbol{0}\,, \tag{A.3}$$

where $\boldsymbol{A}_l$ is the transpose of the node-arc incidence matrix of the sub-graph of component l, i.e., $\mathcal{G}_l$.

Because the aim is a distributed optimization problem, the following coloring scheme of the nodes was adopted for each $l = 1,\ldots,n$, $\bar{\boldsymbol{x}}_l = (\bar{\boldsymbol{x}}_l^1,\ldots,\bar{\boldsymbol{x}}_l^C)$,

$$\bar{\boldsymbol{x}}_l^c = \begin{cases} \{\bar{\boldsymbol{x}}_l\}_{p\in\mathcal{V}\cap\mathcal{C}_c}, & \text{if } \mathcal{V}\cap\mathcal{C}_c \neq \emptyset \\ \emptyset, & \text{if } \mathcal{V}\cap\mathcal{C}_c = \emptyset \end{cases}, \tag{A.4}$$

where $\bar{\boldsymbol{x}}_l^c$ is the set of copies of x_l from the nodes with color c. This same notation applied to constraint (A.3) becomes $\boldsymbol{A}_l^1\,\bar{\boldsymbol{x}}_l^1 + \cdots + \boldsymbol{A}_l^C\,\bar{\boldsymbol{x}}_l^C$, $\forall_l$.

Rewriting problem (A.2) we get

$$\begin{aligned} \underset{\bar{\boldsymbol{x}}^1,\ldots,\bar{\boldsymbol{x}}^C}{\text{minimize}} \quad & \sum_{p\in\mathcal{C}_1} \boldsymbol{f}_p(x_{S_p}^{(p)}) + \cdots + \sum_{p\in\mathcal{C}_C} \boldsymbol{f}_p(x_{S_p}^{(p)}) \\ \text{subject to} \quad & \bar{\boldsymbol{A}}^1\,\bar{\boldsymbol{x}}^1 + \cdots + \bar{\boldsymbol{A}}^C\,\bar{\boldsymbol{x}}^C = \boldsymbol{0}\,, \end{aligned} \tag{A.5}$$

where $\bar{\boldsymbol{A}}^c$ is the diagonal concatenation of matrices $\bar{\boldsymbol{A}}_1^c, \bar{\boldsymbol{A}}_2^c, \ldots, \bar{\boldsymbol{A}}_n^c$

$$\bar{\boldsymbol{A}}^c = \operatorname{diag}(\bar{\boldsymbol{A}}_1^c, \bar{\boldsymbol{A}}_2^c, \ldots, \bar{\boldsymbol{A}}_n^c)\,. \tag{A.6}$$

Problem (A.5) can be solved using the multi-block/extended ADMM, where

- λ_l^{ij} is the dual variable associated to $x_l^{(i)} = x_l^{(j)}$, $(i,j) \in \mathcal{E}_l$, $l \in \{1, \ldots, n\}$,
- the augmented Lagrangian is ($\rho > 0$)

$$\mathcal{L}_\rho(\bar{\boldsymbol{x}}^1, \ldots, \bar{\boldsymbol{x}}^c; \boldsymbol{\lambda}) = \sum_{c=1}^{C} \sum_{p \in \mathcal{C}_c} \boldsymbol{f}_p(x_{S_p}^{(p)}) + \sum_{c=1}^{C} \boldsymbol{\lambda}^\top \bar{\boldsymbol{A}}^c \bar{\boldsymbol{x}}^c + \frac{\rho}{2} \Big\| \sum_{c=1}^{C} \bar{\boldsymbol{A}}^c \bar{\boldsymbol{x}}^c \Big\|^2 . \tag{A.7}$$

The problem we wish to solve consists of a sequence of C sub-problems, obtained by minimizing $\mathcal{L}_\rho$ with respect to each block $\bar{\boldsymbol{x}}^c$, and updating the dual variable $\boldsymbol{\lambda}_l^{ij}$:

Algorithm (updates)

$$\bar{\boldsymbol{x}}^{1,k+1} = \arg\min_{\bar{\boldsymbol{x}}^1} \textstyle\sum_{p \in \mathcal{C}_1} \Big(\boldsymbol{f}_p(x_{S_p}^{(p)}) + \sum_{l \in S_p} \sum_{j \in \mathcal{N}_p \cap \mathcal{V}_l} (\lambda_l^{pj,k} - \rho\, x_l^{(j),k})^\top x_l^{(p)} + \frac{\rho}{2} \sum_{l \in S_p} \mathcal{D}_{p,l}(x_l^{(p)})^2 \Big)$$

$$\vdots$$

$$\bar{\boldsymbol{x}}^{C,k+1} = \arg\min_{\bar{\boldsymbol{x}}^C} \textstyle\sum_{p \in \mathcal{C}_C} \Big(\boldsymbol{f}_p(x_{S_p}^{(p)}) + \sum_{l \in S_p} \sum_{j \in \mathcal{N}_p \cap \mathcal{V}_l} (\lambda_l^{pj,k} - \rho\, x_l^{(j),k})^\top x_l^{(p)} + \frac{\rho}{2} \sum_{l \in S_p} \mathcal{D}_{p,l}(x_l^{(p)})^2 \Big)$$

$$\lambda_l^{ij,k+1} = \lambda_l^{ij,k} + \rho \left(x_l^{(i),k+1} - x_l^{(j),k+1} \right)$$

A.2. Toy Examples

Next we present 4 toy examples where different graphs are used to solve (A.5).

A.2.1. Example 1

$\boldsymbol{f}_p$	$\boldsymbol{M}_p$	$\boldsymbol{b}_p$
$\boldsymbol{f}_1(x_1, x_2)$	$\begin{bmatrix} 1 & 1 \\ 1 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$
$\boldsymbol{f}_2(x_1, x_2)$	$\begin{bmatrix} 1 & 0.25 \\ 0.25 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 10 \\ 10 \end{bmatrix}$

- $\boldsymbol{x} = (x_1, x_2)$ is a global variable;
- constrains on the variable components are observed;
- minimization of global cost function with global solution is observed;

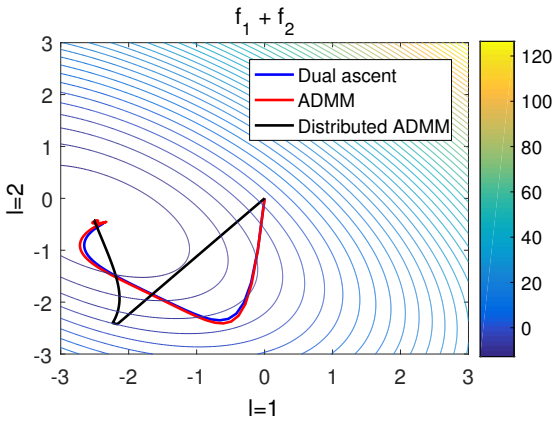
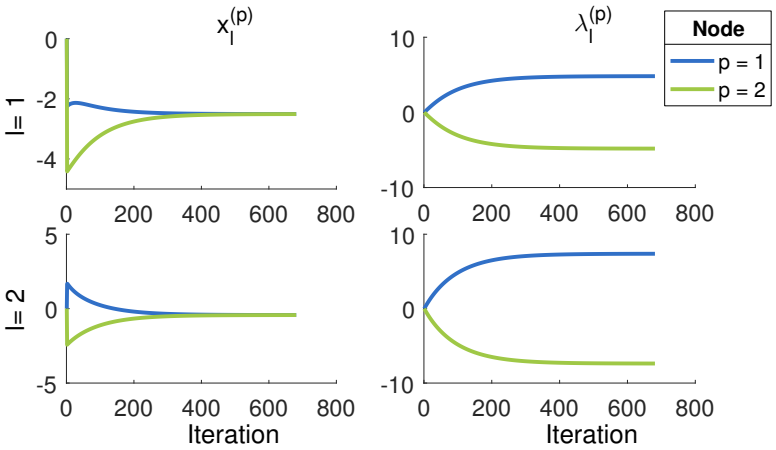
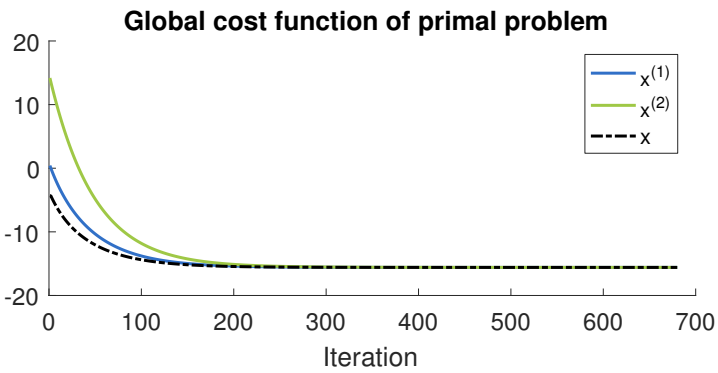


Figure A.1.: Global cost function and evolution of the solution for three optimization algorithms: (blue) Dual Ascent; (red) Alternating Direction Method of Multipliers (ADMM); (black) Distributed/Multi-block ADMM.



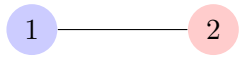
(a)



(b)

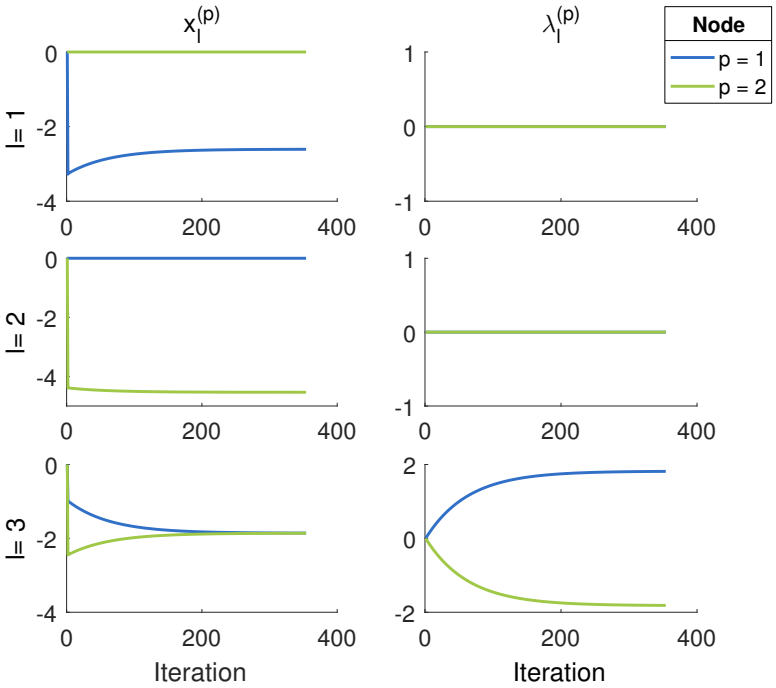
Figure A.2.: Evolution of the solution $\left(x_l^{(p)}\right)$ and the dual variable $\left(\lambda_l^{(p)}\right)$ throughout iterations of D-ADMM. Each variable component ($l = 1, 2$) is shown in the different rows of the graph. Each node is represented by different colors. In this example both cost functions depend on $\boldsymbol{x} = (x_1, x_2)$ – $\boldsymbol{f}_1\left(\boldsymbol{x}^{(1)}\right)$, $\boldsymbol{f}_2\left(\boldsymbol{x}^{(2)}\right)$. The copies of $\boldsymbol{x}$ in both nodes must comply with $x_1^{(1)} = x_1^{(2)}$ and $x_2^{(1)} = x_2^{(2)}$. The solution is $\boldsymbol{x}^* = (-2.506, -0.428)$.

A.2.2. Example 2

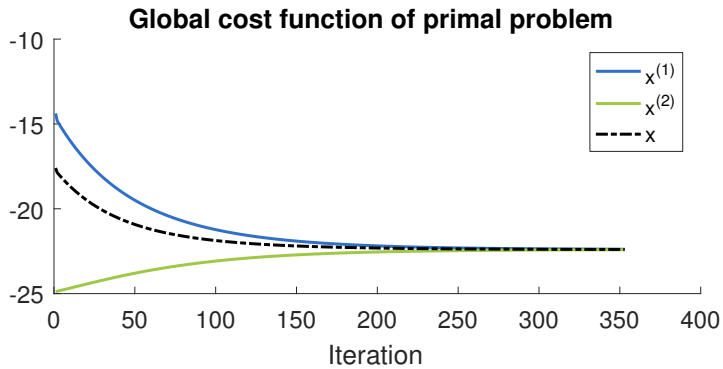


$\boldsymbol{f}_p$	$\boldsymbol{M}_p$	$\boldsymbol{b}_p$
$\boldsymbol{f}_1(x_1, x_3)$	$\begin{bmatrix} 1 & 0 & 0.75 \\ 0 & 1 & 0 \\ 0.75 & 0 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 0 \\ 8 \end{bmatrix}$
$\boldsymbol{f}_2(x_2, x_3)$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0.26 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 10 \\ 10 \end{bmatrix}$

- $\boldsymbol{x} = (x_1, x_2, x_3)$ is a mixed variable, only x_3 is global;
- constrains on x_3 are observed;
- minimization of global cost function with global solution is observed;
- nodes that do not depend on a given component show constant zero values, as expected (e.g., node $p = 2$ does not depend on x_1 and so yields constant zero values for $x_l^{(p)}$ and $\lambda_l^{(p)}$).



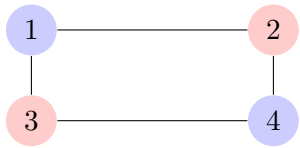
(a)



(b)

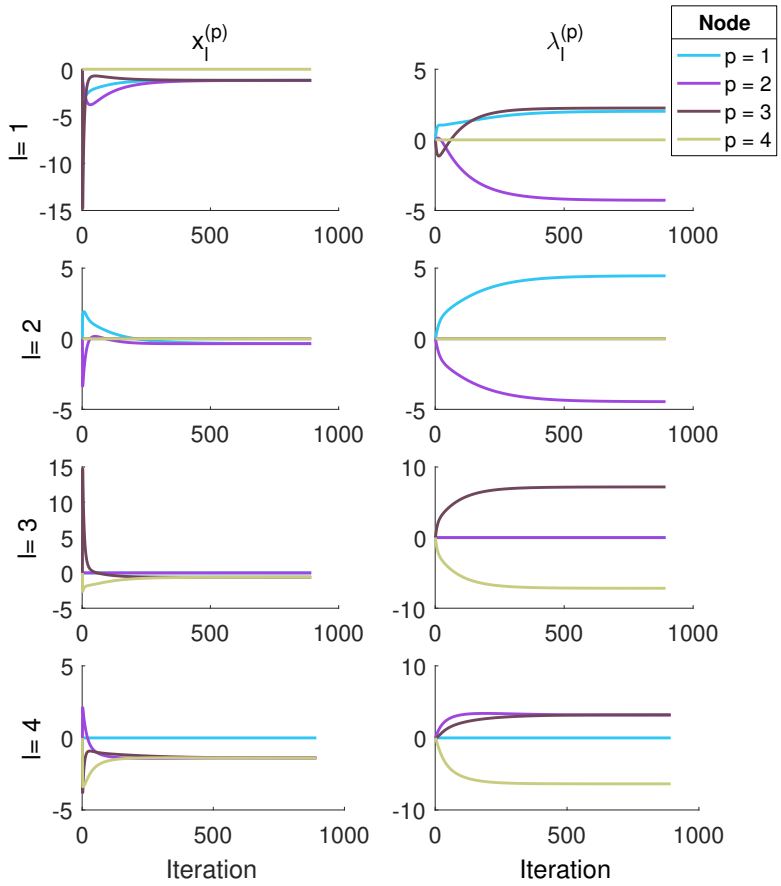
Figure A.3.: Evolution of the solution $\left(x_l^{(p)}\right)$ and the dual variable $\left(\lambda_l^{(p)}\right)$ throughout iterations of D-ADMM. Each variable component ($l = 1, 2, 3$) is shown in the different rows of the graph. Each node is represented by different colors. In this example the optimization variable is $\boldsymbol{x} = (x_1, x_2, x_3)$, and each cost function is described as $\boldsymbol{f}_1\left(x_1^{(1)}, x_3^{(1)}\right)$, $\boldsymbol{f}_2\left(x_2^{(2)}, x_3^{(2)}\right)$. The copies of $\boldsymbol{x}$ in the nodes must comply with $x_3^{(1)} = x_3^{(2)}$. The solution is $\boldsymbol{x}^* = (-2.608, -4.535, -1.861)$.

A.2.3. Example 3

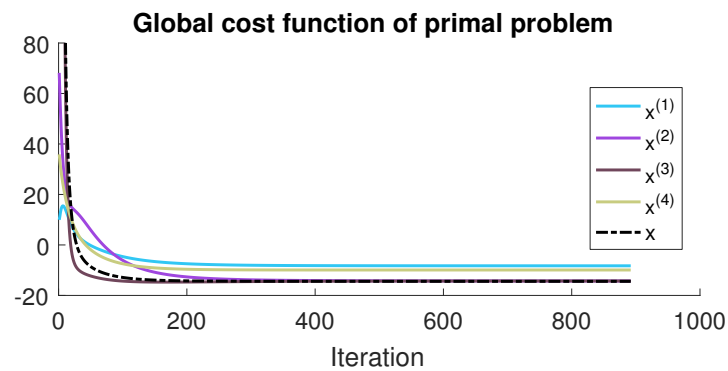


$\boldsymbol{f}_p$	$\boldsymbol{M}_p$	$\boldsymbol{b}_p$
$\boldsymbol{f}_1(x_1, x_2)$	$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1.6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix}$
$\boldsymbol{f}_2(x_1, x_2, x_4)$	$\begin{bmatrix} 1 & 1 & 0 & 0.25 \\ 1 & 1.6 & 0 & 0.75 \\ 0 & 0 & 1 & 0 \\ 0.25 & 0.75 & 0 & 1.1 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 10 \\ 0 \\ 1 \end{bmatrix}$
$\boldsymbol{f}_3(x_1, x_3, x_4)$	$\begin{bmatrix} 1 & 0 & 1 & 0.25 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1.2 & 0.5 \\ 0.25 & 0 & 0.5 & 1.1 \end{bmatrix}$	$\begin{bmatrix} 2 \\ 0 \\ -2 \\ 1 \end{bmatrix}$
$\boldsymbol{f}_4(x_3, x_4)$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1.2 & 0.5 \\ 0 & 0 & 0.5 & 1.1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 10 \\ 10 \end{bmatrix}$

- $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ is a local variable;
- constrains on $\boldsymbol{x}$ are observed;
- minimization of global cost function with global solution is observed;
- nodes that do not depend on a given component show constant zero values.



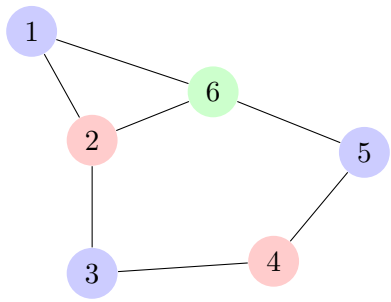
(a)



(b)

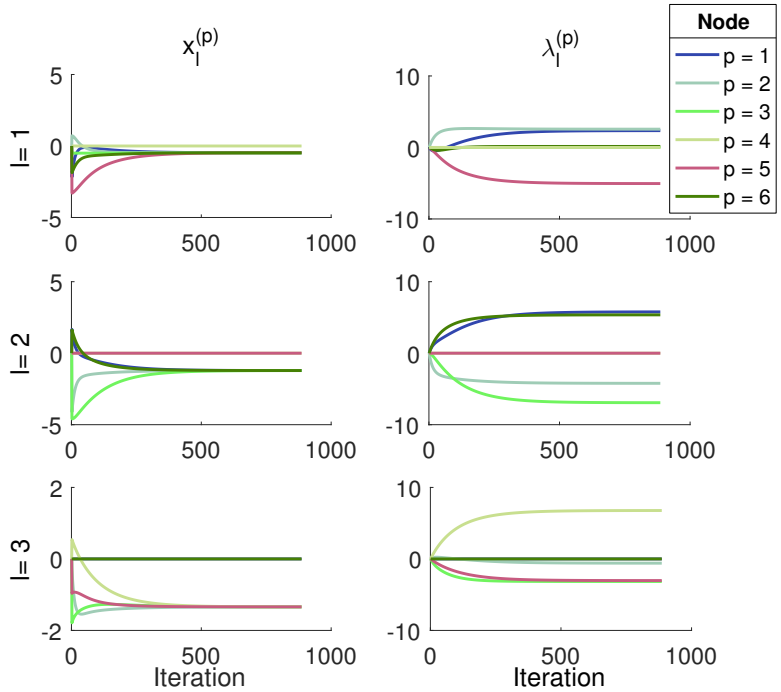
Figure A.4.: Evolution of the solution $\left(x_l^{(p)}\right)$ and the dual variable $\left(\lambda_l^{(p)}\right)$ throughout iterations of D-ADMM. In this example the optimization variable is $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$, and each cost function is described as $\boldsymbol{f}_1\left(x_1^{(1)}, x_2^{(1)}\right)$, $\boldsymbol{f}_2\left(x_1^{(2)}, x_2^{(2)}, x_4^{(2)}\right)$, $\boldsymbol{f}_3\left(x_1^{(3)}, x_3^{(3)}, x_4^{(3)}\right)$, $\boldsymbol{f}_4\left(x_3^{(4)}, x_4^{(4)}\right)$. Each variable component ($l = 1, 2, 3, 4$) is shown in the different rows of the graph. Each node is represented by different colors. The copies of $\boldsymbol{x}$ in the nodes must comply with $x_1^{(1)} = x_1^{(2)} = x_1^{(3)}$, $x_2^{(1)} = x_2^{(2)}$, $x_3^{(3)} = x_3^{(4)}$, and $x_4^{(2)} = x_4^{(3)} = x_4^{(4)}$. The solution is $\boldsymbol{x}^* = (-1.179, -0.346, -0.605, -1.386)$.

A.2.4. Example 4 - from [1]

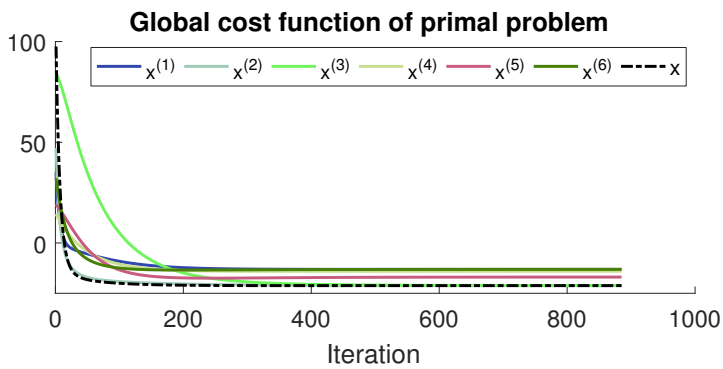


$\boldsymbol{f}_p$	$\boldsymbol{M}_p$	$\boldsymbol{b}_p$
$\boldsymbol{f}_1(x_1, x_2)$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1.6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$
$\boldsymbol{f}_2(x_1, x_2, x_3)$	$\begin{bmatrix} 1 & 0.25 & 0.75 \\ 0.25 & 1.2 & 1 \\ 0.75 & 1 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 10 \\ 8 \end{bmatrix}$
$\boldsymbol{f}_3(x_2, x_3)$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0.25 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 10 \\ 10 \end{bmatrix}$
$\boldsymbol{f}_4(x_3)$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.8 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix}$
$\boldsymbol{f}_5(x_1, x_3)$	$\begin{bmatrix} 1 & 0 & 0.75 \\ 0 & 1 & 0 \\ 0.75 & 0 & 1.6 \end{bmatrix}$	$\begin{bmatrix} 8 \\ 0 \\ 8 \end{bmatrix}$
$\boldsymbol{f}_6(x_1, x_2)$	$\begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1.2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix}$

- $\boldsymbol{x} = (x_1, x_2, x_3)$ is a local variable;
- constrains on $\boldsymbol{x}$ are observed;
- minimization of global cost function with global solution is observed;
- nodes that do not depend on a given component show constant zero values, as expected.



(a)



(b)

Figure A.5.: Evolution of the solution $\left(x_l^{(p)}\right)$ and the dual variable $\left(\lambda_l^{(p)}\right)$ throughout iterations of D-ADMM. In this example the optimization variable is $\boldsymbol{x} = (x_1, x_2, x_3)$, and each cost function is described as $\boldsymbol{f}_1\left(x_1^{(1)}, x_2^{(1)}\right)$, $\boldsymbol{f}_2\left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}\right)$, $\boldsymbol{f}_3\left(x_2^{(3)}, x_3^{(3)}\right)$, $\boldsymbol{f}_4\left(x_3^{(4)}\right)$, $\boldsymbol{f}_5\left(x_1^{(5)}, x_3^{(5)}\right)$, $\boldsymbol{f}_6\left(x_1^{(6)}, x_2^{(6)}\right)$. Each variable component $(l = 1, 2, 3)$ is shown in the different rows of the graph. Each node is represented by different colors. The copies of $\boldsymbol{x}$ in the nodes must comply with $x_1^{(1)} = x_1^{(2)} = x_1^{(5)} = x_1^{(6)}$, $x_2^{(1)} = x_2^{(2)} = x_2^{(3)} = x_2^{(6)}$, and $x_3^{(2)} = x_3^{(3)} = x_3^{(4)} = x_3^{(5)}$. The solution is $\boldsymbol{x}^* = (-0.474, -1.214, -1.341)$.