Interaction Kernel Learning in Multi-Agent Dynamical System

M. Bongini, M. Fornasier, M. Hansen, M. Maggioni

July 30, 2014

Abstract

We address the problem of...

1 The framework

We assume that we are given N agents whose dynamics satisfies, for every $t > 0$, the following set of ordinary differential equations

$$\begin{cases} \dot{x_i}(t) = v_i(t), \\ \dot{v_i}(t) = \frac{1}{N}\sum_{j=1}^N a(\|x_j(t) - x_i(t)\|_{\ell_2^d})(v_j(t) - v_i(t)), \end{cases}$$

where the *state parameter* x_i and the *consensus parameter* v_i of the i-th agent both lie in the euclidean space $\mathbb{R}^d$ equipped with the euclidean norm $\|\cdot\|_{\ell_2^d}$ (whose subscript shall be omitted when clear from the context).

The *interaction kernel* a is assumed to be unknown.

2 Numerical experiments

2.1 The algorithm

The algorithm takes as inputs a snapshots' dataset $\mathcal{D} = \{(x_i(n\Delta t), v_i(n\Delta t)) \mid i = 1, \ldots, N, n = 0, \ldots [T/\Delta t]\}$ in which each couple $(x_i(n\Delta t), v_i(n\Delta t))$ represents the position and the velocity of the i-th agent at time $n\Delta t$. From this dataset we can calculate the empirical accelerations of the agents at time $n\Delta t$ as

$$v_i'(n\Delta t) := \frac{v_i((n+1)\Delta t) - v_i(n\Delta t)}{\Delta t}.$$

To determine the function a, we consider, for any $n = 0, \ldots [T/\Delta t]$, the function

$$\psi_n^{\mathcal{D}}(a) := \frac{1}{N}\sum_{i=1}^N a\left(\|x_i(n\Delta t) - x_j(n\Delta t)\|\right)\left(v_j(n\Delta t) - v_i(n\Delta t)\right) - v_i'(n\Delta t),$$

and consider $\psi^{\mathcal{D}}(a) = \left[\psi_0^{\mathcal{D}}(a), \ldots, \psi_{[T/\Delta t]}^{\mathcal{D}}(a)\right]$.

The idea is to minimize the quantity $\left\|\psi^{\mathcal{D}}(f)\right\|^2$ in a set of functions Λ_m in which a suitable approximation of the unknown function a lives: we start from a function space Λ_0 and enlarge it at every step m until we reach the desired accuracy for a.

To this end, given a vector $\mathbf{t}^m = (t_1^m, \ldots, t_{k_m}^m) \in \mathbb{R}^{k_m}$ of nodes in increasing order, we take Λ_m to be the linear subspace of dimension $k_m - d$ generated by the spline basis of order d with nodal basis $\mathbf{t}^m$, that is

$$\Lambda_m = \operatorname{span}\left(\{\varphi_\ell^{\mathbf{t}^m}\}_{\ell=1}^{k_m-d}\right).$$

Hence, we assume that a good approximation of the function a lies in Λ_m, that is it has the form

$$a \approx \sum_{\ell=1}^{k_m-d} c_\ell \varphi_\ell^{\mathbf{t}^m},$$

for given $\mathbf{c} = (c_1, \ldots, c_{k_m-d}) \in \mathbb{R}^{k_m-d}$. This ansatz allow us to rewrite $\psi^{\mathcal{D}}$ as

$$\begin{aligned}\psi_n^{\mathcal{D}}(a) \approx \psi_n^{\mathcal{D},\mathbf{t}^m}(\mathbf{c}) &= \frac{1}{N}\sum_{i=1}^{N}\sum_{\ell=1}^{k_n-d} c_\ell \varphi_\ell^{\mathbf{t}^m}(\|x_i(n\Delta t) - x_j(n\Delta t)\|)\,(v_j(n\Delta t) - v_i(n\Delta t)) \\ &\quad - v_i'(n\Delta t) \\ &= \mathbf{c} \cdot A_n^{\mathbf{t}^m} - \mathbf{d}_n,\end{aligned}$$

where, for every $n = 0, \ldots, [T/\Delta t]$, we set $(\mathbf{d}_n)_i = v_i'(n\Delta t)$ and

$$(A_n^{\mathbf{t}^m})_{\ell ij} = \frac{1}{N}\varphi_\ell^{\mathbf{t}^m}\left(\|x_i(n\Delta t) - x_j(n\Delta t)\|\right)(v_j(n\Delta t) - v_i(n\Delta t)).$$

This means that, for every m,

$$\psi^{\mathcal{D},\mathbf{t}^m}(\mathbf{c}) = \mathbf{c} \cdot A^{\mathbf{t}^m} - \mathbf{d}, \quad \text{where} \quad A^{\mathbf{t}^m} = [A_0^{\mathbf{t}^m}, \ldots, A_{[T/\Delta t]}^{\mathbf{t}^m}] \text{ and } \mathbf{d} = [\mathbf{d}_0, \ldots, \mathbf{d}_{[T/\Delta t]}].$$

To clarify the situation, we point out that $A_n^{\mathbf{t}^m} \in \mathbb{R}^{(k_m-d)\times N\times N}$ and $\mathbf{d}_n \in \mathbb{R}^N$, thus $A^{\mathbf{t}^m} \in \mathbb{R}^{(k_m-d)\times N\times N[T/\Delta t]}$ and $\mathbf{d} \in \mathbb{R}^{N\times[T/\Delta t]}$.

This means that, for every m, we are facing the following least-squares minimization procedure:

$$P(m) := \text{minimize } \left\|\mathbf{c} \cdot A^{\mathbf{t}^m} - \mathbf{d}\right\|^2 \text{ among all } \mathbf{c} \in \mathbb{R}^{k_m-d}.$$

2.1.1 Refinement procedure

As $\mathbf{t}^0$ we take the vector $\left(j\frac{M}{s}\right)_{j=1}^{s}$ for given $M > 0$ and $s > 1$ (s is an integer), while as $\mathbf{t}^1$ we take $\left(j\frac{M}{2s}\right)_{j=1}^{2s}$ for the same M and s as before. We then solve the two minimization problems $P(0)$ and $P(1)$ (whose solutions we indicate with $\mathbf{c}^0$ and $\mathbf{c}^1$, respectively) and consider the two obtained approximations of the interaction kernel, namely:

$$a_0 = \sum_{\ell}^{s-d} c_\ell^0 \varphi_\ell^{\mathbf{t}^0} \quad \text{and} \quad a_1 = \sum_{\ell}^{2s-d} c_\ell^1 \varphi_\ell^{\mathbf{t}^1},$$

where we remark that d is the order of the spline basis elements.

At this point, we compute the forward difference quotients of a_0 and a_1 at every point of $\mathbf{t}^1$, that is for every j such that $t^1_{j+1} \in \mathbf{t}^1$ we compute

$$a_i'(t^1_j; t^1_{j+1}) = \frac{a_i(t^1_{j+1}) - a_i(t^1_j)}{t^1_{j+1} - t^1_j} \qquad \text{for} \qquad i = 1, 2.$$

We set a tolerance $\varepsilon > 0$ and we indicate with

$$I^1_\varepsilon = \left\{ (t^1_j, t^1_{j+1}) \in \mathbb{R}^2 \mid \left\| a_1'(t^1_j; t^1_{j+1}) - a_0'(t^1_j; t^1_{j+1}) \right\| > \varepsilon \right\}$$

the set of couples of nodes for which the forward differences differ more than the tolerance. Hence, we construct the next nodal set $\mathbf{t}^2$ by taking all the node in $\mathbf{t}^1$, adding the middle point of every couple in the set I^1_ε and then reordering in increasing order: this means that, $t \in \mathbf{t}^2$ if and only if $t \in \mathbf{t}^1$ or there exist a couple $(t^1_j, t^1_{j+1}) \in I^1_\varepsilon$ such that

$$t = \frac{t^1_{j+1} + t^1_j}{2}.$$

At this point we repeat the construction of I^2_ε and $\mathbf{t}^3$ as before, until we have $I^m_\varepsilon = \emptyset$ for some m.

2.1.2 Known issues

The algorithm has basically two issues.

- the oscillatory effect at 0: this is due to the fact that the elements of the spline basis are zero at their starting node. This problem seems to be not solvable with the trick of taking as first node in $\mathbf{t}^0$ the point -1. It could be solved by imposing that the new refinement does not introduce a "worse" oscillatory effect than the old one. How much oscillatory an approximation is can be measured by taking the sum of the moduli of the finite differences of the approximation: the higher this value is, the more oscillatory the approximation is.

- the introduction of new points worsen previously accurate part of the approximation: this seems to occur often. I don't know if this issue is related to the first one.

2.2 Smooth kernel

For the first test, we consider the following interaction kernel:

$$a(r) = \frac{100}{(100 + r^2)^2} + \frac{0.04}{30 + (r - 20)^2} - \frac{0.04}{(30 + (r - 40)^2}.$$

portrayed in red in Figure 1.

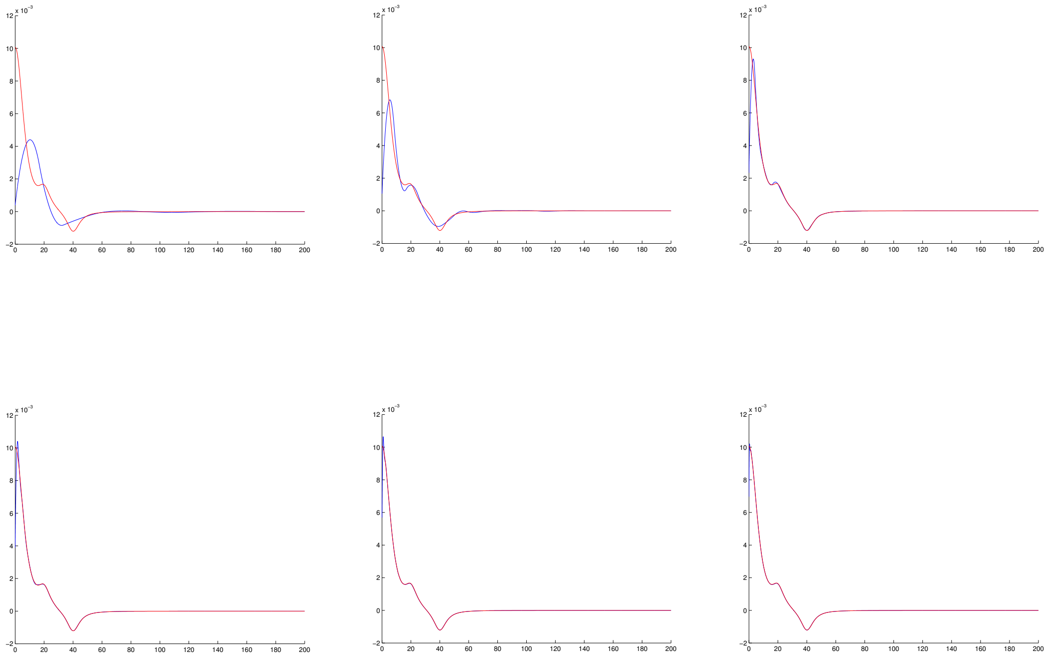

Figure 1:

2.3 Oscillatory kernel

For the second test, we take the following highly oscillating interaction kernel:

$$a(r) = \sin(3r)\frac{100}{(100 + r^2)^{\frac{4}{5}}} + \sin\left(\frac{r}{10}\right)\frac{100}{30 + (r - 50)^2},$$

depicted in red in Figure 2.

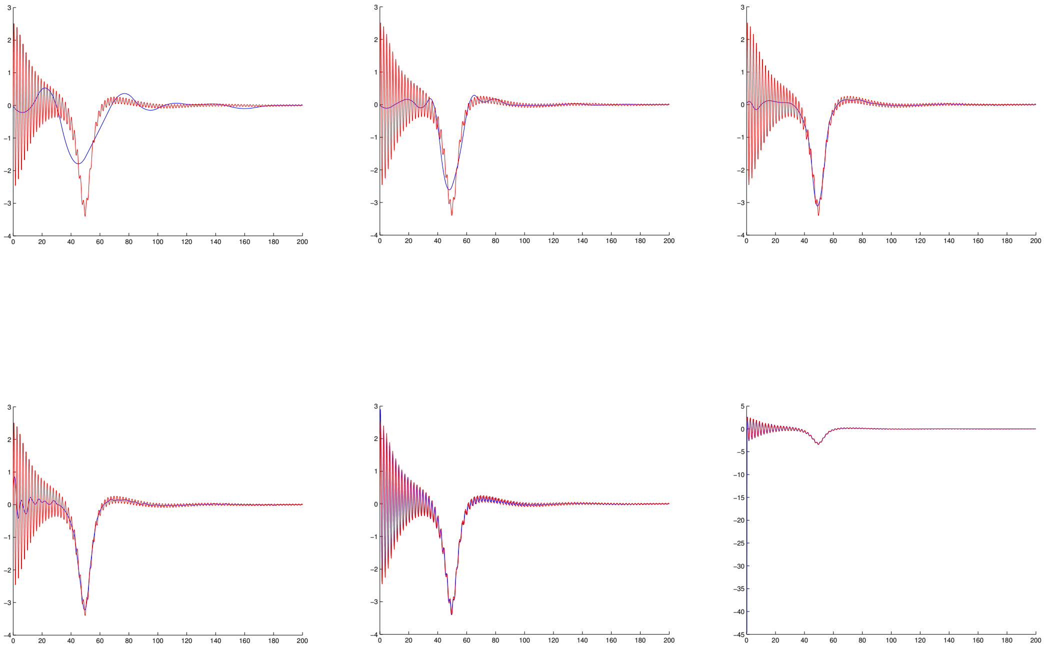

Figure 2:

The difficulty in correctly approximating the value of the interaction kernel at 0 can be seen in the bottom-left picture. The oscillatory effect that takes place adds up at every iteration of the algorithm and has a malicious effect on the whole approximation.

2.4 Discontinuous kernel

As a last example, we consider a discontinuous interaction kernel of the form

$$a(r) = H(10-r)\frac{2}{(100+r^2)^{\frac{4}{5}}} + H(r-20)H(40-r)\frac{2}{35+(r-40)^2} + H(r-45)\frac{2}{35+(r-40)^2},$$

where H is the Heaviside function. It is again portrayed in red in Figure 3. The arising of a Gibbs phenomenon at the discontinuity jumps of the interaction kernel is clearly visible.

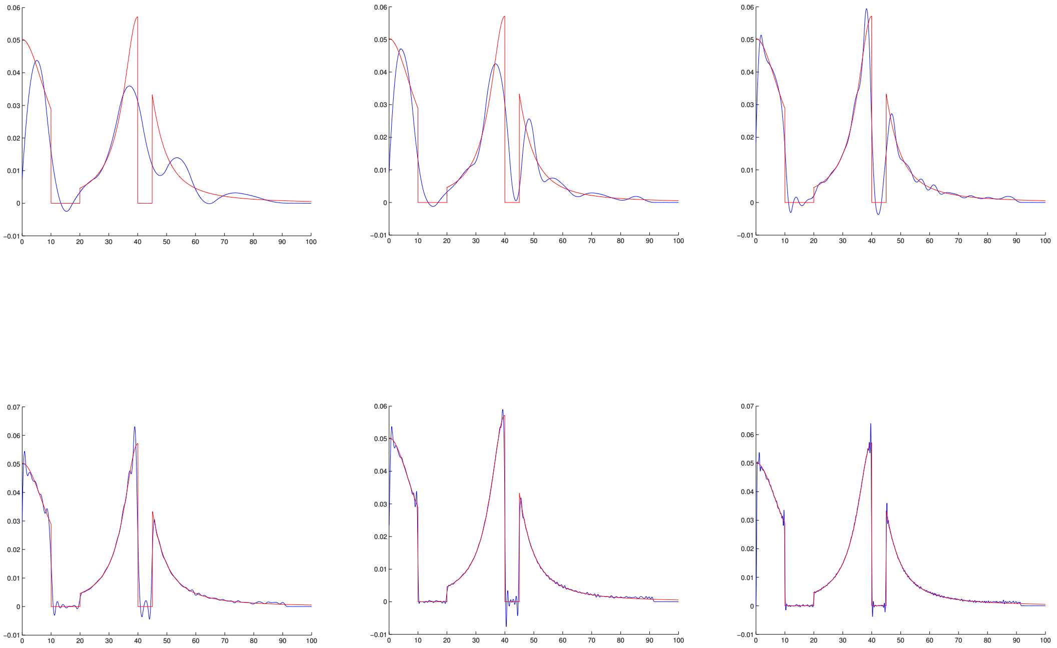
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
−0.01
0
10
20
30
40
50
60
70
80
90
100

Figure 3: