

Interaction Kernel Learning in Multi-Agent Dynamical System

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

July 31, 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 $\mathcal{D} = \{(x_i(n\Delta t), v_i(n\Delta t)) \mid i = 1, \dots, N, n = 0, \dots, [T/\Delta t]\}$ 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 $n = 0, \dots, [T/\Delta t]$, the function

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

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

The idea is to minimize the quantity $\|\psi^{\mathcal{D}}(f)\|^2$ 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 $\mathbf{t}^m = (t_1^m, \dots, t_{k_m}^m) \in \mathbb{R}^{k_m}$ 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

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

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

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

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

$$(\mathbf{A}_n^{\mathbf{t}^m})_{\ell i} = \frac{1}{N} \sum_{j=1}^N \varphi_\ell^{\mathbf{t}^m}(\|x_i(n\Delta t) - x_j(n\Delta t)\|) (v_j(n\Delta t) - v_i(n\Delta t)).$$

This means that, for every m ,

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

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

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

2.1.1 Refinement procedure

As \mathbf{t}^0 we take the vector $(j \frac{M}{s})_{j=1}^s$ for given $M > 0$ and $s > 1$ (s is an integer), while as \mathbf{t}^1 we take $(j \frac{M}{2s})_{j=1}^{2s}$ 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 $t_{j+1}^1 \in \mathbf{t}^1$ we compute

$$a'_i(t_j^1; t_{j+1}^1) = \frac{a_i(t_{j+1}^1) - a_i(t_j^1)}{t_{j+1}^1 - t_j^1} \quad \text{for} \quad i = 1, 2.$$

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

$$I_{\varepsilon}^1 = \{(t_j^1, t_{j+1}^1) \in \mathbb{R}^2 \mid \|a'_1(t_j^1; t_{j+1}^1) - a'_0(t_j^1; t_{j+1}^1)\| > \varepsilon\}$$

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_{ε}^1 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 $(t_j^1, t_{j+1}^1) \in I_{\varepsilon}^1$ such that

$$t = \frac{t_{j+1}^1 + t_j^1}{2}.$$

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

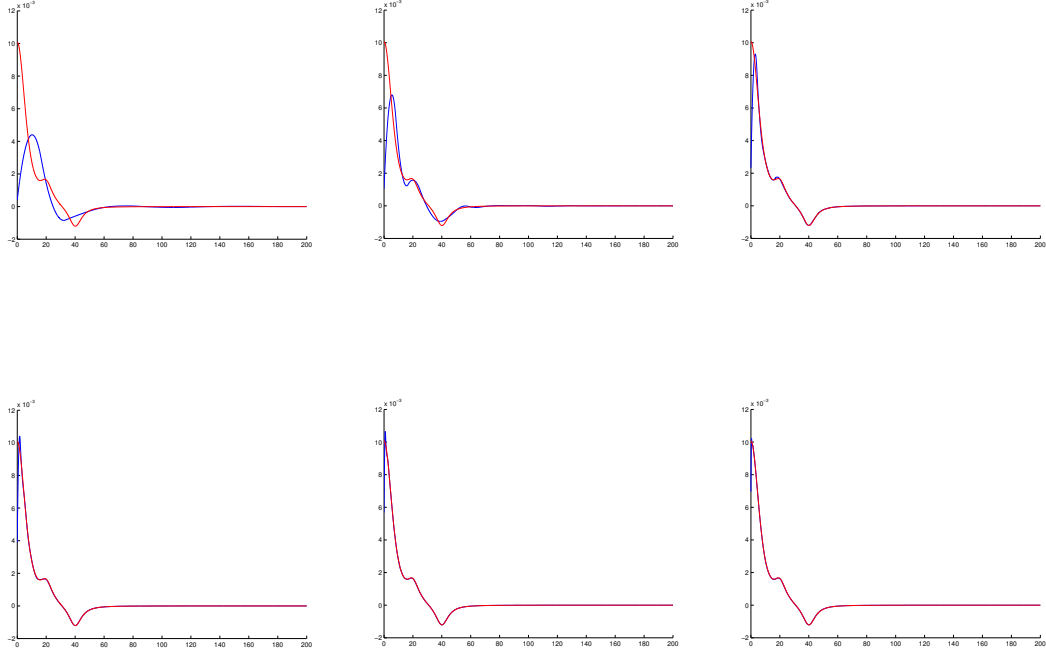


Figure 1:

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.

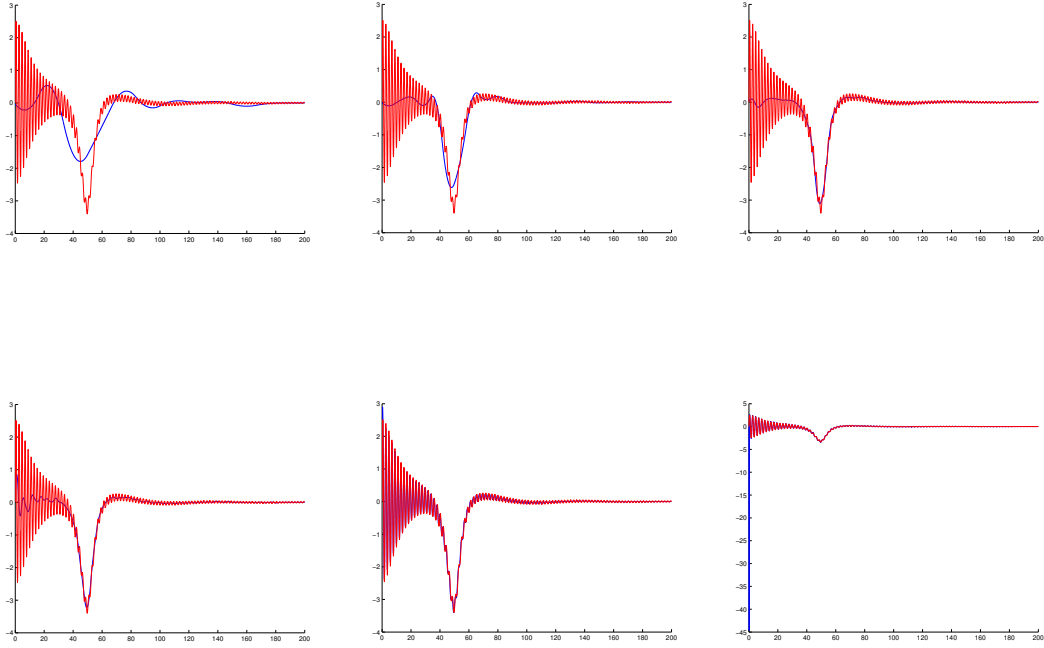


Figure 2:

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.

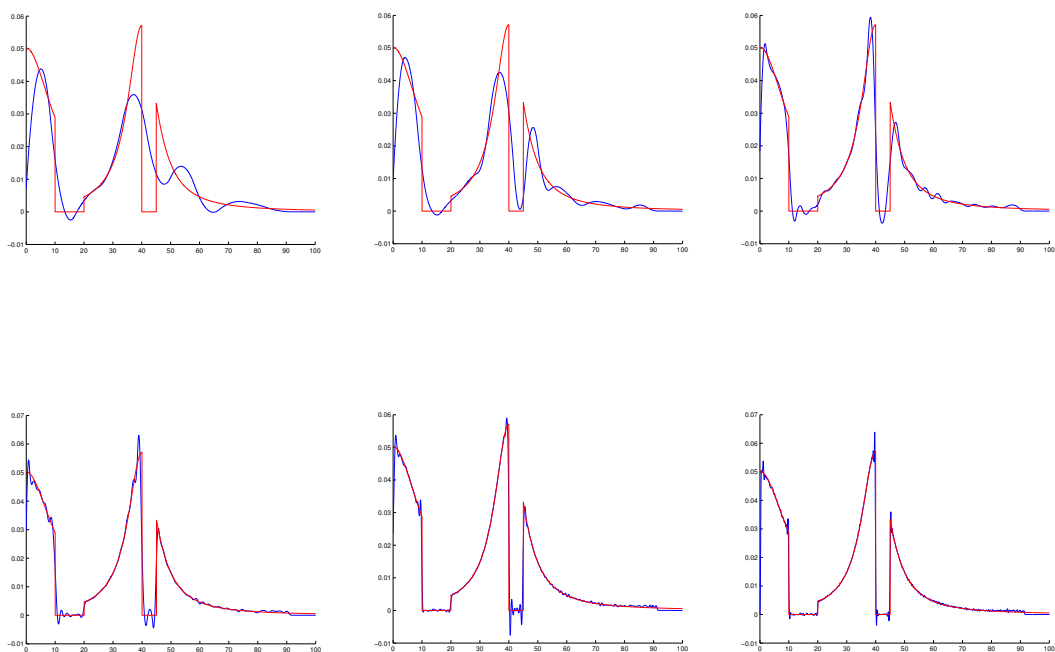


Figure 3:

3 Theoretical framework (as discussed with Massimo)

Here we consider a **first-order** dynamical system: More precisely, with the potential

$$E(x) = \sum_{i,j=1}^N A(|x_i - x_j|)$$

the dynamical system is given by

$$\dot{x} = -\nabla E(x) \iff \dot{x}_i = \sum_{j=1}^N \frac{a(|x_i - x_j|)}{|x_i - x_j|} (x_j - x_i)$$

with N agents $x_j \in \mathbb{R}^d$, where we put $a = A'$.

We assume of the unknown function a that it can be represented in some multiscale basis, i.e. that we can write

$$a(r) \equiv a_\Lambda(r) \sum_{\lambda \in \Lambda} a_\lambda \varphi_\lambda,$$

where φ_λ are e.g. wavelet or spline functions.

First idea: Given the set Λ , we put $J_\Lambda = \bigcup_{\lambda \in \Lambda} \text{supp } \varphi_\lambda$, and choose an interval $I_\Lambda = [t, t + \Delta t]$ such that

$$|x_i(s) - x_j(s)| \in J_\Lambda \quad \forall i, j \quad \forall s \in I_\Lambda.$$

Then we define the functional

$$\begin{aligned} \Delta^L(a, \hat{a}_\Lambda) &= \int_t^{t+\Delta t} \left| \left(\dot{x}_i(s) - \sum_{j=1}^N \frac{\hat{a}_\Lambda(|x_i(s) - x_j(s)|)}{|x_i(s) - x_j(s)|} (x_j(s) - x_i(s)) \right)_{i=1}^N \right|^2 ds \\ &= \int_t^{t+\Delta t} \left| \left(\sum_{j=1}^N \frac{a_\Lambda(|x_i(s) - x_j(s)|) - \hat{a}_\Lambda(|x_i(s) - x_j(s)|)}{|x_i(s) - x_j(s)|} (x_j(s) - x_i(s)) \right)_{i=1}^N \right|^2 ds. \end{aligned}$$

Of course these two versions only coincide if the \dot{x}_i are known exactly (which generally is not true). This functional is to be optimized to find the approximation \hat{a}_Λ . To ensure convergence to the true function a_λ we need to relate $\Delta^L(a, \hat{a}_\Lambda)$ to some (quasi-)norm $\|a_\Lambda - \hat{a}_\Lambda\|$.

Different objective: While in this way a (good) approximation of a_Λ may be obtained, it is equally desirable that the dynamics generated in the above way by the approximation stays close to the one generated by a_Λ .

3.1 Upper bound

Since for $p \leq q$ it holds $\|\alpha\|_{\ell_p} \leq N^{1/p-1/q}\|\alpha\|_{\ell_q}$, we obtain using triangle inequality

$$\begin{aligned}
\Delta^L(a, \widehat{a}_\Lambda) &= \int_t^{t+\Delta t} \left\| \left(\sum_{j=1}^N \frac{a_\Lambda(|x_i(s) - x_j(s)|) - \widehat{a}_\Lambda(|x_i(s) - x_j(s)|)}{|x_i(s) - x_j(s)|} (x_j(s) - x_i(s)) \right)_{i=1}^N \right\|_{\ell_2^N}^2 ds \\
&= \int_t^{t+\Delta t} \sum_{i=1}^N \left\| \sum_{j=1}^N \frac{a_\Lambda(|x_i(s) - x_j(s)|) - \widehat{a}_\Lambda(|x_i(s) - x_j(s)|)}{|x_i(s) - x_j(s)|} (x_j(s) - x_i(s)) \right\|_{\ell_2^d}^2 ds \\
&\leq \int_t^{t+\Delta t} \sum_{i=1}^N \left(\sum_{j=1}^N |a_\Lambda(|x_i(s) - x_j(s)|) - \widehat{a}_\Lambda(|x_i(s) - x_j(s)|)| \right)^2 ds \\
&\leq N^{1/2} \int_t^{t+\Delta t} \sum_{i=1}^N \sum_{j=1}^N |a_\Lambda(|x_i(s) - x_j(s)|) - \widehat{a}_\Lambda(|x_i(s) - x_j(s)|)|^2 ds \\
&\leq N^{1/2} |\Lambda|^{1/2} \int_t^{t+\Delta t} \sum_{i=1}^N \sum_{j=1}^N \sum_{\lambda \in \Lambda} |a_\lambda - \widehat{a}_\lambda|^2 \varphi_\lambda^2(|x_i(s) - x_j(s)|) ds.
\end{aligned}$$

An estimate in some weighted ℓ_2 -norm is now easily obtained, if we assume the functions φ_λ to be bounded.

3.2 Lower bound

This time we can use Jensen's inequality

$$\begin{aligned}
\Delta^L(a_\Lambda, \widehat{a}_\Lambda) &\geq \frac{1}{\sqrt{\Delta t}} \left\| \int_t^{t+\Delta t} \left(\sum_{j=1}^N (a_\Lambda(|z_{ij}|) - \widehat{a}_\Lambda(|z_{ij}|)) \operatorname{sgn}(z_{ji}) \right)_{i=1}^N ds \right\|^2 \\
&= \frac{1}{\sqrt{\Delta t}} \left\| \int_t^{t+\Delta t} \left(\sum_{j=1}^N \sum_{\lambda \in \Lambda} (a_\lambda - \widehat{a}_\lambda) \varphi_\lambda(|z_{ij}|) \operatorname{sgn}(z_{ji}) \right)_{i=1}^N ds \right\|^2 \\
&= \frac{1}{\sqrt{\Delta t}} \left\| \left(\int_t^{t+\Delta t} \operatorname{diag}(\Phi_\Lambda(s)Z(s)) ds (\mathbf{a} - \widehat{\mathbf{a}}) \right)_{i=1}^N \right\|^2.
\end{aligned}$$

Here diag refers to the diagonal entries of a square matrix, in this case of $\Phi_\lambda(s)Z(s)$, which for every $\lambda \in \Lambda$ is an $N \times N$ matrix. In this way lower bounds for Δ^L can be obtained from properties of these matrices. For instance, a bound in some ℓ_2 -norm can be derived from estimates of singular values (eigenvalues) of the $|\Lambda| \times |\Lambda|$ -matrix $\operatorname{diag}(\Phi_\Lambda(s)Z(s))^\top \operatorname{diag}(\Phi_\Lambda(s)Z(s))$.

3.3 Related issues

- If we denote $\Gamma(a)$ the trajectories of the dynamical system generated by a , i.e. $\Gamma(a)(t)_i = x_i$, does $\Delta^L(a, \widehat{a})$ allow assertions on the relation between $\Gamma(a)$ and $\Gamma(\widehat{a})$? In other words,

if a is close to \widehat{a} (in which sense?), is also $x(t)$ close to $\widehat{x}(t)$ (at every time t , or in an L_2 -sense). Basically, we need to relate the global error on the time interval $[0, T]$ to the local error on $[t, t + \Delta t]$.

- If, as indicated above, $\Delta^L(a, \widehat{a})$ is equivalent to $\|a - \widehat{a}\|_{\ell_2(\Lambda)}$, then minimizing Δ^L also gives a quasi-optimal approximation of a . Then, depending on the choice of the (multiscale) basis/system (wavelet, Finite element, spline), regularity of a immediately transfers to convergence rates of the adaptive algorithm. Vice versa, choosing a spline basis, we can describe Besov spaces via Δ^L (since Approximation spaces for spline approximation exactly coincide with Besov spaces, as shown by DeVore and Popov, 1988)