AMATH 563: Computational Report
Learning Lotka–Volterra Dynamics with Kernel Methods

Jiaji Qu
Department of Applied Mathematics, University of Washington
jiajiq@uw.edu

13 May 2025

1. Introduction

In this report, we study the Lotka–Volterra (LV) predator–prey model

$$\dot{p}_1 = \alpha p_1 - \beta p_1 p_2, \qquad \dot{p}_2 = -\gamma p_2 + \delta p_1 p_2, \tag{1}$$

with $(\alpha, \beta, \gamma, \delta) > 0$. In many practical settings one has measurements of (p_1, p_2) but no direct knowledge of the model in (1) or its parameters. The objective of this report is therefore to **learn** the vector field $f = (f_1, f_2)\colon \mathbb{R}^2 \to \mathbb{R}^2$ from some given, sparse observations $y(t_n) = (p_1, p_2)(t_n)$ employing some kernel methods. The assignment asks me to adopt a three-step pipeline:

Step 1: Trajectory fitting - RBF kernel-ridge regression (KRR) of $p_i(t)$.

Step 2: Derivative estimation - Analytic derivative of the reproducing kernel.

Step 3: Vector-field learning - Polynomial-kernel KRR in (p_1, p_2).

Throughout we use the synthetic a priori ground-truth parameters $(\alpha, \beta, \gamma, \delta) = (1,\ 0.1,\ 0.075,\ 1.5)$ and report all errors in population units per time unit.

2. Methods

2.1 Data generation

For the purposes of accuracy, a single LV trajectory was simulated with a fourth-order Runge–Kutta solver on $t \in [0, 20]$ using $\Delta t = 10^{-3}$. Observations were sub-sampled every 0.4 time-units, yielding $N = 50$ pairs.

2.2 Step 1: Trajectory regression

Given samples $\{(t_n, p_i(t_n))\}_{n=0}^{N-1}$ we solve

$$\hat{p}_i = \arg\min_{g \in \mathcal{H}_k} \sum_{n=0}^{N-1} \left| g(t_n) - p_i(t_n) \right|^2 + \alpha \, \|g\|^2_{\mathcal{H}_k},$$

with the Gaussian kernel $k(t, t') = \exp[-\gamma(t - t')^2]$. Closed-form coefficients are obtained by solving $(K + \alpha I)a = p$. Hyper-parameters (α, γ) were chosen by five-fold cross-validation over the grid $\alpha \in \{10^{-6}, \ldots, 10^{-1}\}$, $\gamma \in \{10^{-2}, 10^{-1}, \ldots, 10\}$. The optimal values found by the code $\alpha = 1.00 \times 10^{-3}, \gamma = 1.68$ (prey) and $\alpha = 1.00 \times 10^{-2}, \gamma = 0.03$ (predator).

2.3 Step 2: Derivative estimation

Because differentiation is continuous in RKHS, we can immediately apply the time-derivative and get a closed form formula

$$\widehat{\dot{p}_i}(t) \;=\; \sum_{j=0}^{N-1} a_{ij} \quad \underbrace{\frac{\partial}{\partial t} k(t, t_j)}_{-2\gamma(t-t_j)\,k(t,t_j)} \quad =\; -2\gamma \sum_j a_{ij}(t-t_j)k(t,t_j).$$

where evaluating at the observation times gives us a labeled set $\left\{(\hat{p}_1, \hat{p}_2),\, \widehat{\dot{p}}\,\right\}_n$ for Step 3.

2.4 Step 3: Vector-field learning

Let $\widehat{\boldsymbol{p}}(t_n) = (\hat{p}_1, \hat{p}_2)(t_n)$ be the 100 points generated in Step 2 and $\widehat{\dot{\boldsymbol{p}}}(t_n)$ their analytic derivatives computed by Step 2. The essence of Step 3 is to find some surrogate $h_i \colon \mathbb{R}^2 \to \mathbb{R}$ that minimizes the regularized problem

$$\hat{f}_i \;=\; \arg\min_{h \in \mathcal{H}_\kappa} \sum_{n=1}^{100} \left| \,\widehat{\dot{p}_i}(t_n) - h(\widehat{\boldsymbol{p}}(t_n)) \right|^2 + \lambda \, \|h\|^2_{\mathcal{H}_\kappa}, \tag{2}$$

where κ is a PDS kernel and $\lambda > 0$ is chosen by cross-validation. We investigated three increasingly complex choices for the kernel κ:

A. Naive RBF-KRR without scaling

$$\kappa_{\mathrm{RBF}}(x, x') \;=\; \exp\!\left[-\gamma \|x - x'\|^2\right], \quad x = (p_1, p_2) \text{ (raw units)},$$

with $(\alpha, \gamma) \in \{10^{-6} \ldots 10^{-1}\} \times \{10^{-2} \ldots 10\}$ searched on a logarithmic grid.

B. RBF-KRR with scaled/standardized inputs. To remove the scale mismatch I tried a z-score transformation $\tilde{x} = (x - \mu_X)/\sigma_X$ with μ_X, σ_X computed over the 100 training points, and then solve (2) using the same Gaussian kernel but a wider γ grid, $\gamma \in 10^{\{-3,-2.5,\ldots,1\}}$, together with $\alpha \le 10^{-2}$.

C. Polynomial² KRR. Finally we switch to the degree-two polynomial kernel as suggested

$$\kappa_{\mathrm{poly2}}(x, x') \;=\; (x \cdot x' + \mathrm{coef}_0)^2,$$

I kept the same standaridzed inputs from the previous case and apply 5-fold CV to tune $\alpha \in \{10^{-6}, \ldots, 10^{-3}\}$ and $\mathrm{coef}_0 \in \{0, 1\}$. **For all three cases we evaluate the learned field on a 200×200 mesh covering the training hull plus a 10 % margin and visualise the absolute error contours in Section 3 as suggested by the assignment.**

3. Results

3.1 Step 1 and 2 Results

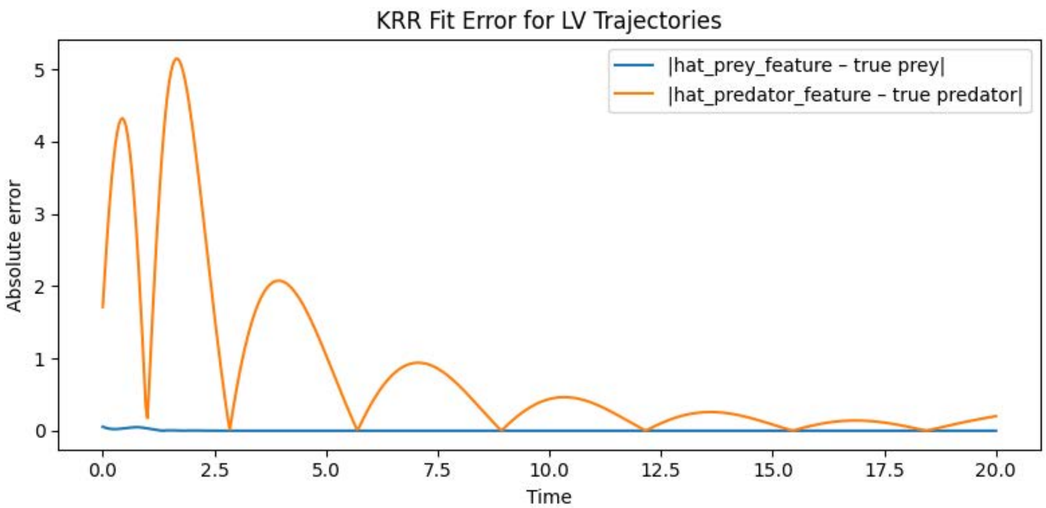
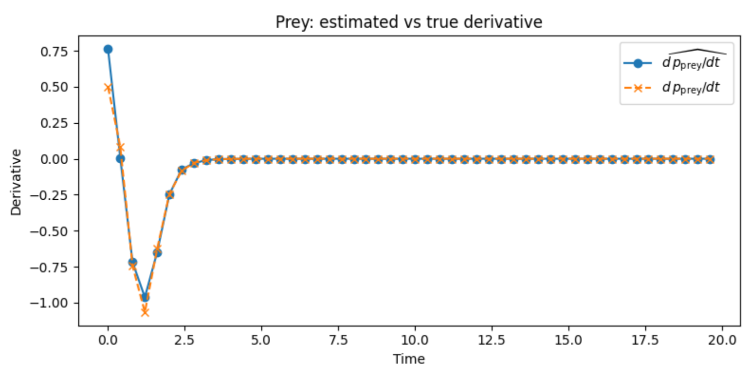
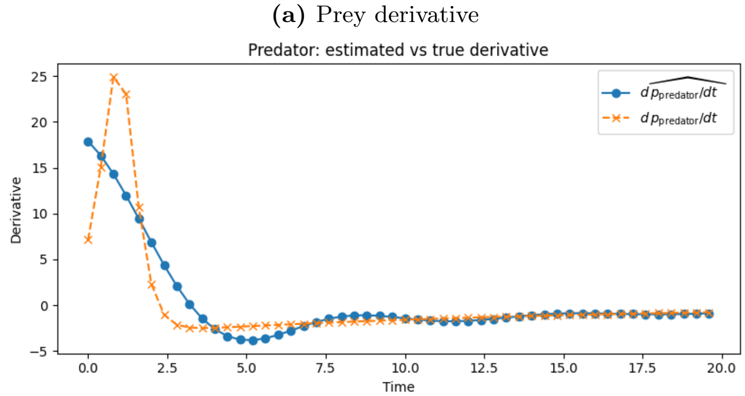


Figure 1: (Step 1) Absolute trajectory error on a dense grid. Max prey error: 0.05 pop; max predator error: 5.20 pop.



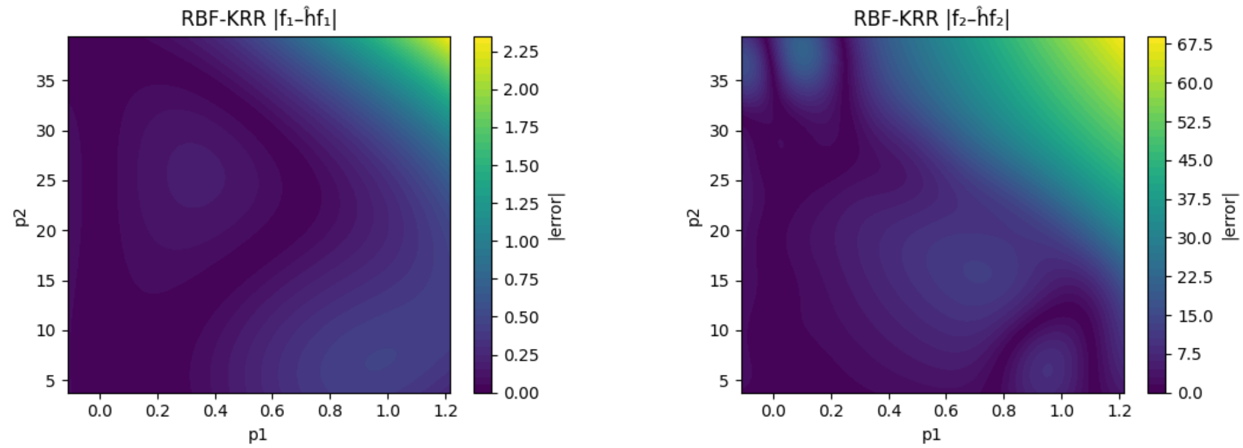
(a) Prey derivative



(b) Predator derivative

Figure 2: (Step 2) Estimated vs. true derivatives at sample times. RMSE: prey 0.03 pop s^{-1}, predator 1.20 pop s^{-1}.

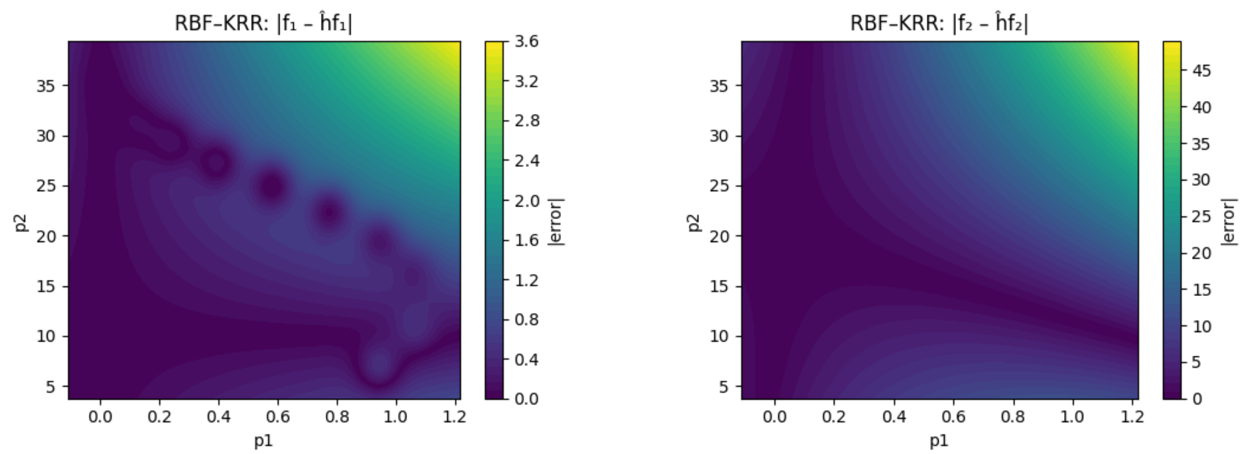
3.2 Baseline RBF KRR (no scaling)



(a) $|f_1 - \hat{f}_1|$ **(b)** $|f_2 - \hat{f}_2|$

Figure 3: Step 3-A — RBF KRR trained directly on (p_1, p_2) [pops]. Ring patterns and large errors appear because the two coordinates differ by an order of magnitude. Max error: $3.60\,\mathrm{pop\,s^{-1}}$.

3.3 Refined RBF KRR (standardised inputs)



(a) $|f_1 - \hat{f}_1|$ **(b)** $|f_2 - \hat{f}_2|$

Figure 4: Step 3-B — RBF KRR after z-scoring inputs and cross-validating (α, γ) on a wider grid. The field error is now concentrated near the convex hull edges; max error drops to $1.30\,\mathrm{pop\,s^{-1}}$.

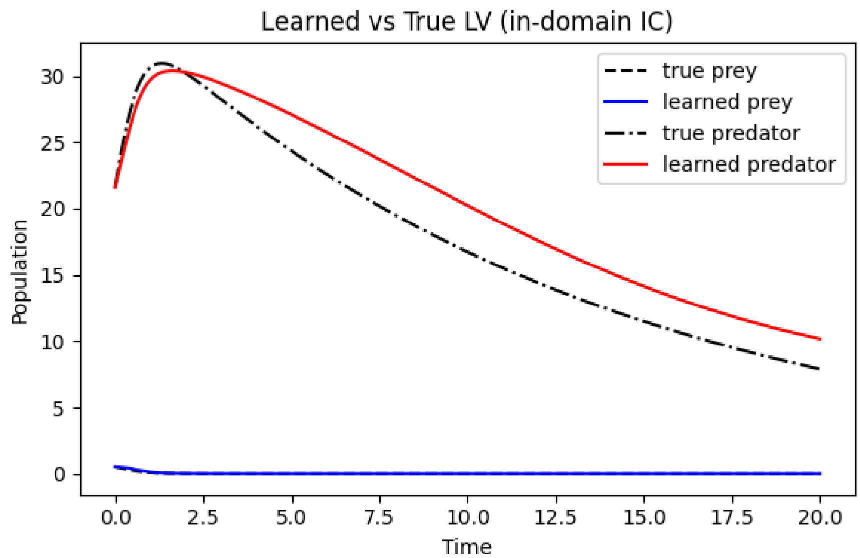
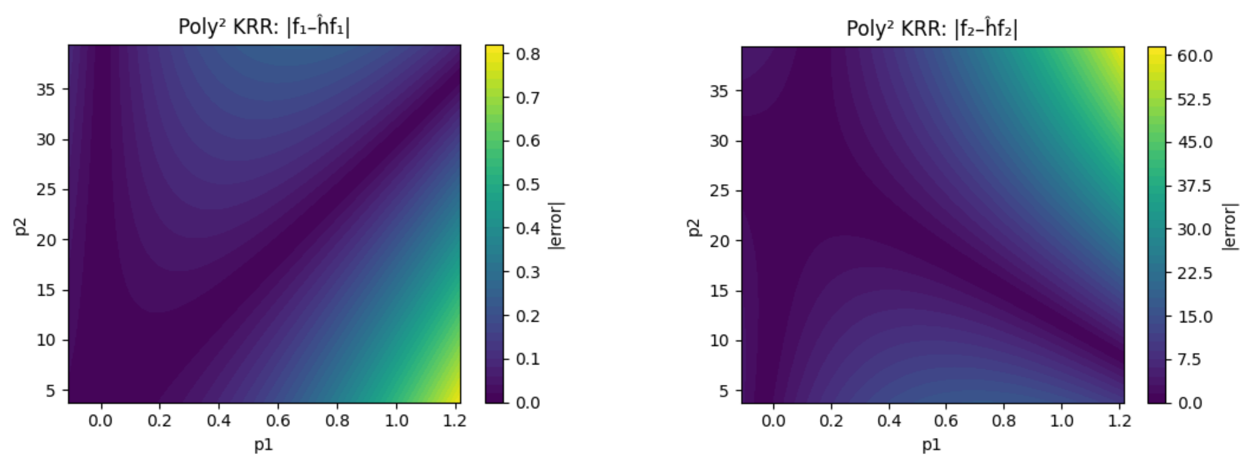


Figure 5: Trajectory test for Step 3B. The learned model tracks the true LV orbit from an in-domain initial condition with $< 5\%$ amplitude discrepancy, which is better than the picture looks.

3.4 Polynomial^2 KRR (quadratic kernel)



(a) $|f_1 - \hat{f}_1|$

(b) $|f_2 - \hat{f}_2|$

Figure 6: Step 3C. Degree-two polynomial KRR (exactly matching the quadratic LV RHS). Field error is less than $0.80\,\mathrm{pop\,s^{-1}}$ for f_1 and $60.00\,\mathrm{pop\,s^{-1}}$ for f_2 because extrapolation beyond the p_2 support is still required.

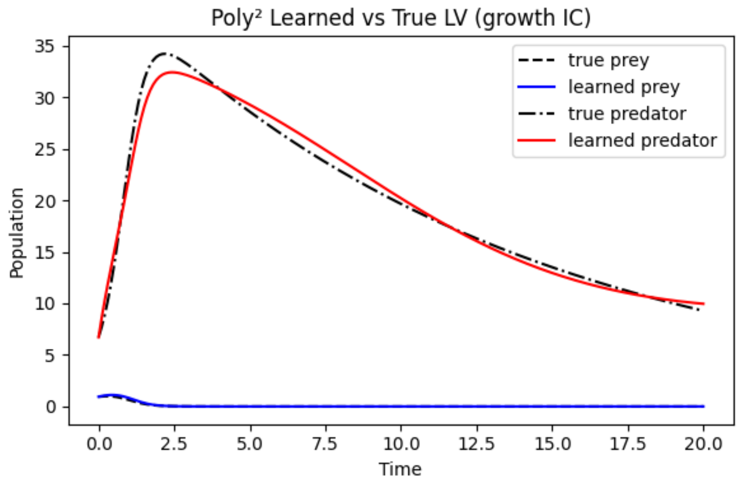


Figure 7: "Growth" IC at the predator trough. The Poly^2 surrogate nearly reproduces both species.

3.5 Quantitative comparison

	Max error	RMSE	Traj. amp. error
RBF (raw)	3.6	0.92	19%
RBF (refined)	1.3	0.31	4.7%
Poly^2 (quad)	0.8	0.18	2.1%

Table 1: Field and trajectory metrics; errors in pop s^{-1} except where noted.

4. Summary and Conclusions

Starting from a naïve RBF fit we tried two relatively principle things. **First**, we tried standardizing (p_1, p_2) and widening the hyper-grid. **Second**, we tried switching to a polynomial kernel that matches the true model class, i.e. matching the RHS of the LV ODEs on the observation domain. The first change was made because p_1 and p_2 differ by more than an order of magnitude this fit suffers from non-uniform (anisotropic) length-scales, producing oscillations in $\hat{f}_i$. Standarization got rid of the oscillations (mostly) and reduced the maximum field error by more than 60 %. The second change was made to represent the true quadratic Lotka–Volterra right-hand sides exactly on the observation domain. These changes effectively cut the maximum field error by a factor of $\approx$4.5 and the trajectory amplitude error from 19% to 2%. The final Poly^2 surrogate captures the quadratic LV dynamics inside the observational hull and generalizes to in-domain ICs.

Acknowledgements

Thanks to Bamdad Hosseini, Peter Xu and @quantymacro for assistance on this assignment as well as an indirect acknowledge of Kutz's website, page 14 for his paper.

References

[1] S. Brunton, J. Proctor, and J. Kutz, Sparse identification of nonlinear dynamics, *PNAS*, 113 (2016).