ACM 270

Inverse Problems and Data Assimilation: A Machine Learning Approach

Instructor: Eviatar Bach

- Email: ebach@caltech.edu

Instructor: Andrew Stuart

- Email: astuart@caltech.edu

TA: Yixuan (Roy) Wang

- Email: roywang@caltech.edu

Time and location: Tuesdays 9:00–10:30 and Thursdays 9:00–10:30 in ANB 213.

Office hours: Office hours for the instructors are by email appointment.
Office hours for the TA: Fri 3-4pm weekly starting from the second week, with the exception of Apr 26th; location: Steele House conference room.

Assessment: The course will be assessed through a group project (60%) and group presentations on research papers (40%); see Sections paper packets and projects for details. The group project will be evaluated via a written research report (c. 10 pages), due May 28th 12noon PST. The order of the group presentations will be assigned at random. Each group gets to choose its project and will be assigned a distinct set of paper packets, based on interests and under the coordination of the TA, by the end of the second week.

Lecture notes and supporting materials: Will be posted on Piazza.

Piazza: For all class-related discussions (in particular for Q/A).
Link: `https://piazza.com/caltech/spring2024/acm270`

Schedule:

- April 2nd (A) Ch12 Metrics and Distance-Like Measures
- April 4th (A) Ch13 Generative Modeling
- April 9th (A) Ch14 Supervised Learning
- April 11th (E) Ch16 Optimization
- April 16th (A) Ch1,2 Bayesian Inversion/Variational Inference

- April 18th (A) Ch3,4 Surrogate Modeling, Learning The Prior
- April 23th (A) Ch5,6 Transport to Posterior, Data Dependence
- April 25th (E) Ch7,8 Filtering and Smoothing, Learning The Model
- April 30th (E) Ch9,10 Learning Analysis Step, Learning The Filter
- May 2nd (E) Ch15,12 Time Series Forecasting, Using Forecasting Models,
- May 7th Student Presentations
- May 9th Student Presentations
- May 14th Student Presentations
- May 16th Student Presentations
- May 21st No Lecture/Project Consultation
- May 23rd Student Presentations
- May 28th Student Presentations
- May 30th No Class

Group Divisions

1. Calvello, Edoardo. Borcherds, Charlotte Helen. Cocke, Carter Karl. Yu, Jing. Schmitt, Julian Francis. Wang, Chuwei.
2. Berne, Alexander Clarence. Darcy, Sean Sebastian. Deng, Zijun. Li, Zongyi. Chen, Yishu. Viswesh, Annika Shanta.
3. Eichberger, Ethan Robert. Dey, Sreemanti. Liu, Yuan Kai. Luk, Enoch Chunlok. Raj, Mayank. Zhao, Brandon Yun.
4. Kaveh, Hojjat. Liu-Schiaffini, Miguel Omar. Liu, Shengduo (audit). Teegavarapu, Ritvik Sai. Darcy, Matthieu David. Wihardja, Adeline Emily.

Paper Packets

1. FourCastNet [16, 12, 20]
2. GraphCast and Pangu-Weather [21, 14, 26, 3]
3. Variational filtering [27, 24, 18, 8]
4. Alternative approaches to learning filters/controllers (connections to reinforcement learning and online learning) [11, 1, 25, 13]
5. Learning model/model error [5, 4, 15]
6. Variational inference with transports with applications in physics [22, 19, 23, 6]
7. Learning regularizers to solve inverse problems in imaging [17, 2, 7]

Projects

We recommend that the groups use Python for the computational projects, since the provided models and libraries all use Python.
Resources

- The code for the ML and numerical models described below is available at `https://github.com/eviatarbach/acm270_projects`.
- For the data assimilation problems, the Python DAPPER library provides nice implementations of many algorithms, and may be used.
- The Python library PyMC provides implementations of Markov chain Monte Carlo (MCMC) algorithms.
- Probabilistic scoring rules, including the continuous ranked probability score, are implemented in the Python properscoring package.
- `jax` is a Python library that allows one to autodifferentiate code, as well perform just-in-time compilation (allowing it to run much faster than regular Python code). The cd_dynamax library provides implementations of many data assimilation algorithms in `jax`. For the projects below that involve autodifferentiation, use of `jax` is recommended.

1. **Inverse problems with ML surrogate models**

   Description

   This project will involve solving inverse problems using ML surrogate models. This is motivated by the fact that the latter are often faster to run than the true forward model. They can hence significantly speed-up multi-query methods such as MCMC.

   The two models to be used will be

   - The 40-dimensional Lorenz96 model, a chaotic ODE meant to represent the dynamics of the midlatitude atmosphere. Both a numerical integrator of the ODE and an ML model are provided in the `lorenz96` folder. The ML model is described in [5].
   - The Darcy equation, describing flow through a porous medium. Here, a Fourier neural operator surrogate [16] is provided in the `darcy` folder.

   Project goals

   - Solve an inverse problem in the numerical Lorenz96 model. Take the forward model to be an integration of the Lorenz96 dynamics for 0.2 time units[1] from some initial state, and try to recover this initial state from a noisy observation of the state at the end of the interval. Obtain samples of the posterior using MCMC. Do this for both full observations $H = I$ and partial observations (observing every other location).
   - Repeat the same with the ML Lorenz96 model and compare the results, using the surrogate likelihood for MCMC.

[1] 0.2 time units leads to a non-trivial inverse problem but is still well under the mixing time.

- Solve the inverse problem of obtaining the permeability field given the pressure field for the Darcy equation. First, take a ground truth permeability field and add noise to the resulting pressure field (permeability–pressure pairs can be obtained as discussed below). Then, use the Darcy ML surrogate likelihood in MCMC. Compare the posterior you obtain with MCMC with the ground truth input.

Tips

- Look in `lorenz96/example.py` to see how to run the Lorenz96 numerical model and surrogate, and in `darcy/example.py` to see how to run the Darcy flow surrogate.
- The "true" numerical model for the Darcy flow is not provided, but `darcy/example.py` shows how to load ground truth input–output pairs, which you will use to generate the observations for the inverse problem.

Bonus

Implement a derivative-based MCMC sampler such as Hamiltonian Monte Carlo as opposed to random walk Metropolis.

2. **Ensemble data assimilation with ML model**

Description

This project will consist of implementing the ensemble Kalman filter using an ML forecast model. In particular, since the ML forecast is significantly less computationally expensive than the numerical one, large ensembles of the ML forecasts can be generated.

The project will make use of a two-layer quasigeostrophic model, a simplified form of the equations governing atmospheric flow. In the `qg` folder, we have made available both a numerical solver and an ML surrogate, which uses a U-Net architecture described in [9].

Project goals

- Produce a true trajectory and synthetic observations using the numerical model.
- Perform DA by applying an ensemble Kalman filter to the ML model.
- Compare the filtering performance using the ML model to filtering using the numerical model. The performance should be measured using by verifying against the true trajectory using one of the probabilistic scoring rules discussed in class (e.g., the continuous ranked probability score). Compare the performance using different ensemble sizes; in particular, using a larger ML ensemble than numerical ensemble.

Tips

- Look in `qg/example.py` to see how to run the numerical and ML models.
- It may be a good idea to test the filter with a simpler model first, such as Lorenz96 (also provided; see project 1).

- The models are configured such that each evaluation of the function corresponds to a forecast 1.0 time units into the future, which corresponds to about 6 hours on Earth's atmosphere. This is a good choice for the assimilation interval.
- Unless very large ensembles are used, localization will be required to have a stable filter.

Bonus

Perform data assimilation using both a physical ensemble and an ML ensemble using one of the multi-model or multifidelity methods described in class. This could possibly lead to a paper. Ask for more details if interested!

3. **Model error correction in ensemble data assimilation**

Description

This project focuses on learning a model error correction in data assimilation.

Here we will take the true system to be the two-scale Lorenz96 model, and the imperfect model to be the single-scale Lorenz96. The single-scale Lorenz96 will incur error due to the lack of the small-scale interactions.

Main project goals (80%)

- Produce a true trajectory and noisy observations using the two-scale Lorenz96.
- Assimilate those observations into the single-scale Lorenz96 using an ensemble Kalman filter.
- Train a neural network (your choice of architecture) to predict the analysis increment (analysis − forecast) from the forecast state.
- Run the EnKF again with the imperfect model, but add the neural network correction. Compare the filtering performance to that of the uncorrected model using one of the probabilistic scoring rules discussed in class.

Additional project goals (20%)

Instead of using analysis increments to learn the model error, use the autodifferentiable EnKF [10]. This will require jax. Compare the results to the correction learned by analysis increments.

Tips

- The single-scale and two-scale Lorenz96 models are provided in `lorenz96/numerical_model.`
- Due to the model error, assimilating observations of the two-scale Lorenz96 model into the single-scale version will require inflation to function well.
- The time-step 0.005 will work for integrating the two-scale Lorenz96 model with 4th-order Runge–Kutta.

References

[1] Haldun Balim et al. *Can Transformers Learn Optimal Filtering for Unknown Systems?* Aug. 16, 2023. DOI: `10.48550/arXiv.2308.08536`. arXiv: `2308.08536 [cs, eess]`. preprint.

[2] Martin Benning and Martin Burger. "Modern regularization methods for inverse problems". In: *Acta numerica* 27 (2018), pp. 1–111.

[3] Kaifeng Bi et al. "Accurate Medium-Range Global Weather Forecasting with 3D Neural Networks". In: *Nature* 619.7970 (July 2023), pp. 533–538. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06185-3.

[4] Marc Bocquet et al. "Bayesian Inference of Chaotic Dynamics by Merging Data Assimilation, Machine Learning and Expectation-Maximization". In: *Foundations of Data Science* 2.1 (2020), p. 55. DOI: 10.3934/fods.2020004.

[5] Julien Brajard et al. "Combining Data Assimilation and Machine Learning to Emulate a Dynamical Model from Sparse and Noisy Observations: A Case Study with the Lorenz 96 Model". In: *Journal of Computational Science* (June 20, 2020), p. 101171. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2020.101171.

[6] Johann Brehmer and Kyle Cranmer. "Simulation-based inference methods for particle physics". In: *Artificial Intelligence for High Energy Physics*. World Scientific, 2022, pp. 579–611.

[7] Luca Calatroni et al. "Bilevel approaches for learning of variational imaging models". In: *Variational Methods: In Imaging and Geometric Control* 18.252 (2017), p. 2.

[8] Andrew Campbell et al. "Online Variational Filtering and Parameter Learning". In: Advances in Neural Information Processing Systems. Nov. 9, 2021. URL: https://openreview.net/forum?id=et2st4Jqhc (visited on 02/08/2024).

[9] Ashesh Chattopadhyay et al. "Deep Learning-Enhanced Ensemble-Based Data Assimilation for High-Dimensional Nonlinear Dynamical Systems". In: *Journal of Computational Physics* 477 (Mar. 15, 2023), p. 111918. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2023.111918.

[10] Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. "Auto-Differentiable Ensemble Kalman Filters". July 19, 2021. arXiv: 2107.07687 [cs, stat]. URL: http://arxiv.org/abs/2107.07687 (visited on 04/20/2022).

[11] Claude-Nicolas Fiechter. "PAC Adaptive Control of Linear Systems". In: *Proceedings of the Tenth Annual Conference on Computational Learning Theory*. COLT '97. New York, NY, USA: Association for Computing Machinery, July 1, 1997, pp. 72–80. ISBN: 978-0-89791-891-6. DOI: 10.1145/267460.267481.

[12] John Guibas et al. "Efficient Token Mixing for Transformers via Adaptive Fourier Neural Operators". In: International Conference on Learning Representations. Oct. 6, 2021. URL: https://openreview.net/forum?id=EXHG-A3jlM (visited on 03/20/2024).

[13] Mohamad Abed El Rahman Hammoud et al. *Data Assimilation in Chaotic Systems Using Deep Reinforcement Learning*. Jan. 1, 2024. DOI: 10.48550/arXiv.2401.00916. arXiv: 2401.00916 [physics]. preprint.

[14] Remi Lam et al. "Learning Skillful Medium-Range Global Weather Forecasting". In: *Science* 382.6677 (Dec. 22, 2023), pp. 1416–1421. DOI: 10.1126/science.adi2336.

[15] Matthew Levine and Andrew Stuart. "A Framework for Machine Learning of Model Error in Dynamical Systems". In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10.

[16] Zongyi Li et al. "Fourier Neural Operator for Parametric Partial Differential Equations". In: International Conference on Learning Representations. Oct. 2, 2020. URL: https://openreview.net/forum?id=c8P9NQVtmnO (visited on 03/20/2024).

[17] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. "Adversarial regularizers in inverse problems". In: *Advances in neural information processing systems* 31 (2018).

[18] Joseph Marino, Milan Cvitkovic, and Yisong Yue. "A General Method for Amortizing Variational Filtering". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/hash/060afc8a563aaccd288f98b7c8723b61-Abstract.html (visited on 02/08/2024).

[19] George Papamakarios et al. "Normalizing flows for probabilistic modeling and inference". In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64.

[20] Jaideep Pathak et al. *FourCastNet: A Global Data-driven High-resolution Weather Model Using Adaptive Fourier Neural Operators*. Feb. 22, 2022. DOI: 10.48550/arXiv.2202.11214. arXiv: 2202.11214 [physics]. preprint.

[21] Tobias Pfaff et al. "Learning Mesh-Based Simulation with Graph Networks". In: International Conference on Learning Representations. Oct. 2, 2020. URL: https://openreview.net/forum?id=roNqYL0_XP (visited on 03/20/2024).

[22] Danilo Rezende and Shakir Mohamed. "Variational inference with normalizing flows". In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538.

[23] Ali Siahkoohi et al. "Preconditioned training of normalizing flows for variational inference in inverse problems". In: *arXiv preprint arXiv:2101.03709* (2021).

[24] Tobias Sutter, Arnab Ganguly, and Heinz Koeppl. "A Variational Approach to Path Estimation and Parameter Inference of Hidden Diffusion Processes". In: *Journal of Machine Learning Research* 17.190 (2016), pp. 1–37. ISSN: 1533-7928. URL: http://jmlr.org/papers/v17/16-075.html (visited on 11/16/2023).

[25] Anastasios Tsiamis and George J. Pappas. "Online Learning of the Kalman Filter With Logarithmic Regret". In: *IEEE Transactions on Automatic Control* 68.5 (May 2023), pp. 2774–2789. ISSN: 1558-2523. DOI: 10.1109/TAC.2022.3207670.

[26] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (visited on 03/20/2024).

[27] Michail D. Vrettas, Manfred Opper, and Dan Cornford. "Variational Mean-Field Algorithm for Efficient Inference in Large Systems of Stochastic Differential Equations". In: *Physical Review E* 91.1 (Jan. 30, 2015), p. 012148. DOI: 10.1103/PhysRevE.91.012148.