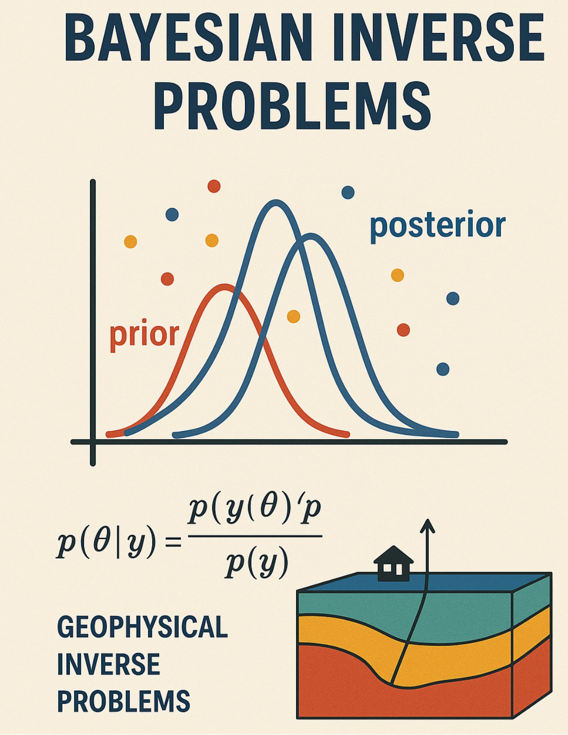
Inverse Problems and DA PRACTICAL

Mark Asch - MAKUTU/2025


BAYESIAN INVERSE PROBLEMS
posterior
prior
$p(\theta|y) = \frac{p(y(\theta)'p}{p(y)}$
GEOPHYSICAL INVERSE PROBLEMS

Practical for Lecture 01

Ex. 1—Ill-Posedness

Consider the highly nonlinear Duffing's equation,

$$\ddot{x} + 0.05\dot{x} + x^3 = 7.5 \cos t$$

with (true) initial state $x(0) = 3$ and $\dot{x}(0) = 4$. This equation exhibits high sensitivity to the initial conditions.

1. Solve the equation using a suitable ODE integrator over the interval $t \in [0, 50]$.

2. Show that two very closely spaced initial states lead to a large discrepancy in the trajectories:

- introduce an error of 0.03% - until when do we have an accurate forecast?

- introduce an error of 0.06% - until when do we have an accurate forecast?

Ex. 2—Deterministic Inversion: Adjoint Method

- See slides of DA on Adjoints, pp. 4–14.

Consider the convection-diffusion equation, where the diffusion coefficient is spatially varying. This model is close to many physical situations, where the medium is not homogeneous and we have zones with differing diffusive properties.

- The system is described by

$$\begin{cases} -\left(a(x)u'(x)\right)' - u'(x) = q(x), \quad 0 < x < 1, \\ u(0) = 0, \; u(1) = 0. \end{cases} \tag{1}$$

- Define the L_2-mismatch cost function as

$$J[a] = \frac{1}{2}\int_0^1 \left(u(x) - u^{\mathrm{obs}}(x)\right)^2 \,\mathrm{d}x, \tag{2}$$

where $u^{\mathrm{obs}}(x)$ denotes the observations on $[0,1]$.

- Objective: Derive the gradient using the Lagrangian (or variational formulation).

- Step 1: Let the cost function

$$J^*[a,p] = \frac{1}{2}\int_0^1 \left(u(x) - u^{\mathrm{obs}}(x)\right)^2 \,\mathrm{d}x + \int_0^1 p\left(-\left(au'\right)' - u' - q\right)\mathrm{d}x,$$

noting that

⇒ the second integral is zero when u is a solution of (1)

⇒ and that the adjoint variable, p, can be considered here to be a Lagrange multiplier function.

- Step 2: Take the variation of J^* with respect to its variables, a and p,

$$\begin{aligned}\delta J^{*} = & \int_0^1 \left(u - u^{\text{obs}}\right) \delta u \, \mathrm{d}x \\ & + \int_0^1 \delta p \overbrace{\Big(-\left(a u'\right)' - u' - q\Big)}^{=0} \mathrm{d}x \\ & + \int_0^1 p \left[\left(-\delta a \, u' - a \, \delta u' - \delta u\right)'\right].\end{aligned}$$

- Step 3: Integrate by parts, passing derivatives form u to p and define the adjoint equation and boundary conditions on p so as to obtain an integral expression for the variation δJ^*. Assume zero boundary conditions on the perturbation δu.

- Step 4: Based on the key result relating the variation to the gradient,

$$\delta J \doteq \nabla_{\mathbf{m}} J \delta \mathbf{m},$$

prove that the explicit expression for the gradient of J^* with respect to the unknown parameter a is given by

$$\nabla_{a(x)} J^* = u'p'.$$

- Conclusion: With

⇒ one solution of the direct system (1), plus

⇒ one solution of the adjoint system for p, we recover the gradient of the cost function with respect to the sought for diffusion coefficient, $a(x)$.

⇒ This gradient can then be used to find (numerically) the optimal function $a(x)$ that minimizes J by a suitable descent algorithm, usually by a quasi-Newton method.

Ex. 3: Adjoint method—constant parameter case

- The code adj_inv.py solves a constant-valued parameter estimation inverse problem

  $$-bu''(x) + cu'(x) = f(x))$$

  with $u(0) = u(1) = 0$ and the cost function (2).

- Create a fully-documented notebook based on this code, with:

  ⇒ formulation of the problem;

  ⇒ presentation of the adjoint equation and the expression for the cost function and its gradient.

- Run the code and compare the accuracy of the inversion for:

  ⇒ varying noise levels, starting from zero;

⇒ varying initial guesses for b and c;
⇒ tuning of the L-BFGS optimizer.

- Draw detailed conclusions regarding the complexity (difficulties) of the inversion.

Ex. 4: Adjoint method—variable parameter case

In Exercise 2 above, we derived the adjoint state and cost function gradient for the convection-diffusion ordinary differential equation, with a spatially varying diffusion coefficient.

- The code `adj_inv_cx.py` solves a slightly simpler version of the variable parameter estimation inverse problem, where

$$-u'' + c(x)u' = f(x)$$

with $u(0) = u(1) = 0$ and the cost function (2).

- Create a fully-documented notebook based on this code, with

⇒ formulation of the problem;

- ⇒ presentation of the adjoint equation and the expression for the cost function and its gradient;
- ⇒ explanation of the regularization strategy.

- Run the code and compare the accuracy of the inversion for
  - ⇒ varying noise levels, starting from zero;
  - ⇒ varying initial guesses for $c(x)$;
  - ⇒ different (simpler) functions for the spatially-varying coefficient $c(x)$.

- Draw detailed conclusions regarding the complexity (difficulties) of the inversion.

Ex. 5: Adjoint method—Linear PDE

- The natural extension of the ordinary differential equations seen above is the initial-boundary-value problem known as the diffusion equation,

$$\frac{\partial u}{\partial t} - \nabla \cdot (\nu \nabla u) = 0, \qquad x \in (0, L), \quad t > 0,$$
$$u(x, 0) = u_0(x), \qquad u(0, t) = 0, \quad u(L, t) = \eta(t).$$

- This equation has multiple origins emanating from different physical situations.

  ⇒ The most common application is particle diffusion, where u is a concentration and ν is a diffusion coefficient.

  ⇒ Then there is heat diffusion, for which u is a temperature and ν is a thermal conductivity.

  ⇒ The equation is also found in finance, being closely related to the Black-Scholes model.

- ⇒ Another important application is population dynamics.
- ⇒ These diverse application fields, and hence the diffusion equation, give rise to a number of inverse and data assimilation problems.

- A variety of different controls can be applied to this system:

  - ⇒ *internal* control, $\nu(x)$: this is the parameter identification problem, also known as tomography;
  - ⇒ *initial* control, $\xi(x) = u_0(x)$: this is a source detection IP or DA problem;
  - ⇒ *boundary* control, $\eta(t) = u(L,t)$: this is the "classical" boundary control problem, also a parameter identification IP.

- As above, we can define the mismatch/L2 *cost function*,

$$J[\nu,\xi,\eta] = \frac{1}{LT}\int_0^T\int_0^L \left(u - u^{\mathrm{obs}}\right)^2 \,\mathrm{d}x\,\mathrm{d}t,$$

which is now a space-time multiple integral, and its related LAGRANGIAN,

$$J^* = \frac{1}{LT}\int_0^T\int_0^L \left(u - u^{\mathrm{obs}}\right)^2 \,\mathrm{d}x\,\mathrm{d}t \\ + \frac{1}{LT}\int_0^T\int_0^L p\left[u_t - (\nu u_x)_x\right]\,\mathrm{d}x\,\mathrm{d}t.$$

- Now take the variation of J^*,

$$\begin{aligned}\delta J^* \quad = \quad & \frac{1}{LT}\int_0^T\int_0^L 2(u - u^{\mathrm{obs}})\delta u \,\mathrm{d}x\,\mathrm{d}t \\ & + \frac{1}{LT}\int_0^T\int_0^L \delta p \overbrace{\left[u_t - (\nu u_x)_x\right]}^{=0} \,\mathrm{d}x\,\mathrm{d}t \\ & + \frac{1}{LT}\int_0^T\int_0^L p\left[\delta u_t - (\delta\nu\, u_x + \nu\delta u_x)_x\right]\,\mathrm{d}x\,\mathrm{d}t,\end{aligned}$$

and perform integration by parts to obtain

$$\begin{aligned}\delta J^* = &\frac{1}{LT}\int_0^T\int_0^L \delta\nu\, u_x p_x \,\mathrm{d}x\,\mathrm{d}t \qquad (3)\\ &-\frac{1}{LT}\int_0^L p\ \delta u|_{t=0}\,\mathrm{d}x\\ &+\frac{1}{LT}\int_0^T p\ \delta\eta|_{x=L}\,\mathrm{d}t,\end{aligned}$$

where we have defined the adjoint equation as

$$\begin{aligned}&\frac{\partial p}{\partial t} + \nabla\cdot(\nu\nabla p) = 2(u-u^{\mathrm{obs}}), \quad x\in(0,L), \quad t>0\\ &p(0,t)=0, \quad p(L,t)=0,\\ &p(x,T)=0.\end{aligned}$$

- As before, this equation is of the same type as the original diffusion equation, but must be solved backwards in time.

- Finally, from (3) we can pick off each of the three desired terms of the gradient,

$$\begin{aligned} \nabla_{\nu(x)} J^* &= \frac{1}{T}\int_0^T u_x p_x \,\mathrm{d}t, \\ \nabla_{u|_{t=0}} J^* &= -\left.p\right|_{t=0}, \\ \nabla_{\eta|_{x=L}} J^* &= \left.p\right|_{x=L}. \end{aligned}$$

- **Conclusion**:

  ⇒ Once again, at the expense of a single (backward) solution of the adjoint equation, we obtain explicit expressions for the gradient of the cost function with respect to each of the three control variables.

  ⇒ This is quite remarkable and completely avoids “brute force” or exhaustive minimization, though, as mentioned earlier, we only have the guarantee to find a local minimum.

  ⇒ However, if we have a good starting guess that is usually obtained from historical or other “physical" knowledge of the system, we are sure to arrive at a good (or at least, better) minimum.

- **Exercise**: go through the calculations, step-by-step, and reproduce the results.

Ex. 6: Adjoint method—Nonlinear PDE

We study here Burgers' Equation (a simplified, but realistic model for Navier-Stokes) with control of the initial condition and boundary control.

- Viscous Burgers equation in the interval $x \in [0, L]$ is defined by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f,$$

$$u(0,t) = \psi_1(t), \quad u(L,t) = \psi_2(t),$$

$$u(x,0) = u_0(x).$$

- The control vector

$$(u_0(x), \psi_1(t), \psi_2(t)).$$

- The cost function is taken as

$$J(u_0, \psi_1, \psi_2) = \frac{1}{2} \int_0^T \int_0^L \left(u - u^{\mathrm{obs}}\right)^2 dx\, dt.$$

- The adjoint model is

$$\begin{gathered}
\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + \nu \frac{\partial^2 p}{\partial x^2} = u^{\mathrm{obs}} - u, \\
p(0,t) = 0, \quad p(L,t) = 0, \\
p(x,T) = 0.
\end{gathered}$$

- Gradient/variation/directional derivative of J is finally,

$$\begin{aligned}
\hat{J}\left[u_0, \psi_1, \psi_2\right](\delta_u, \delta_1, \delta_2) = & - \int_0^L \delta_u p(x,0)\, dx \\
& + \int_0^T \nu \delta_2 \frac{\partial p}{\partial x}(L,t) \\
& - \nu \delta_1 \frac{\partial p}{\partial x}(0,t)\, dt
\end{aligned}$$

that gives,

$$\begin{aligned} \nabla_{u_0} J &= -p(x, t=0) \\ \nabla_{\psi_1} J &= -\nu \frac{\partial p}{\partial x}(x=0, t) \\ \nabla_{\psi_2} J &= \nu \frac{\partial p}{\partial x}(x=L, t). \end{aligned}$$

- These explicit gradients enable us to solve inverse problems for

⇒ the initial condition, which is a data assimilation problem;

⇒ or for the boundary conditions, which is an optimal boundary control problem;

⇒ or for both.

- **Exercise**: Consider a parameter identification problem for the viscosity ν. Derive the adjoint and gradient expressions for this case.

DA Codes

Various open-source repositories and codes are available for both academic and operational data assimilation.

1. DARC: `https://research.reading.ac.uk/met-darc/` from Reading, UK.

2. DAPPER: `https://github.com/nansencenter/DAPPER` from Nansen, Norway.

3. DART: `https://dart.ucar.edu/` from NCAR, US, specialized in ensemble DA.

4. OpenDA: `https://www.openda.org/`.

5. Verdandi: `http://verdandi.sourceforge.net/` from INRIA, France.

6. PyDA: `https://github.com/Shady-Ahmed/PyDA`, a Python implementation for academic use.

7. Filterpy: `https://github.com/rlabbe/filterpy`, dedicated to KF variants.

8. EnKF; `https://enkf.nersc.no/`, the original Ensemble KF from Geir Evensen.