Inverse Problems 52506, Project work: Inverse problems of generalized projections

Mikko Kaasalainen, Tampere University of Technology

October 13, 2009

Abstract

We determine the surface map of a body in $\mathbb{R}^3$ from the measurements of the volumes of its generalized projections in various geometries. The inverse problem is posed as an integral equation for which a stable solution is sought with various methods.

1 Setup

We consider a convex polyhedron $\mathcal{B}$ with given vertices and their circuits for facets, i.e., we know the areas and normals of the facets. The shape can be anything convex, e.g., an approximated sphere (see below). Our unknowns are the darkness values of the facets: $0 < g_i \leq 1$, $i = 1, \ldots, M$. From $g \in \mathbb{R}^M$, the observations $L \in \mathbb{R}^N$ are, in $(\omega_0, \omega) \in S^2 \times S^2$ (see lecture notes)

$$L(\omega_0, \omega) = A(\mathcal{B}; \omega_0, \omega) g, \tag{1}$$

where the matrix A is given by the shape of $\mathcal{B}$ and the geometry:

$$A_{ij} = S_j \left(\mu^{(ij)}, \mu_0^{(ij)} \right) \sigma_j, \tag{2}$$

where S_j and σ_j are the scattering model and surface area of facet j; $\mu^{(ij)} = \omega_i \cdot \nu_j$ and $\mu_0^{(ij)} = \omega_{0i} \cdot \nu_j$, where ω_i and ω_{0i} are the viewing and illumination directions for the observation i, and $\nu_j \in S^2$ is the unit outward normal of facet j. For $\mu^{(ij)} \leq 0$ or $\mu_0^{(ij)} \leq 0$, $A_{ij} = 0$ due to convexity.

2 Basic linear case

Find the minimum of

$$\|L - Ag\|_2^2 = \sum_{i=1}^{N} \left(L_i - \sum_{j=1}^{M} A_{ij} g_j \right)^2. \tag{3}$$

The most basic linear solution is obtained from

$$(A^T A)g = A^T L. \tag{4}$$

Add random noise: $L \to L + \epsilon$ and try again. Can you stabilize with SVD: write

$$A = UWV^T, \tag{5}$$

and then, in

$$A^{-1} = V[\mathrm{diag}(1/w_j)]U^T \tag{6}$$

set suitable diagonal elements to zero. With a stabilized matrix

$$\widetilde{A} = \mathrm{diag}(w_j)_{j \le M'}, \quad M' \le M, \tag{7}$$

we have

$$g = \sum_{j=1}^{M'} \left(\frac{U_{(j)} \cdot L}{w_j} \right) V_{(j)} \tag{8}$$

(where $U_{(j)}, V_{(j)}$ are the columns j of the matrices U, V as vectors).

2.1 Statistical inversion: Bayes inference

If we assume Gaussian distributions, our linear model yields a simple maximum likelihood estimate $\hat{g}$ for the posterior distribution. Let our a priori distribution be concentrated around g_0 (with tightness and correlation given by the matrix Σ_0):

$$p_{pr}(g) \sim \exp[-\frac{1}{2}(g - g_0)^T \Sigma_0^{-1}(g - g_0)]. \tag{9}$$

On the other hand, the conditional distribution of the measurement is

$$p(L|g) \sim \exp[-\frac{1}{2}(L - Ag)^T \Sigma_1^{-1}(L - Ag)], \tag{10}$$

where Σ_1 is the covariance matrix of the measurements ($\Sigma_{ij} = \langle \epsilon_i \epsilon_j \rangle$). Then Bayes says

$$p(g|L) \sim p_{pr}(g)p(L|g) \tag{11}$$

so we obtain

$$p(g|L) \sim \exp\{-\frac{1}{2}[g^T Q g - 2g^T Q Q^{-1}(\Sigma_0^{-1} g_0 + A^T \Sigma_1^{-1} L) + ...]\} \tag{12}$$

where the Fisher information matrix is

$$Q = (\Sigma_0^{-1} + A^T \Sigma_1^{-1} A). \tag{13}$$

Thus our posterior distribution is the exponential quadratic form

$$p(g|L) \sim \exp[-\frac{1}{2}(g - \hat{g})^T Q (g - \hat{g})] \tag{14}$$

with the centre of the posterior distribution $\hat{g}$ at

$$\hat{g} = Q^{-1}(\Sigma_0^{-1} g_0 + A^T \Sigma_1^{-1} L). \tag{15}$$

The formal error estimates (squared) for the parameters $\hat{g}_j$ are the diagonal elements of Q^{-1}. If the a priori part is uniform, we simply get the old LSQ normal equations.

3 Nonlinear form, positivity constraint, regularization

Instead of g, we can solve for $a \in \mathbb{R}^M$ as the parameters in

$$g_j = e^{a_j} \tag{16}$$

to ensure positivity. Levenberg-Marquardt optimization (see appendix) is useful for this, as now our objective function (3) is of the nonlinear least-squares form

$$\chi^2 = \sum_{i=1}^{N} [f(a; (\omega_0, \omega)_i) - L_i]^2. \tag{17}$$

Simple regularization functions $R(g)$ for

$$g = \arg\min\{\chi^2 + \lambda R(g)\} \tag{18}$$

can be added in pseudo-χ^2 form (so we can use L-M), e.g.,

$$R(g) = \sum_{i=1}^{M} (g_i/\langle g \rangle - 1)^2. \tag{19}$$

This can be written in a form emphasizing local smoothness (rather than global uniformity) a la Tikhonov.

4 Project work

Fix the shape $\mathcal{B}$. E.g., a convex polyhedron approximating a sphere can be created by octant triangulation (see lecture notes) or simply by making a latitude-longitude grid and connecting the grid points with lines instead of curves (a "disco ball"). Then the area of a facet is

$$\sigma_j = 2c_j R^2 \sin\Delta\theta \, \sin\frac{\Delta\varphi}{2} \, \sin(\theta_j - \frac{\Delta\theta}{2}), \tag{20}$$

where R is the sphere radius,

$$c_j = \sqrt{1 - \sin^2\frac{\Delta\varphi}{2} \, \cos^2(\theta_j - \frac{\Delta\theta}{2})}, \tag{21}$$

and $\Delta\theta$, $\Delta\varphi$ are the latitude- and longitude intervals and θ_j the "southernmost" latitude of the facet (θ increases towards "south"). The spherical coordinates $\nu = (\theta_n, \varphi_n)$ of the direction of the facet's outward normal are

$$\cos\theta_n = \frac{1}{c_j} \cos\frac{\Delta\varphi}{2} \, \cos(\theta_j - \frac{\Delta\theta}{2}), \tag{22}$$

$$\varphi_n = \varphi_j + \frac{\Delta\varphi}{2}, \tag{23}$$

where φ_j is the smaller longitude of the side edges of the facet.

For creating the original maps, give a value $0 < g_j \leq 1$ for each facet (and plot the maps of your worlds). You can write a black letter over a white surface, make a random map, use a smooth function, etc. Also choose a scattering law, e.g., a linear combination of

$$S_{LS} = \frac{\mu\mu_0}{\mu + \mu_0}, \quad S_L = \mu\mu_0. \tag{24}$$

Generate L at a number of observing geometries in $S^2 \times S^2$ using various noise levels. In particular, create series of so-called lightcurve observations at sets of $\omega = (\vartheta, 0 \leq \psi_i \leq 2\pi), i = 1, \ldots, n$ (corresponding to typical time series). Use these data to investigate the inverse problem of reconstructing the world map from L, especially in the light of the theoretical results discussed in the lecture notes. Remember to avoid inverse crime (typically, use a sparser grid for inversion, or a set of basis functions $L^2(S^2)$ with which the data are not created).

The project work report (≥ 7 pages) should be organized in the form of a scientific paper, including at least the following sections:

- A short introduction and background
- Your problem and simulation setup
- Methods and results
- Analysis of results, conclusions, discussion
- References

Use any software packages and programming languages you like. Use your imagination: just about all "typical" theoretical and practical aspects of inverse problem solving are encountered in this problem. Approach this as something studied for the first time: what would we like to know? What approaches should we try? The nominal maximum points for the project work apply to "standard good work", but the number of possible extra points is unlimited. The project can be done in groups of ≤ 3 people. Plot your solutions vs. original map and model fits vs. data.

4.1 Suggestions and comments

1. Simulate the effects of the number and measurement space coverage of data: what happens with a small set of L, what if $\omega_0 \to \omega$ etc.

2. How small must the noise level be for details of a given size to be discernible in the data (this can be initially checked already via the simulation of the direct problem)? Compare with a rough estimate from the theoretical Laplace series coefficients (see lecture notes: l, m of the Y_l^m-term depict a corresponding angular size on S^2).

3. Start with the linear form, check the condition number of your system matrix, see if SVD helps, and so on.

4. Study the effect of systematic errors: incorrect scattering law, somewhat incorrect ω_0, ω, etc.

5. Blind test: exchange data sets with your collaborators – is inversion harder now (inverse crime means that you use prior info you are not expected to have in real life).

6. Try the Bayesian approach: use some expected g distribution for the prior (uniform grayscale, expected letter somewhere, correlation between some facets etc.)

7. Use the nonlinear form and L-M to guarantee $g > 0$. Is this more stable? Do you need regularization functions R?

8. Use a fully analytical model for ellipsoids of uniform $g = 1$ and geometric scattering law $S = \mu$ to create your data:

$$L = \pi abc\Big[\sqrt{\omega^T M\omega} + \frac{\omega^T M\omega_0}{\sqrt{\omega_0^T M\omega_0}}\Big], \tag{25}$$

where

$$M = \begin{pmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{pmatrix}, \tag{26}$$

and a, b, c are the semiaxis lengths of the ellipsoids (with c-axis parallel to the z-coordinate axis). How well can you fit the data with your discretized model? Do you need smoothness regularization?

9. Use some (smooth) positive function on S^2 to create the values g_i for constructing simulated data. Are smoothness regularization functions useful? Compare with the letter case. What happens with different discretization levels of the inversion model?

10. Use the coefficients of a Laplace (spherical harmonics) $L^2(S^2)$ series as the unknown parameters (linear if direct series, nonlinear if exponential – see lecture notes). With this choice, you can investigate smooth functions on a dense facet grid with considerably fewer parameters than facets.

11. How well can different letters as world maps be distinguished? What happens if you use another letter as a priori function?

12. Since we know from theory that $S^2 \times S^2$ data are more than sufficient for the determination of g_j and we can in fact extract information on the scattering law S, try it. Add, e.g., the relative weight of S_{LS} vs. S_L to the set of unknown parameters (a nonlinear problem). Can you determine the weight from the data?

13. For real enthusiasts: add rotation parameters to the set of unknowns using the rotation matrix equations in the lecture notes (the data are lightcurves generated for various geometries with a fixed rotation axis direction and rotation speed). This is nonlinear even if you use the linear form for g_j, so use L-M optimization. Or use MCMC to plot the a posteriori distribution of the rotation parameters (anyone who gets this far will get top marks for the course).

Appendix: Levenberg-Marquardt optimization

Let us approximate the nonlinear χ^2 function

$$\chi^2(a) = \sum_{i=1}^{N} [f(a; (\omega_0, \omega)_i) - L_i]^2, \tag{27}$$

$a \in \mathbb{R}^M$, by

$$\chi^2(a) \approx C - d \cdot a + \frac{1}{2} a^T D a, \tag{28}$$

with $d \in \mathbb{R}^M$ and D an $M \times M$ matrix. If the approximation is already a good one (we are near the minimum of χ^2), we hit the optimal a_{min} by jumping from the current (iterated) a_{cur} with ($\nabla := \partial/\partial a$)

$$a_{min} = a_{cur} + D^{-1}[-\nabla\chi^2(a_{cur})]. \tag{29}$$

On the other hand, if we are not near the minimum, we could take the steepest descent- type step down the gradient direction:

$$a_{next} = a_{cur} - c\nabla\chi^2(a_{cur}). \tag{30}$$

Now

$$\frac{\partial\chi^2}{\partial a_k} = -2\sum_i [f(a; (\omega_0, \omega)_i) - L_i]\frac{\partial f}{\partial a_k}, \tag{31}$$

and

$$\frac{\partial\chi^2}{\partial a_k \partial a_l} = 2\sum_i [\frac{\partial f_i}{\partial a_k}\frac{\partial f_i}{\partial a_l} - \mathcal{O}(\epsilon)], \tag{32}$$

where we ignore the term of the order of noise level ϵ because the uncorrelated $f(a; (\omega_0, \omega)_i) - L_i \sim \pm\epsilon$ contributions tend to cancel each other out. Writing

$$\beta_k := -\frac{1}{2}\frac{\partial\chi^2}{\partial a_k}, \quad \alpha_{kl} := \frac{1}{2}\frac{\partial\chi^2}{\partial a_k \partial a_l}, \tag{33}$$

and the above approximation

$$\alpha_{kl} = \sum_i \frac{\partial f_i}{\partial a_k}\frac{\partial f_i}{\partial a_l}, \tag{34}$$

our "single step to a minimum" can be written as

$$\sum_l \alpha_{kl} \delta a_l = \beta_k, \tag{35}$$

to be solved for the step δa_l. Likewise, the steepest descent reads

$$\delta_l = c\beta_l. \tag{36}$$

Levenberg-Marquardt combines these two smoothly by using a "fudge" factor λ such that the steepest descent step is given by

$$\delta a_l = \frac{1}{\lambda \alpha_{ll}} \beta_l, \tag{37}$$

and defining

$$\alpha'_{jj} := \alpha_{jj}(1+\lambda), \quad \alpha'_{jk} = \alpha_{jk}, \quad j \neq k, \tag{38}$$

so the whole thing can be written as

$$\sum_{l=1}^{M} \alpha'_{kl} \delta a_l = \beta_k. \tag{39}$$

If $\lambda >> 1$, we have the steepest descent domain, while $\lambda \to 0$ takes us to the "single step to minimum" regime.

In practice, the Levenberg-Marquardt algorithm is simply:

1. Compute $\chi^2(a_0)$ for the initial guess a_0.
2. Choose a small λ, e.g., $\lambda = 10^{-3}$.
3. Solve (39) for δa and evaluate $\chi^2(a + \delta a)$.
4. If $\chi^2(a + \delta a) \geq \chi^2(a)$, increase $\lambda \to 10\lambda$ and go to 3.
5. If $\chi^2(a + \delta a) < \chi^2(a)$, decrease $\lambda \to \lambda/10$, update $a \to a + \delta a$ and go to 3.

Stop if the convergence of $\chi^2(a)$ has saturated.

L-M is best for a moderate number of unknowns (say, less than 500) as it stores the Hessian matrix (or its approximation). For a large number of unknowns (or functions not writable in χ^2-form) it is better to use, e.g., conjugate gradient or other methods that work with an M-vector instead of an $M \times M$ matrix.