

Learning regularization functionals - A supervised training approach

E. Haber* and L. Tenorio†

December 31, 2002

Abstract

We consider the solution of a distributed parameter estimation problem where the data are contaminated by noise. A common approach to solve such a problem is to use Tichonov style regularization, however, it is not always clear what type of regularization penalty should be used for a given problem as different regularization operators yield very different solutions. Here we use supervised learning techniques to estimate the regularization functional given a training set of feasible solutions. Our approach leads to a constraint optimization problem that we solve using inexact SQP type methods. We illustrate the methodology with two examples.

1 Introduction

We consider the problem of recovering an unknown function m given indirect noisy observations b (usually discrete) whose connection to m is

$$F[m] + \epsilon = b, \quad (1)$$

where F is a linear or nonlinear ('forward') operator and ϵ represents the noise.

Recovering m given the data (1) is typically an ill-posed problem. Among all the possible ways to obtain a meaningful solution \hat{m} to (1) [8, 20, 19, 17], we work within the framework of constrained optimization of a model functional $R(m)$ where \hat{m} is a solution of an optimization problem

$$\begin{aligned} \hat{m} &= \operatorname{argmin} R(m) \\ \text{s.t.} \quad & \|F[m] - b\|^2 \leq \tau^2, \end{aligned} \quad (2)$$

*Departments of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA. (haber@mathcs.emory.edu).

†Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, CO 80401, USA. (ltenorio@mines.edu).

or in the context of generalized Tikhonov regularization

$$\hat{m} = \operatorname{argmin} \|F[m] - b\|^2 + R(m). \quad (3)$$

(the Tikhonov regularization parameter has been included in $R(m)$). In either case, the basic idea is to restrict the set of a priori plausible models by choosing a functional $R(m)$ that penalizes deviations from what we a priori define as desirable model characteristics. These characteristics can be learned from prior available models.

For example, in a Bayesian regularization approach one assumes a Gaussian distribution for the errors and a Gibbs prior probability density function (PDF) for the discretized models

$$\text{PDF}(m) = \frac{1}{Z(R)} e^{-R(m)}, \quad (4)$$

so that (3) is the MAP estimate for the posterior distribution [26, 27]. The functional R in the prior PDF can be estimated from a set of available training models. However, in a proper Bayesian framework the prior model distribution (prior information) and the forward problem (regularization) are separate issues. In practice the regularization functional depends on the forward operator; it cannot be defined only by prior model information. It is not clear either what type of correlation structure is introduced into the posterior by a prior that is just chosen to provide the correct MAP estimate. Furthermore, MAP estimates depend on the required model discretization and the chosen discretization of the regularization operator. Unlike Tichonov regularization, it is not clear that different discretizations yield MAP estimates that are the same (up to discretization errors). Finally, to avoid the partition function $Z(R)$ in (4), which often requires computationally demanding calculations of multi-dimensional integrals, one has to make further, sometimes dubious, approximations through mean field calculations Monte-Carlo integration or pseudo-likelihood [12, 25].

In this paper we present an alternative penalized regularization approach that avoids the selection of a model prior distribution; we determine an ‘optimal’ regularization functional based on regularization arguments and supervised learning. We still assume that enough information about the noise is available, so that there is a known constant τ such that $\|\epsilon\|^2 \leq \tau^2$ for the deterministic case or that, for random noise, the PDF is known with $E\|\epsilon\|^2 = \tau^2$. We also assume the availability of a family of K training models m_T^1, \dots, m_T^K chosen by an “expert” that we use to learn a functional that regularizes the inverse problem.

The idea of supervised learning has been used in many practical problems where an expert can train a system to infer the connection between two sets of variables (see for example [14, 9, 15] and reference therein). However, to our knowledge, it has not been used to find an optimal regularization operator in inverse problems.

In principle the form of $R(m)$ can depend on location \mathbf{x} and involve derivatives or other filters. For example, $R(m)$ may be of the form

$$R(m) = \rho_0(\mathbf{x}, m) + \rho_1(\mathbf{x}, \partial_x m, \partial_y m) + \rho_2(\mathbf{x}, \partial_{xx} m, \partial_{xy} m, \partial_{yy} m) + \dots, \quad (5)$$

where $\rho_0(\mathbf{x}, t)$, $\rho_1(\mathbf{x}, t_1, t_2)$ and $\rho_2(\mathbf{x}, t_1, t_2, t_3)$ are nonnegative functions that depend on \mathbf{x} and on derivatives of the model. Obviously, this representation may depend on too many argument functions and thus we need to impose some restrictions on the space of permissible functionals. We will assume that R belongs to a class of functionals chosen based on prior model information. This class could be defined, for example, by model characteristics such as smoothness [17], entropy [11], edge enhancement [22], or by a family of nonnegative functionals parameterized by a vector θ . For example:

- i. A depth dependent regularization for potential fields problems introduced by Li & Oldenburg [13] is based on a regularization functional of the form

$$R(m; \theta) = \int \left[\theta_1 \left(\frac{\partial m}{\partial x} \right)^2 + \theta_1 z^{-\theta_2} \left(\frac{\partial m}{\partial z} \right)^2 \right] dx dz, \quad (6)$$

which imposes a higher penalty on surface features. Training models can be used to determine optimal values of $\theta = (\theta_1, \theta_2)$.

- ii. The functional in (i) is a particular case of the more general functional

$$R(m; \theta) = \|\theta(x, y) \nabla m(x, y)\|^2,$$

where θ is a function (distributed parameter) that controls the smoothness of the model at different locations.

- iii. In anisotropic diffusion regularization [22], regularization functionals have the form

$$R(|\nabla m|) = \int \theta(|\nabla m|) dV, \quad (7)$$

where $\theta(t)$ is a real function. Our goal will be to determine an appropriate such function. Note that (7) is homogeneous because the function θ does not directly depend on location (x, y) .

The rest of the paper is organized as follows. In Section 2 we describe the procedure to determine a regularization functional based on training models; its numerical implementation is discussed in Section 3. In Section 4 we provide some examples of geophysical applications to show the improvement in model estimates that we can obtain by learning the regularization functional. We close with a brief discussion in Section 5.

2 Learning regularization functionals

We start by describing a procedure to learn the regularization functional $R(m)$ from a family of training models $\{m_T^1, \dots, m_T^K\}$. This functional will then be used to find

a solution of the inverse problem (3). Note R can be chosen to minimize the error $\|m - \hat{m}\|^2$ of the model estimate instead of the prediction error $\|F[m] - F[\hat{m}]\|^2$, as it is done with cross-validation.

Our basic assumption is that the training and sought models are similar in the sense that they share the same “optimal” regularization functional. For example, consider first the case when the data are noiseless

$$F[m] = b \tag{8}$$

and F is invertible with a stable inverse, then one can recover m exactly: $m = G[b] = F^{-1}[b]$. There is no need for regularization and the same G can be used to recover any m in the domain of F .

When F is not injective, such as when F is a linear function from \mathbb{R}^s into \mathbb{R}^l and $s > l$, there is no unique model that solves (8); we need prior information to restrict the space of permissible models. This is not really a regularization but rather a stabilization process¹. In this case Tichonov stabilization is equivalent to the constrained optimization problem

$$\begin{aligned} \hat{m} &= \operatorname{argmin} R(m) \\ \text{s.t.} \quad &F[m] - b = 0, \end{aligned} \tag{9}$$

where R is a model penalty that the desired solution should minimize. The solution \hat{m} of the optimization problem (9) (assuming that R has been chosen so that a unique solution exists) defines a transformation G_R from the data space to the model space \mathcal{M} : $\hat{m} = G_R[b]$. Since the estimate \hat{m} of the unknown model m satisfies this constraint, it follows that $G_R[F(\hat{m})] = \hat{m}$. In addition, if R is a regularization functional that correctly complements the active space of the forward operator, then we also have $G_R[F(m)] = m$. We thus define a family \mathcal{M}_R of models that can be recovered exactly with G_R :

$$\mathcal{M}_R = \{ m \in \mathcal{M} \mid G_R[F(m)] = m \}. \tag{10}$$

For example, take a discrete model m , $R(m) = \|m\|^2$ and $F[m] = Jm$ for some full rank $s \times l$ matrix J with $s < l$. In this case \mathcal{M}_R is just the orthogonal complement of the nullspace of J ; any model without a component in $\operatorname{Null}(J)$ can be recovered exactly with the norm functional. If instead we use $R(m) = \|Lm\|^2$, then it is easy to verify, through the generalized SVD of J and L , that

$$\operatorname{Null}(F)^\perp \cap \operatorname{Null}(L^t L) \subset \mathcal{M}_R.$$

If the training models m_T^i are to be used to determine an appropriate penalty functional, we must assume that they can all be estimated with the same, albeit

¹By stabilization we mean that the solution of the problem fits the data exactly, which is not necessarily the case in regularization.

unknown, R . We therefore assume that there is a functional $R(m)$ and a tolerance constant $\eta \geq 0$ such that

$$m_T^i \in \mathcal{M}_R(\eta) = \{ m \in \mathcal{M} \mid \| G_R[F(m)] - m \| \leq \eta \}; \quad i = 1, \dots, K. \quad (11)$$

The set $\mathcal{M}_R(\eta)$ reduces to \mathcal{M}_R as $\eta \rightarrow 0$. The reason we introduce η in (11) is that it is unlikely that the “expert” can choose models that exactly belong to the same family as defined in (10). The parameter η represent the “noise” in the training models. Thus, although we assumed noiseless observed data, the training set is assumed to be noisy.

Given assumption (11), we propose to find an approximation \hat{R} to the functional R by minimizing an average of the residuals

$$\hat{R} = \operatorname{argmin} \frac{1}{2} \sum_i \| G_R[F(m_T^i)] - m_T^i \|^2. \quad (12)$$

In practice we assume that R belongs to a family of functions parameterized by a vector θ ; the goal is to find θ that minimizes (12).

2.1 The noisy case

Assume now that the operator F is singular and/or the data are noisy. In this case, the optimization problem has to regularize the inverse problem as well as to separate signal from noise. We modify definitions (10) and (11) to account for uncertainties in the data and in the functional R : Given a fixed functional $R(m)$ and data of the form

$$b = F[m] + \epsilon, \quad \mathbb{E} \|\epsilon\|^2 = \tau^2 \quad (13)$$

for some $\tau > 0$, we define the family of models

$$\mathcal{M}_R(\eta, \tau) = \{ m \in \mathcal{M} \mid \mathbb{E} \| G_R[F(m) + \epsilon] - m \|^2 \leq \eta^2, \mathbb{E} \|\epsilon\|^2 = \tau^2 \} \quad (14)$$

for some $\eta = \eta(\tau)$ that depends on τ and R . Better training sets and functionals R will lead to smaller η . Note that $\mathcal{M}_R(\eta, \tau)$ reduces to $\mathcal{M}_R(\eta)$ as $\tau \rightarrow 0$. In most applications the expected misfit $\eta(\tau)$ is not known a priori but can be estimated using techniques such as cross-validation. We will return to this point in Section 3.

The assumption we make on the training models is similar to that made in the noiseless case but this time we have to account for noise in the data: For a given noise level τ , we assume that there is a function $R(m)$ and a constant $\eta \geq 0$ such that

$$m_T^i \in \mathcal{M}_R(\eta, \tau); \quad i = 1, \dots, K. \quad (15)$$

We use the training models together with simulated noise ϵ^i to generate K data vectors b_T^i through the forward problem (13). Then, we solve the K optimization problems

$$\hat{m}_T^i = G_R[b_T^i] = \operatorname{argmin} \| F[m] - b_T^i \|^2 + R(m); \quad i = 1, \dots, K. \quad (16)$$

A good choice of training models and penalty functional $R(m)$ should lead to small errors $E \|m_T^i - \hat{m}_T^i\|^2$. Finding an optimal regularization operator is thus reduced to a supervised learning problem: Given the training pairs $\{m_T^i, b_T^i\}$ ($i = 1, \dots, K$), find a regularization operator R such that the function G_R provides good estimates of the training models. The goodness of the estimates will be measured in an average sense

$$\begin{aligned} \hat{R} &= \operatorname{argmin} \frac{1}{2} \sum_i \|m_T^i - \hat{m}_T^i(R)\|^2 \\ \text{s.t.} \quad &R \in \mathcal{R}, \end{aligned} \tag{17}$$

where, to avoid over-fitting, we restrict R to be in a family of models \mathcal{R} . In practice, the optimization (17) is regularized by either choosing a family of functionals parameterized by a low dimensional vector θ , or by imposing smoothness constraints on the functionals. We provide some examples in Section 4, but the selection of a good parameterization is outside the scope of this paper; the reader is referred to [27] for a discussion of this question in the Bayesian framework. For the deterministic case, [5] discusses the selection of differential operators as penalty functionals for deconvolution problems.

2.2 Discussion

We have reduced the set of possible regularization operators to a specific parametric family \mathcal{R}_θ . This is a kind of subspace regularization [8] that works well only if the selection of the subspace is adequate to the problem. The functionals $R(m; \theta)$ of this family have to be complex enough to properly model the training set as well as the connection between the unknown and training models. On the other hand, if $R(m; \theta)$ is too flexible, then the training models are overfit and G_R performs poorly on models that are not in the original training set. In this case we need to regularize the procedure of estimating θ . These problems are typical to any supervised learning framework (see for example [9, 15, 14, 24]) We will discuss how to overcome these problems by the usual techniques of regularization, testing and cross-validation.

The proposed procedure for learning regularization operators is hardly new to the inverse problem practitioner. Most regularization operators are chosen with these exact ideas in mind; one starts with a reasonable synthetic test example and tailors the regularization so that the optimization procedure “works” for the synthetic data. For example, the regularization operators discussed in [13] in the context of gravity and magnetics applications are based on similar arguments. In medical imaging the Shepp-Logan model [10] is often used to test new regularization algorithms and to tailor them so that they perform well on this model.

As mentioned in the introduction, our approach is fundamentally different from Bayesian regularization where the regularization functional is determined by the prior distribution of the models; regularizations of completely different forward operators rely on the same prior model distribution. However, it is not hard to find practical

examples where the choice of good regularization operators strongly depends on the forward problem. We provide some examples in Section 4.

By learning the regularization functional from training data, the regularization is tailored to a specific forward operator and noise level.

3 Numerical solution of the optimization problem

If the functional $R(m) = R(m, \theta)$ belongs to a parametric family \mathcal{R}_θ , a regularized version of (17) is

$$\hat{\theta} = \operatorname{argmin} \sum_i \frac{1}{2} \|m_T^i - \hat{m}_T^i(\theta)\|^2 + \gamma r(\theta) \quad (18a)$$

$$\text{s.t. } \hat{m}_T^i(\theta) = \operatorname{argmin} \frac{1}{2} \|F[m] - b_T^i\|^2 + R(m; \theta), \quad i = 1, \dots, K; \quad (18b)$$

where $r(\theta)$ is a regularization functional for θ and γ is a regularization parameter. We now briefly discuss how to find a solution to this optimization problem.

To simplify notation, we will write $m_i(\theta)$ instead of $\hat{m}_T^i(\theta)$. For simplicity we will assume that F is linear but similar techniques can be used with nonlinear operators (although this requires higher order derivatives and therefore is more involved). We assume that the models have been discretized on some grid h and that the operator F is a $n \times d$ matrix J . A necessary condition for local extrema of the constraints (18b) is

$$g(m_i) = J^T (J m_i - b_T^i) + \nabla_m R(m_i; \theta) = 0. \quad (19)$$

The standard procedure in supervised learning is to eliminate this constraint and solve an unconstrained optimization problem by plugging (18b) into (18a). This approach is very expensive in our context because (18b) may require the solution of the nonlinear equation (19). We therefore take an alternative approach based on constraint optimization techniques similar to those in [6]. We use a sequential quadratic programming (SQP) type approach [16] to solve for θ and m_i simultaneously.

The Lagrangian associated with the optimization problem is

$$\mathcal{L} = \sum_i \frac{1}{2} \|m_T^i - m_i\|^2 + \sum_i \lambda_i^T [J^T (J m_i - b_T^i) + \nabla_m R(m_i; \theta)] + \gamma r(\theta), \quad (20)$$

where the gradient is with respect to the model and λ_i are Lagrange multipliers. Differentiating with respect to m_i, θ and λ_i we obtain

$$\mathcal{L}_{\lambda_i} = J^T (J m_i - b_T^i) + \nabla_m R(m_i; \theta) = 0 \quad (21a)$$

$$\mathcal{L}_{m_i} = m_i - m_T^i + J^T J \lambda_i + H_R(m_i; \theta) \lambda_i = 0 \quad (21b)$$

$$\mathcal{L}_\theta = \sum_i \left(\frac{\partial \nabla_m R(m_i; \theta)}{\partial \theta} \right)^T \lambda_i + \gamma r'(\theta) = 0, \quad (21c)$$

where $H_R = \partial^2 R / \partial m^2$ is the matrix of second derivatives of R with respect to the model variables. System (21) is a discrete nonlinear system for (m_i, θ, λ_i) that we solve using inexact all-at-once methods developed in [7, 6].

To avoid a solution of (21) that overfits the data, we use a training-validating approach [9, 24]. We divide the training set into two groups: $\{m_T^{tr_i}\}$ and $\{m_T^{cv_i}\}$ are, respectively, the validating and training sets used to determine θ . The optimization problem (18) is solved a few times with different values of the regularization parameter γ . We start with large γ and slowly decrease it. For each value of γ we check both the training misfit

$$\phi_{tr} = \sum \|m_T^{tr_i} - m_i(\theta)\|^2$$

and the validating set misfit

$$\phi_{cv} = \sum \|m_T^{cv_i} - m_i(\theta)\|^2.$$

Both the training and the testing misfit decrease simultaneously when the problem is over-determined. However, if the problem of finding θ is ill-posed, then at some stage the training misfit will continue to decrease while the validating misfit increases. At this stage one stops the iteration to avoid over-fitting.

4 Examples

In this section we consider three applications where training models are used to learn a regularization functional. We start with simple one-dimensional models penalized by a functional $R(m; \theta)$ that belongs to a parametric family to illustrate how different models have similar optimal regularization functional.

4.1 Example I - A parametric 1D problem

The data are defined by

$$b_k = J_k m + \epsilon_k = \int_0^{2\pi} e^{-kx/2} \cos(kx) m(x) dx + \epsilon_k, \quad (22)$$

where $\{\epsilon_k\}$ is uncorrelated zero mean Gaussian noise of standard deviation 1%. This example arises as the first iteration in a Gauss-Newton method applied to a 1D magnetotelluric problem [23]

The models are generated so that they are continuous on the interval $[0, 2\pi]$ but with twice the frequency in the interval $[\pi, 2\pi]$; four examples are shown in Figure (1). We assume that these models belong to the same family $\mathcal{M}_R(\eta, \tau)$ for some functional R and tolerance η . We use a regularization functional that allows different degrees of smoothness in the two intervals

$$R(m; \theta) = \int_0^\pi \theta_1 \left(\frac{dm}{dx}\right)^2 dx + \int_\pi^{2\pi} \theta_2 \left(\frac{dm}{dx}\right)^2 dx,$$

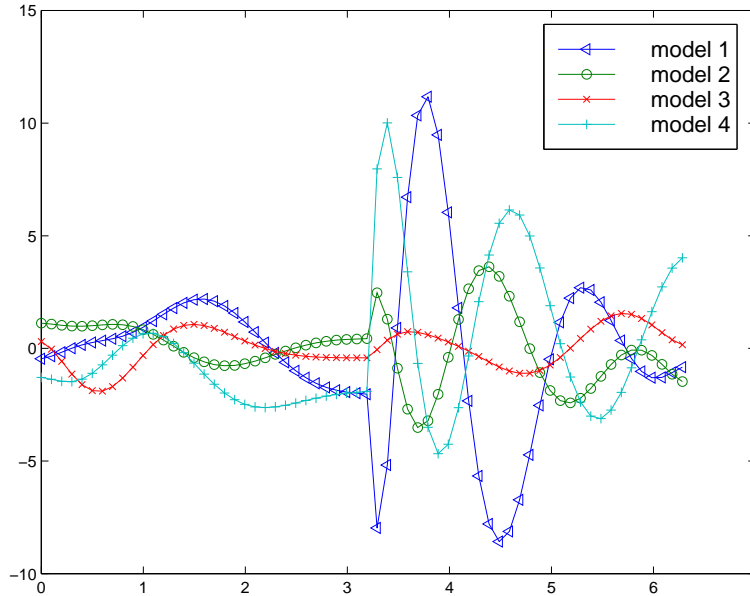


Figure 1: Four examples of the 1D models used in the forward problem (22).

where $\theta_1, \theta_2 \geq 0$. Figure 2 shows contour plots of the errors $\phi_{ij} = \|m_T - \hat{m}_T(\theta_1^i, \theta_2^j)\|^2$ for each the four training models shown in Figure 1. The four maps are similar with an optimal value of $\theta \approx (0.1, 0.0001)$ and a relative error $\eta/\tau \approx 0.43$, so that the four models belong to $\mathcal{M}(\eta \approx 0.43\tau, \tau)$.

4.2 Example II - Comparison to the MAP estimate

For our second example we use the same 1D forward problem (22) with 1% noise but with different regularization functional and types of models. As a regularization functional we use

$$R(m; \theta) = \int_0^{2\pi} \left(e^{\theta(x)/2} \frac{dm}{dx} \right)^2 dx. \quad (23)$$

This type of regularization operators has been used in [13] to compensate for the lack of resolution in different parts of the model. Here we show that they can be learned from training data.

The models m are realizations from the multivariate Gaussian distribution $N(0, C)$, where $C = -(d^2/dx^2)_h^{-1}$ is a discrete inverse Laplacian with the condition that the mean of m is zero so as to make C invertible. The MAP estimate corresponding to this prior is

$$\hat{m}_{\text{map}} = (J^T J + \tau^2 C^{-1})^{-1} J^T b. \quad (24)$$

We generated five models from this distribution; the four that will be used to determine θ are shown in Figure 3. This θ will be used to recover a fifth model shown in Figure 5.

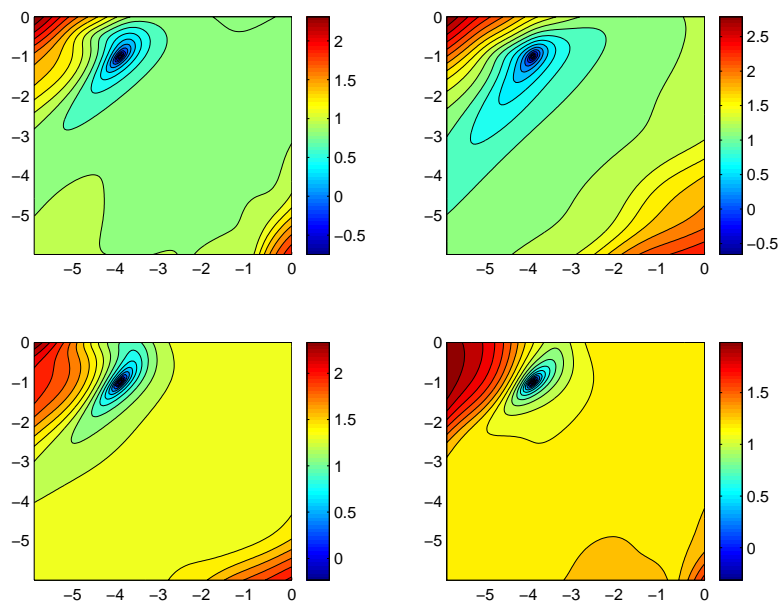


Figure 2: The difference $\|m_T - \hat{m}(\theta_1^i, \theta_2^j)\|^2$ as a function of $\log \theta_1$ and $\log \theta_2$ for model 1 (top left), model 2 (top right), model 3 (bottom left) and model 4 (bottom right), which are shown in Figure 1.

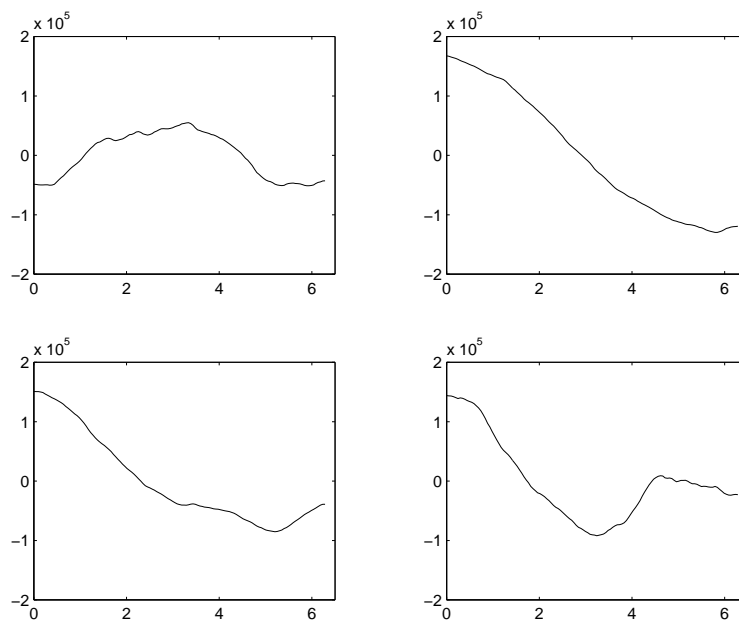


Figure 3: Synthetic 1D models from the distribution $N(0, C)$ used in Example II.

We first discretize θ on a grid. Assuming m is discretized in cell centers x_i ($i = 1, \dots, n$), the operator dm/dx is simply discretized as

$$\frac{dm}{dx} \approx h^{-1}(m_{i+1} - m_i).$$

This approximation is centered at $x_{i+\frac{1}{2}}$ and therefore θ is discretized at the nodes $x_{i+\frac{1}{2}}$ ($i = 1, \dots, n - 1$). The discrete regularization operator can be written as

$$R_h(m; \theta) = \frac{1}{h} \sum_i \exp(\theta_{i+\frac{1}{2}}) (m_{i+1} - m_i)^2. \quad (25)$$

Our goal is to estimate the $n - 1$ components of θ but since we only have four training models, we regularize the problem as in (18a) by imposing a discrete smoothness constraint on θ ; instead of (17) we use (18a)

$$\hat{\theta} = \operatorname{argmin} \frac{1}{2} \sum_i \|m_T^i - \hat{m}_i(\theta)\|^2 + \gamma \left\| \left(\frac{d}{dx} \right)_h \theta \right\|^2, \quad (26)$$

where $(d/dx)_h$ is the discrete derivative operator that operates on the nodes and γ is a regularization parameter which we determine using cross validation. The estimate $\hat{\theta}$ and the fit to the fifth model are shown in Figures 4 and 5, respectively. Figure 5 also shows the MAP estimate. It is clear that the model we obtain by learning the regularization operator does much better than the MAP estimate for large x where little information is provided by the forward operator. In this region the Bayes estimate is the zero prior mean. This is no surprise as the learned regularization operator has been tailored to the specific forward problem; it complements the forward operator and tends to recover more of the model structure for large x . The average relative errors $\|m - \hat{m}\|/\|m\|$ over the four training models, were, respectively, 0.41 and 0.59 for our reconstruction and the MAP estimate. This shows that learning the regularization functional can significantly improve the reconstruction.

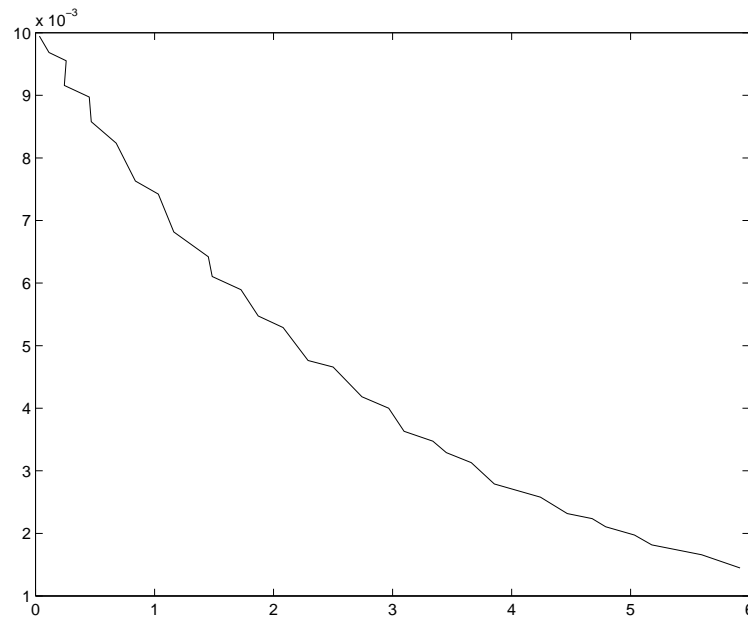


Figure 4: The estimated function $\hat{\theta}(x)$ learned from the models shown in Figure 3 through equation (26).

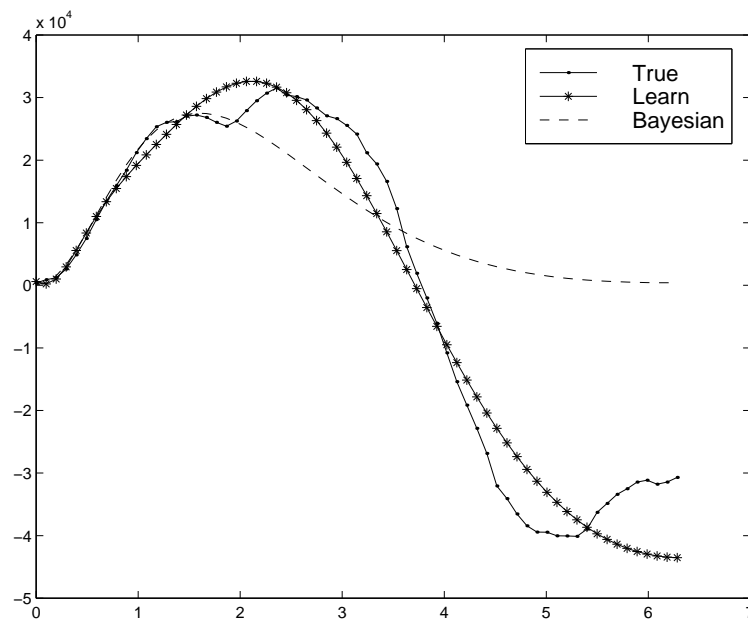


Figure 5: The solid line with asterisks is the reconstruction of a 1D model (solid line) using the optimal θ shown in Figure 4. The dashed line shows the MAP estimate (24).

4.3 Example III - Anisotropic diffusion in 2D

In our third example we learn regularization operators for a 2D borehole narrow angle ray tomography experiment. In this experiment, sources are placed in one borehole and receivers in another. For each source-receiver pair the travel time along the ray is

$$b = \int_{\text{ray}} m(x, y) dl, \quad (27)$$

where $m(x, y)$ is the slowness (inverse velocity) of the medium at depth y and horizontal displacement x . This connection between the model and the travel time is an example of the well studied limited angle Radon transform. To compute the travel time we use a straight-ray approximation.

To allow discontinuities in the slowness function we use a regularization operator of the form

$$R(m) = \iint \theta(|\nabla m|) dx dy, \quad (28)$$

where $\theta(z)$ is a symmetric positive function and

$$|\nabla m| = \sqrt{m_x^2 + m_y^2}.$$

(the partials are in the sense of distributions.) In earlier applications of this type of regularization functionals in geophysics [4, 3], θ was typically the absolute value function (or an approximation thereof). This method has been rediscovered and adopted to PDE based image processing (see for example [22, 1, 12, 18, 2]). In the last few years numerous choices of the function $\theta(z)$ have been proposed (see for example [22]). Here we determine θ using training models.

For the slowness model we use the rough and smooth “Marmousi” test problems², which are semi-real geophysical sections typically used in geophysical applications to test inversion procedures. Each model is made of 192×61 cells (we assume that each cell is $1m \times 1m$). To obtain training models, we divide each of the Marmousi models into two; the left half is used for learning and the other half for the inverse problem. The training and “true” models are shown in Figure 6. Even though we have only one model for learning, the problem is not too ill-posed because θ will be a spline parameterization of a 1D function.

To generate the synthetic data using the true models, we place 61 sources on the left end of the model and 61 receivers on the right end. The travel times are then determined using (27) and perturbed by adding 5% noise. This set up provides $61^2 = 3721$ data points and $61 \times 96 = 5856$ model parameters. The data are shown in Figure 7. Note that the data are very similar; indeed, most of the difference between the two data sets is below the noise level and thus without a-priori information both models are equally good solutions of the inverse problem.

²<http://www-rocq.inria.fr/~benamou/testproblem.html>

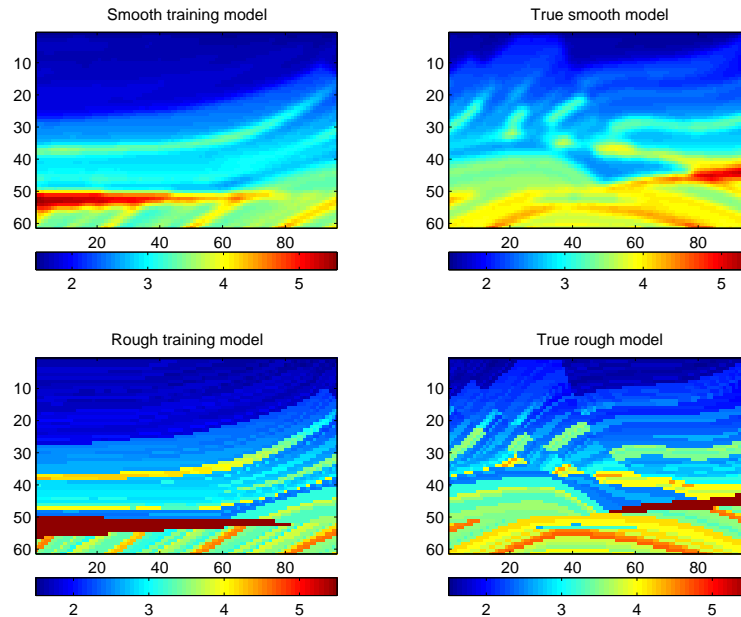


Figure 6: The Marmousi models. The smooth Marmousi (top) and the rough Marmousi (bottom). We have used the models on the left to learn the regularization functional and recover the models on the right.

To determine an optimal function θ we first assume that $\theta(z)$ is a symmetric function ($\theta(z) = \theta(-z)$) and parameterize it using natural cubic splines discretized by 32 points on the interval $[0, 7]$. Define the vector $\boldsymbol{\theta} = [\theta(0), \theta(z_1), \dots, \theta(z_n)]^T$, where z_i are the collocation points and z_n is chosen large enough. We have used the natural boundary conditions for both sides of the spline which implies that $\theta'(z) = \text{const}$ for $z > z(n)$. Given a point larger than $z(n)$, these boundary conditions yield functions that grow linearly for large z and therefore reduce the effects of large gradients caused, for example, by edges in the model.

The spline parametrizations leads to the following approximation of the regularization operator

$$R(m; \boldsymbol{\theta}) \approx e^T A (|\nabla_h m|^{(\eta)}) \boldsymbol{\theta}$$

$$|\nabla_h m|_{ij}^{(\eta)} = h^{-1} \sqrt{(m_{i+1,j} - m_{i,j})^2 + (m_{i,j} - m_{i,j+1})^2 + \delta^2},$$

where A is a sparse interpolation matrix that depends on the values of $|\nabla_h m|$, and $e = [1, \dots, 1]^T$. As it is commonly done in TV regularization (see [21]), we have included a small parameter δ to make the problem differentiable where the gradient vanishes. We have chosen $\delta = 0.01$ which is small enough compared to the values of the gradients.

Given the lack of additional training models, we have not implemented a validating criteria but included a penalty for non-smooth θ by adding a regularization term of the form $\|d/dz\theta\|^2$. This is especially important for the tomography example because we need the derivatives of $\theta(z)$ to solve the inverse problem through the optimization

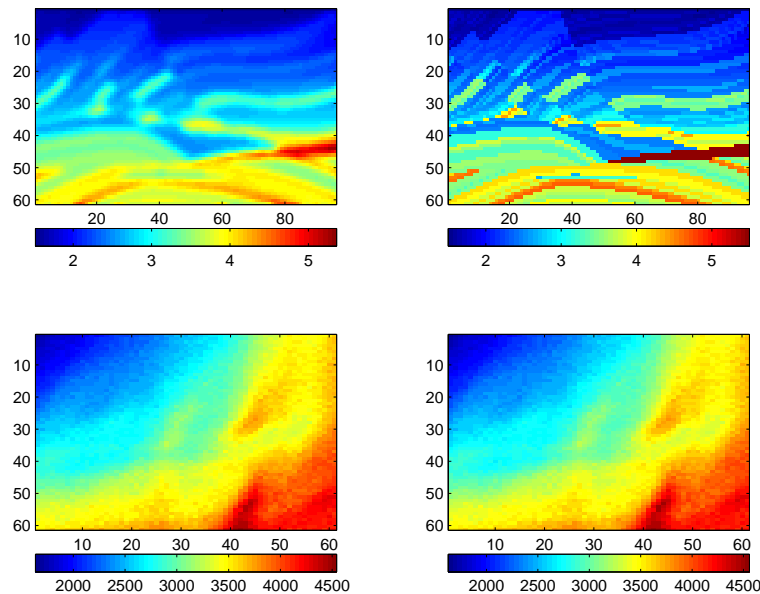


Figure 7: The Marmousi training models (top) and the data (bottom). The data are plotted as source location (x axis) vs receiver location (y-axis).

problem (3). Highly oscillatory $\theta(z)$ may trap inversion estimates in local minima; we thus stop the training process when θ is no longer smooth. This is done by simple visual inspection.

The regularization operators obtained for the soft and hard Marmousi learning models are shown in Figure 8. Their corresponding optimal regularization functionals are very different: The soft model regularization resembles the function z^2 while the hard model, which has rougher structures, leads to a regularization with smaller penalization on large gradients.

The optimal regularization functionals were used to estimate the true models (the second half of the Marmousi models) by solving the optimization problem (3). We have used lagged diffusivity [21] to solve the equations. For comparison, Figure 10 shows Tikhonov regularization estimates of the models obtained with the regularization functionals $\beta\|\nabla m\|^2$ where β is chosen such that the misfit is equivalent to the misfit of the trained inversions.

We emphasize that all the reconstructions fit the data to the same extent and thus without prior information there is no reason to prefer one model estimate over another. However, given appropriate information, it is evident that learning the regularization functional yields better model estimates.

5 Summary

We have presented a methodology to find regularized solutions of inverse problems by learning regularization functionals from training models m_T^i . The final model estimate

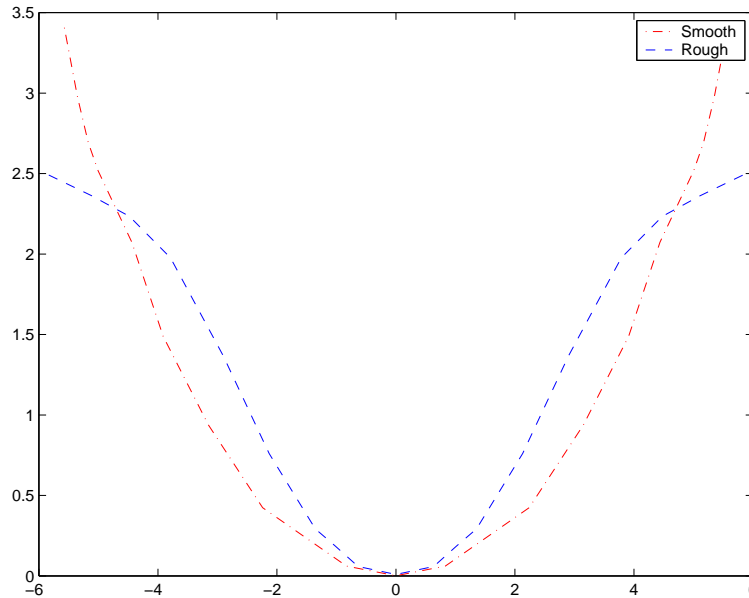


Figure 8: The two different regularization functional obtained for the smooth (dash-dot) and rough (dashed) Marmousi models.

\hat{m}_R given data (1) is obtained as the result of a sequence of regularizations. In the first regularization we use the training models to find a regularization functional R that minimizes the average residual errors of the Tikhonov regularized estimates of the m_T^i . These estimates are based on synthetic data generated for the training models through (1). The goodness of this functional depends on the appropriate choice of training models and family of functionals. We have provided examples of families \mathcal{R}_θ parametrized by θ . The methodology can be applied to functionals whose dependence on θ can be nonlinear. The linear case is straightforward but the case of nonlinear regularization functionals is more involved and requires an iterative solver.

As an initial check on the choice of \mathcal{R} , one can determine the smallest discrepancy η so that $m_T^i \in \mathcal{M}(\eta, \tau)$. If this η is acceptable, then we assume that the training models are similar enough to m in the sense that m also belongs to $\mathcal{M}(\eta, \tau)$.

The last step in the process is a Tikhonov regularization with the learned regularization functional and the original data (1). The selection of numerical procedures to solve each of the optimization problems is very much problem dependent, the methodology has to be tailored to the particular forward operator and regularization functional.

References

- [1] D. Black, G. Sapiro, D. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Tran. on Image Processing*, 7:442–449, 1998.

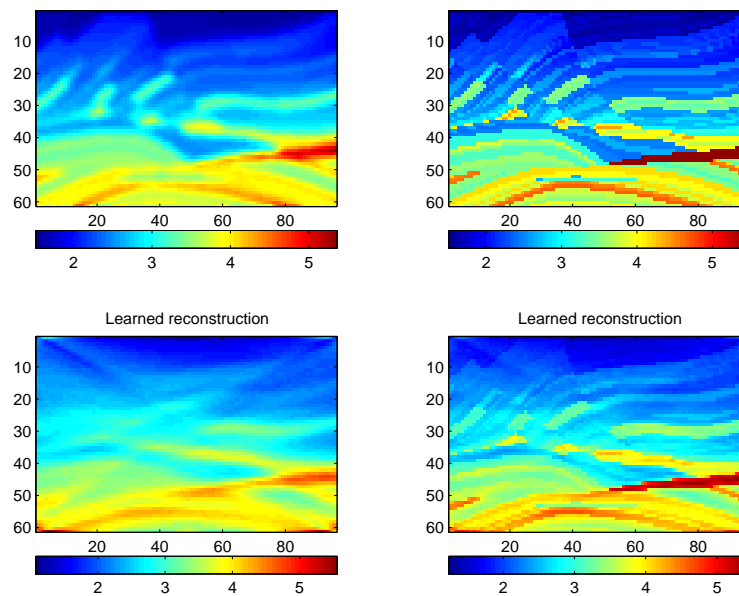


Figure 9: Reconstructions of the hard and smooth Marmousi models using the learned regularization operators shown in Figure 8. The true models (top) and reconstructed models (bottom).

- [2] T. Chan and J. Shen. Mathematical models for image inpainting. *SIAM J. Appl. Math.*, 62:1019–1043, 2001.
- [3] J. Claerbout. *Imaging the Earth's Interior*. Blackwell Scientific Publications, 1985.
- [4] J. Claerbout and F. Muir. Robust modeling with erratic data. *Geophysics*, 38:826–844, 1973.
- [5] J. Cullum. The effective choice of the smoothing norm in regularization. *Mathematics of Computation*, 33:149–170, 1979.
- [6] E. Haber and U. Ascher. Preconditioned all-at-one methods for large, sparse parameter estimation problems. *Inverse Problems*, 2001. To appear.
- [7] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse problems*, 16:1263–1280, 2000.
- [8] P. C. Hansen. *Rank Deficient and Ill-Posed Problems*. SIAM, Philadelphia, 1998.
- [9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, New-York, 2001.
- [10] A. K. Jain. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

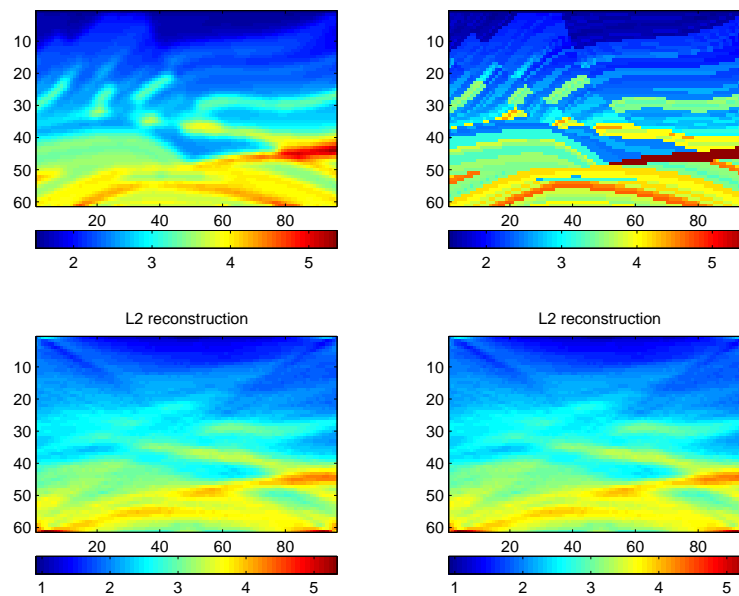


Figure 10: Reconstructions of the hard and smooth Marmousi models using the smooth Tichonov regularization. The true models (top) and reconstructed models (bottom).

- [11] E. Janes. *Probability Theory, the Logic of Science*. www.bayes.wustl.edu/etj/prob.html, 1999.
- [12] S. Li. *Markov Random Fields Modeling in Image Analysis*. Springer-Verlag, 2001.
- [13] Y. Li and D.W. Oldenburg. 3-D inversion of magnetic data. *Geophysics*, 61:394–408, 1996. 2.
- [14] D. Mackay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.
- [15] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [16] J. Nocedal and S. Wright. *Numerical Optimization*. New York: Springer, 1999.
- [17] R. L. Parker. *Geophysical Inverse Theory*. Princeton University Press, Princeton NJ, 1994.
- [18] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal*, 12:629–639, 1990.
- [19] A. Tarantola. *Inverse Problem Theory*. Elsevier, Amsterdam, 1987.
- [20] A.N. Tikhonov and V.Ya. Arsenin. *Methods for Solving Ill-posed Problems*. John Wiley and Sons, Inc., 1977.

- [21] C. R. Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2001.
- [22] J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G Teubner Stuttgart, 1998.
- [23] K. P. Whittall and D. W. Oldenburg. *Inversion of Magnetotelluric Data for a One Dimensional Conductivity*, volume 5. SEG monograph, 1992.
- [24] Z. Yuval. *Strategies for Implementing Neural Networks in Ocean and Atmosphere studies*. PhD thesis, University of British Columbia, Vancouver, BC, Canada, 2001.
- [25] S. C. Zhu and X. W. Liu. Learning in Gibbsian fields: how accurate and how fast can it be? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24:104–109, 2002. n8.
- [26] S. C. Zhu and D. Mumford. Learning generic priors for visual computation. *IEEE Tran. on Pattern Analysis and Machine Intelligence*, 13:1–20, 1999.
- [27] S.C. Zhu, Y.N. Wu, and D. Mumford. FRAME: filters, random fields, and maximum entropy - towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27:1–20, 1998. n2.