THESIS PROPOSAL

JORDAN R. HALL

Abstract. Obtaining the solution to stochastic inverse problems is an important challenge in computational science. By solving a stochastic inverse problem, one may find an updated characterization of parameter space that combines prior beliefs and observable data. Using a new approach detailed in [Butler], we may solve for a probability distribution in parameter space, Λ, such that its push-forward (*i.e.,* image) through the parameter-to-data map f matches the observed probability density on the data, $\mathcal{D}$. Under certain conditions [Butler, Tarantola], an approximate solution of the stochastic inverse problem can be obtained as the solution of a regularized, convex minimization problem.

In this proposal, we consider a noisy map f from a high-dimensional Λ to $\mathcal{D}$ of specified dimension, where the gradient of f exists, but may be inaccessible. If the gradient of f is accessible, the minimization problem can be solved using gradient-based descent methods. If the gradient of f is inaccessible, one may apply Derivative-Free Optimization (DFO) schemes as in [Chen and Wild]. In either case, the effectiveness of the optimization algorithm may be enhanced by performing dimension reduction in Λ as in [Russi, Constantine]. We consider different strategies for computing the active subspace of $f(\Lambda)$ in the context of both approaches, and apply them to the data-consistent inversion of two model problems. Both of these model problems are related to stochastic inverse problems arising in the simulation of plasmas in fusion reactors.

Contents

Literature Review and Framework 2
Research Questions 6

Literature Review and Framework

In the proceeding section we provide a literature review of inverse problem theory, derivative-free (DF) optimization, and dimension reduction. In the process, we will build a theoretical framework upon which to pose research questions, state initial results, and form a research plan.

We begin our discussion by defining a parameter space Λ of dimension N, a map or "model" f, and a data space $\mathcal{D}$. We assume that $\mathcal{D}$ has dimension M, where most often $M \leq N$. Points in $\mathcal{D}$ may be known values of $f(\lambda), \lambda \in \Lambda$; as such, we may write $d = f(\lambda)$ to denote the particular datum corresponding to the evaluation of a point $\lambda \in \Lambda$ under f. Points in $\mathcal{D}$ may also be *observed* data, denoted d_{obs}, where the corresponding $\lambda \in \Lambda$ may be unknown. We also assume that f is noisy. Hence, we may model draws of f with $f(\lambda) = f(\lambda) + \epsilon$, which is an additive noise structure, or $f(\lambda) = f(\lambda)(1 + \epsilon)$, which is a multiplicative noise structure.

Data-Consistent Inversion We follow [B,T,S] to formulate an inverse problem in our setting. Then, we briefly present the data-consistent solution; we refer the interested reader to [B] for details.

First, we assume we have prior knowledge of the parameter space, given by the so-called "prior distribution," $\pi_\Lambda^{\text{prior}}(\lambda)$. In practical applications, a prior distribution may be given by application experts, but may also be a state of knowledge on Λ obtained by other mathematical or statistical processes. Second, we assume that we have a so-called "observed" density, $\pi_{\mathcal{D}}(d)$, which represents our uncertain state of knowledge of observed data and is uncertain due to noise in f and potential measurement error. Informally, the inverse problem is the task of finding an *updated* probability distribution, $\pi_\Lambda^{\text{update}}$, in Λ space that combines the given prior information and data.

It has been shown [] that the so-called classical Bayesian or statistical Bayesian [T,S] solution to the inverse problem is not generally a "pull-back probability measure," in the sense that the image of the updated distribution $\pi_\Lambda^{\text{update}}$ under the map f, called the push-forward, is not equal to (and often not "close" to) the "observed" probability distribution, $\pi_{\mathcal{D}}$.

Data-consistent inversion (DCI) seeks an updated solution $\pi_\Lambda^{\text{update}}$ for which the push-forward exactly equals $\pi_{\mathcal{D}}$. To obtain such a solution, it is necessary to compute the push-forward of $\pi_\Lambda^{\text{prior}}$. We may form the push-forward of the prior by solving a *forward problem*; with certain assumptions on f, Λ, and $\mathcal{D}$ the forward problem may be straightforward, with other assumptions, this problem is non-trivial. We denote the solution to the forward problem with $\pi_{\mathcal{D}}^{f(\Lambda)}$. Then, as in [], the *data-consistent* solution to the inverse problem is

$$\pi_\Lambda^{\text{update}} = \pi_\Lambda^{\text{prior}}(\lambda) \frac{\pi_{\mathcal{D}}(f(\lambda))}{\pi_{\mathcal{D}}^{f(\Lambda)}(f(\lambda))}. \tag{1}$$

Optimization Methods for Solving Inverse Problems With our assumptions of an expensive f and a high-dimensional Λ, approximately solving the forward problem to obtain the push-forward density generally requires some type of kernel density estimation (KDE), which converges slowly, typically at Monte Carlo rates of $\frac{1}{\sqrt{N}}$. As in [Tarantola], depending on f, the solution to the inverse problem can be obtained exactly or approximately by solving an equivalent deterministic convex optimization problem. As in [Butler], we may

alter the aforementioned classical formulation of the deterministic optimization problem to ensure we obtain, depending on f, the *data-consistent* (exact or approximate) solution to the inverse problem.

We begin by examining the classical formulation of a deterministic optimization problem which corresponds to either exactly or approximately solving the stochastic inverse problem under consideration. In particular, for a linear map f, the exact solution is obtained; for nonlinear maps, we obtain an approximate solution. As in [Taratola], given observed data $d_{\text{obs}} \sim \pi_{\mathcal{D}}$ and a draw $\lambda_{\text{prior}} \sim \pi_{\Lambda}^{\text{prior}}$, we define a *misfit function*

$$S(\lambda) = \frac{1}{2}\left(||f(\lambda) - d_{\text{obs}}||_{\mathcal{D}}^2 + ||\lambda - \lambda_{\text{prior}}||_{\Lambda}^2\right). \tag{2}$$

The misfit function appears in the classical (i.e., statistical Bayesian) solution to the inverse problem, as $\pi_{\Lambda}^{\text{update}}(\lambda) \sim \exp(-S(\lambda))$. We observe, as in [T], that finding the MAP point is equivalent to finding the λ for which S is minimized.

As in the preceding section, we assume $\pi_{\Lambda}^{\text{prior}}$ is a known probability density with mean $\bar{\lambda}$ and covariance matrix C_{Λ}; likewise, $\pi_{\mathcal{D}}$ is a known probability density with mean $\bar{d}$ and covariance matrix $C_{\mathcal{D}}$. As in [B], we can use the assumed covariance structures on Λ and $\mathcal{D}$ to rewrite (2) as

$$S(\lambda) = \frac{1}{2}\left(\left|\left|C_{\mathcal{D}}^{-1/2}(f(\lambda) - d_{\text{obs}})\right|\right|_2^2 + \left|\left|C_{\Lambda}^{-1/2}(\lambda - \lambda_{\text{prior}})\right|\right|_2^2\right). \tag{3}$$

Classically ([T]), it is assumed that $\pi_{\Lambda}^{\text{prior}}$ and $\pi_{\mathcal{D}}$ are Gaussians, and the misfit function is specified by $\lambda_{\text{prior}} = \bar{\lambda}$ and $d_{\text{obs}} = \bar{d}$ in our notations. In (2) and (3), we choose to define a more general misfit function, so that we may consider the "misfit" given any draws $\lambda_{\text{prior}} \sim \pi_{\Lambda}^{\text{prior}}$ and $d_{\text{obs}} \sim \pi_{\mathcal{D}}$. The misfit function defined above can be understood term-by-term. Given $\lambda \in \Lambda$, the term $||f(\lambda) - d_{\text{obs}}||_{\mathcal{D}}^2$ corresponds to finding the mismatch between the observed d_{obs} and $f(\lambda)$ in data space; the term $||\lambda - \lambda_{\text{prior}}||_{\Lambda}^2$ corresponds to performing Tikhonov regularization, which improves conditioning in minimizing S over Λ. The minimum of S is typically referred to as the maximum a posteriori point or *MAP point*.

Authors in [] rewrite S so that a data-consistent MAP point may be obtained. An additional "deregularization" term is appended so that if a unique solution exists, the regularization will be "turned off." For now, we assume that f is linear (or can be linearized locally [T]), and write the matrix A to define that action of f on Λ. We also assume that π_{Λ} and $\pi_{\mathcal{D}}$ are Guassian. In this case, we write the *data-consistent misfit function,*

$$T(\lambda) = \frac{1}{2}\left(\left|\left|C_{\mathcal{D}}^{-1/2}(f(\lambda) - d_{\text{obs}})\right|\right|_2^2 + \left|\left|C_{\Lambda}^{-1/2}(\lambda - \lambda_{\text{prior}})\right|\right|_2^2 - \left|\left|C_{A}^{-1/2}(f(\lambda) - f(\lambda_{\text{prior}}))\right|\right|_2^2\right), \tag{4}$$

where $C_A = AC_{\Lambda}A^T$. We note that in the case that π_{Λ} and $\pi_{\mathcal{D}}$ are Gaussian and f is linear, the data-consistent solution to our inverse problem is given exactly by $\pi_{\Lambda}^{\text{update}}(\lambda) \sim \exp(-T(\lambda))$. Generally, the deregularization term will ensure a solution that updates the distribution on Λ only in the directions in which the data is informative. The linear case illustrates the action of the un-regularization term. In the case that a unique solution to the inverse problem exists for a linear f (i.e., $N = M$ and A is full rank), the un-regularization term will equal the Tikhonov regularization term, so that the MAP point solves the data

misfit exactly. When the problem is under-determined ($N > M$), then with a linear f, the data-consistent solution to inverse problem will lie in a hyperplane in Λ containing all the possible points that could have produced the M observed data. Regardless of whether $N = M$ or $N > M$, the MAP point obtained by minimizing the "classical" misfit, in general, will not be a point that could have produced the observed data. We highlight the difference in solutions in the example below.

Example 1. *Let $f(\lambda) = 2\lambda$, $\lambda_{prior} = 0.1$, $C_\Lambda = [0.5]$, $d_{obs} = 0.25$, and $C_\mathcal{D} = [0.25]$; note, $N = M = 1$. Then we find that $S(\lambda) = 2((2\lambda - 0.25)^2 + (\lambda - 0.1)^2)$, which is minimized by $\lambda_S^* = 3/25$. Assuming that λ_{prior} and d_{obs} are the means of Gaussians with standard deviations corresponding to the above covariance matrices, we have $\pi_\Lambda^{post} \sim \exp(-S(\lambda_S^*))$. Notice $f(\lambda_S^*) = 6/25 \neq d_{obs}$. We write $T(\lambda) = 2(2\lambda - 0.25)^2$, which is minimized by $\lambda_T^* = 1/8$. Notice $f(\lambda_T^*) = d_{obs}$. With Gaussian assumptions on the prior and data, we have $\pi_\Lambda^{post} \sim \exp(-T(\lambda_T^*))$.*

In the preceding example, we observe for linear f and $N = M$, the classical MAP point, given by $\lambda_S^* = 0.12$ and obtained by minimizing S, strikes a balance between the prior belief that $\lambda = 0.1$ and the fact that the value $\lambda = 0.125$ is the pre-image of 0.25 under f, the observed data point. The data-consistent MAP point, given by $\lambda_T^* = 0.125$, is exactly the pre-image of d_{obs}. Indeed, since the linear map A is full rank in Example 1, the third term in (3) fully deregularizes (cancels out) the second term, so that the minimum of T is actually just the minimum of the first term, the data mismatch. We see here that classical Bayesian inverse problem theory and DCI ask and answer different questions. For a similar, concrete example where the number of parameters N is greater than the number of data, M (so that the problem is under-determined), we refer the interested reader to [].

So far, we have presented a deregularization term in (3) which will hold only for linear (or linearizable) f and Gaussian assumptions on the parameter and data spaces. For general nonlinear functions f, the deregularization must be rewritten. We expand the term, leaving f as a nonlinear map, and defining $A^\top$ as the transpose of the linearization of f at some point $\bar{\lambda} \in \Lambda$. For example, $A^\top$ may be a solution to some form

$$(f(e), v)_\mathcal{D} = (e, A^*(e)v)_\Lambda,$$

which in general is not unique.

Varis: This was the best I could do, and I am not sure it is correct! Can you edit/perhaps add some content here??

Dimension Reduction In our setting, it may be advantageous to perform dimension reduction on f. In particular, we shall consider Active Subspace methods described by Paul Constantine in [] and an equivalent method by T.M. Russi in []. These techniques seek to explain outputs $f(\Lambda)$ in a subspace $A \subset \Lambda$ for which the $\dim(A) < N$. Here we discuss the theoretical formulation of A, but also how it may be found in practice.

We begin by noting $\nabla f(\lambda) \in \Lambda$ is a column vector of the same dimension as inputs λ with rows containing the N partial derivatives of f, which for this discussion we assume exist, and are square integrable in Λ equipped with some probability density that is positive everywhere in Λ and 0 otherwise; for instance, we could consider $\pi_\Lambda^{\text{prior}}(\lambda)$, the density describing our prior state of knowledge, which we choose to abbreviate as π_Λ for this discussion. In classical active subspace analysis, one transforms inputs λ to the origin with some fixed variance, typically so that $\lambda \in [-1, 1]$. Then, as in [C,E,W, and C], we may write

$$W = \int_{\Lambda} \nabla f(\lambda) \nabla f(\lambda)^{\top} \pi_{\Lambda}(\lambda) d\lambda, \tag{5}$$

which is an $N \times N$ symmetric positive semi-definite matrix which defines a certain covariance structure of ∇f over Λ. This interpretation of (9) leads one to the idea of computing the Singular Value Decomposition of W,

$$W = U \Sigma V^{*}, \tag{6}$$

where U is $N \times N$ unitary, Σ is $N \times N$ diagonal with the singular values of W along its diagonal, and V^{*} is $N \times N$ unitary. With the singular values of W in hand, we search for a drop-off in the spectrum of W. In detail, we plot the singular values, $\{\sigma_i\}_{i=1}^{n}$ and seek a drop-off in magnitude between some pair of singular values, σ_j and σ_{j+1}. The active subspace is the span of $u_1, \ldots, u_j$, which are the first j columns of W, the so-called "left singular vectors" of W.

The fact that $u_1, \ldots, u_j$ correspond to the nontrivial singular values is exactly why they account for the most amount of variance in function values. In fact, one can view AS analysis as an artful choice of principal components after a *full* Principal Component Analysis (PCA) is performed in the gradient space W; for more details on this viewpoint, we refer the interested reader to Section 6.4 in T.M. Russi.

For a point $\lambda \in \Lambda$, we define

$$\mathcal{P}_A(\lambda) = \sum_{i=1}^{j} \left(u_i^T \lambda\right) u_i \in A, \tag{7}$$

which is a projection of the point λ in the "active directions" of f. We call this projection an "active variable," which is a point in the active subspace A. We have arrived at the property that

$$f\left(\mathcal{P}_A(\lambda)\right) \approx f(\lambda). \tag{8}$$

In practice, finding an active subspace of f will require forming an approximation to W in a Monte Carlo fashion; see []. We choose to present a Monte Carlo approach that is simple to implement. In short, for a random draw $\lambda_i \in \Lambda$, we will find its evaluation under the approximate or analytic gradient (depending on whether we have ∇f analytically), and store the row vector $f(\lambda_i)^{\top}$ in a matrix with an SVD corresponding to (6), up to scaling.

In the following, we assume that we lack an analytic form of ∇f; if the analytic gradient is available, the proceeding Monte Carlo method remains valid, and all notations corresponding to approximating ∇f may be dropped.

One initializes the method by performing M random draws of $\lambda_i \in \Lambda$. We then compute $f(\lambda_i)$ for all $i = 1, \ldots, M$ samples, which we note will require, at the very least, M evaluations of f; in a realistic setting, this would require M solves of a model (e.g., M solves of a PDE constrained system). In our notations, we have $\mathcal{D} = \{f(\lambda_i)\}_{i=1}^{M}$. Next, we need $\nabla_{\Lambda} f$ evaluated at λ_i for all $i = 1, \ldots, M$, which we assume that we do not have in closed analytic form. Hence, we generally need some gradient approximation method, and typically a locally linear approximation to the gradient is a fair balance between reasonably estimating the gradient and not pushing computational expenses to an unreasonable regime. With this

approximation formed, we denote each estimation to $\nabla f(\lambda_i)$ with $\widehat{\nabla f}(\lambda_i)$ and we define the $N \times M$ matrix

$$\tilde{W} := \left[\widehat{\nabla f}(\lambda_1)^\top \cdots \widehat{\nabla f}(\lambda_M)^\top \right]. \tag{9}$$

Forming the SVD of $\tilde{W}$, $\tilde{W} = \tilde{U}\tilde{\Sigma}\tilde{V}^*$, we search for a drop off in the magnitude of the singular values $\{\tilde{\sigma}_i\}_{i=1}^M$. Assuming such a drop off occurs for an index $j : 1 < j < M$, we have the j corresponding left singular vectors,$\tilde{u}, \ldots, \tilde{u}_j$. We let $A(f; \mathcal{D}) := \text{span}\{\tilde{u}, \ldots, \tilde{u}_j\}$ to denote the active subspace of f with respect to the data $\mathcal{D}$. We choose to use a notation with $\mathcal{D}$ included to emphasize the dependence of the active subspace on the random draws made in Λ, which led to our particular $\mathcal{D}$.

In practice, we can check the extent to which the active subspace accounts for functions values $f(\lambda)$ by checking for resolution in a so-called "sufficient summary plot" [Constantine], where we plot active variables against function values. In these plots, we hope to see a pattern between the active variables versus the actual function values. For example, if f is quadratic in its active variables, then we expect to see quadratically-resolved sufficient summary plots. We will revisit sufficient summary plots in the proceeding sections. For now, we continue our discussion with the understanding that sufficient summary plots allow us to check the "fit" of the active subspace.

Varis: I'll write an intro here connecting DF algorithms to noisy simulations due to turbulence/chaos to

Derivative-Free Optimization

We consider derivative-free optimization (DFO) algorithms suited for additive and multiplicative noise as in [C/W]. This technique requires nothing more than evaluations of the noisy model and random draws from a normal distribution. Briefly, this method finds a new iterate by randomly perturbing the previous iterate in Λ; iterates are not allowed to stray much, though, due to relatively small smoothing factors and step sizes. The smoothing factor and step size in the DF algorithms are of great importance to their convergence and termination. As in [Chen and Wild], both the smoothing factor and step size will depend on a scale factor of the L_1 Lipschitz constant of f. As such, it will be of interest to obtain estimates of L_1, which is not straightforward in a gradient-free setting. We refer to [Others] for Lipschitz constant learning in this setting.

Varis: I'll write this, covering Grad-Shafonov Equations, and why these MHD equilibria are important to kinetic simulations.

Application: Magnetic Equilibria in Tokamaks

Research Questions