Probabilistic inverse Theory. Assignment 5 – Inverting a Horizontal Slice of 3D Seismic Reflection Data

J. Alejandro Duran, ETH December 10th, 2017

1 Objective

We want to solve the inversion problem given the data of an horizontal slice of 3D seismic reflection data. We also have information about the source (pointspread) and a priori knowledge of the the Earth model given in the training image "Strebelle.tif". The objective is to use a metropolis algorithm and obtain a proposed model which explain the observed data within the error bars.

2 Considerations

The training image is a black and white image as it is shown in figure 1. It simulates a pre-historic river system, consisting of sandstone channels (black pixels) surrounded by shale (white pixels).

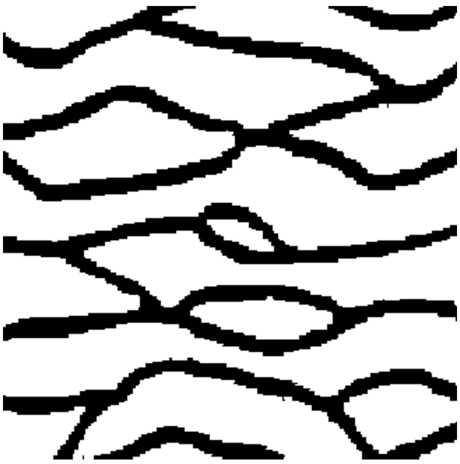

Figure 1: Training image

Following the previous homework, we know that we can obtain statistics from the training image by using a 2x2 moving window such that it scans all possible configurations of black and white patterns given in the image. From this, we obtain the distribution of the training image (we obtain a histogram and then the marginal probabilities) that we will continuously use as a reference. It is presented in the figure 2.

It is important to notice that there are two configurations (configuration 7 and 10) that are not present at all in the training image. This is difficult to see from figure 2 due to the scale, the high probability of configuration number 16 (all white) and 1 (all black) do not allow to see the tiny details. However, if we have a look to the numbers we realize that configuration 7 and 10 have probability equal to zero. This will be important when considering the computation of the prior of a given model via equation 1.

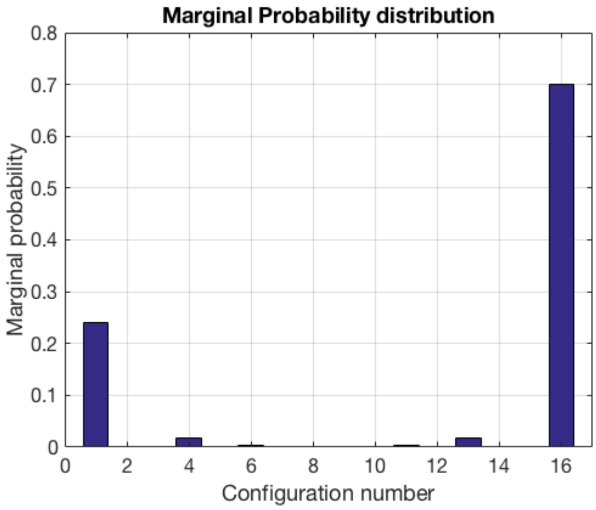

Figure 2: Marginal probability of each pattern obtained by scanning 2x2 squares in the training image.

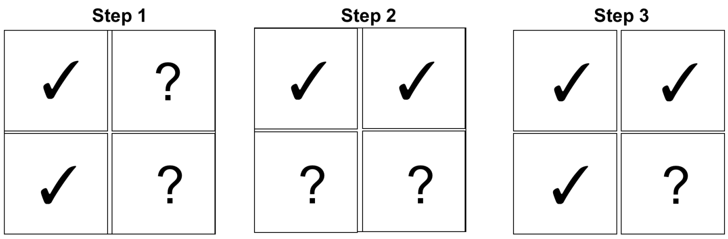

Figure 3: Representation of the three steps performed when creating the initial model

The procedure now consist in creating an initial model which follows the statistics of the training image. This is in fact the solution of the previous homework. Following three steps (please observe figure 3 or have a look to the previous homework) I computed conditional probabilities which are based on the distribution of the training image and build the new image by picking random configurations from this distribution.

Once the new model is built, I can compute its prior and its likelihood (actually I worked first with the misfit to avoid round to zero as I will show in the next section) an evaluate the Metopolis condition.

2.1 Computation of the prior

First we consider that an arbitrary image (model) is made as a realization from a Markov random field with a given neighborhood structure. The structure is given by the steps 1 to 3 in the figure 3 which describe how the model was built.

The prior of a model $\mathbf{m}$ is computed by the product of the conditional probabilities as:

$$P(\mathbf{m}) = \prod_{i=1}^{N} p(\mathbf{r}_i|\mathbf{s}_i) \tag{1}$$

3 The Metropolis algorithm

Here I make a summary of the procedure performed to get the solution of the inverse problem via Metropolis algorithm.

- Use the statistics of the training image to create a starting model $\mathbf{m}_0$ of the same size of the observable data (41x41). While constructing it, compute the prior of that specific model based on equation 1.
- Pick randomly a pixel in $\mathbf{m}_0$ and then change the color of that pixel. This creates a new model: $\mathbf{m}_{new}$.
- Compute the prior of $\mathbf{m}_{new}$ and test if it is non zero (it could happen that when modifying a pixel we end up with a 2x2 pattern which did not appear in the training image and therefore has probability zero. Then, from equation 1, one probability could make the whole prior goes to zero).
- Obtain the synthetic data of the new model by doing `syndata =conv2(`$\mathbf{m}_{new}$`,pointspread,'same')` Later I converted both data and synthetic data (which are matrices 41x41) into vectors.
- Obtain the likelihood, not by computing it directly but first computing the misfit

  $$mis = -\frac{1}{2}(\mathbf{d} - \mathbf{d}_{pred})^T \mathbf{C}_D^{-1}(\mathbf{d} - \mathbf{d}_{pred}) \tag{2}$$

  and then use the exponential to get it. The noise on the data is Gaussian and its standard deviation satisfies std=0.07max[data]
- Evaluate the metropolis condition. The new model is accepted if it satisfies:

  $$\alpha \leqslant min(1, \frac{\sigma(\mathbf{m}_{new})}{\sigma(\mathbf{m}_{cur})}) \tag{3}$$

  where α is a random number such that $\alpha \in [0, 1)$, $\mathbf{m}_{curr}$ it the current model and $\sigma(\mathbf{m})$ is the posterior.

  However, there are some inconveniences in computing the acceptance probability in this way, mainly because the likelihood is round to zero by the computer when its value is too small. It is better to work with the logarithm of the likelihood (the misfit) and perform the exponentiation at the end.
- If the new model is accepted, we set $\mathbf{m}_{curr} = \mathbf{m}_{new}$ and repeat the process. Otherwise we do not save anything and make a new model (by changing 1 pixel in the initial model).

4 The results

I perform tests by running the algorithm at several iterations. The higher the number the better the results, but from a certain threshold the improvement is too small that it does not worth to run it for so many iterations.

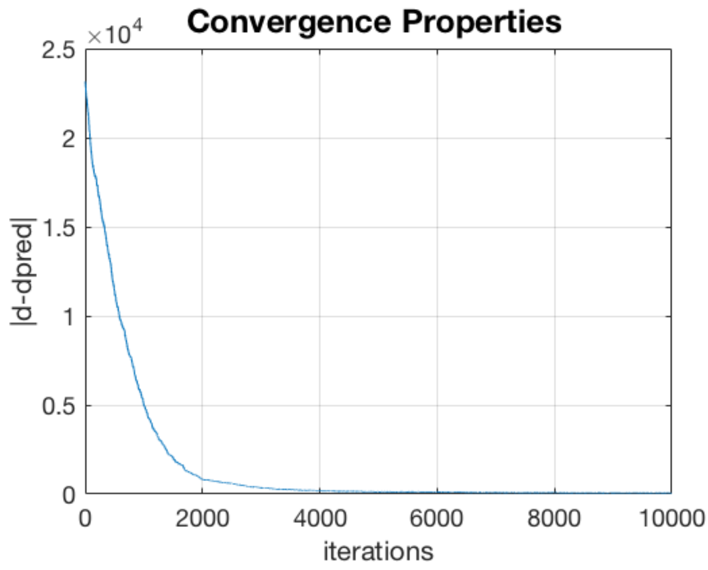

Figure 4: Analysis of the convergence

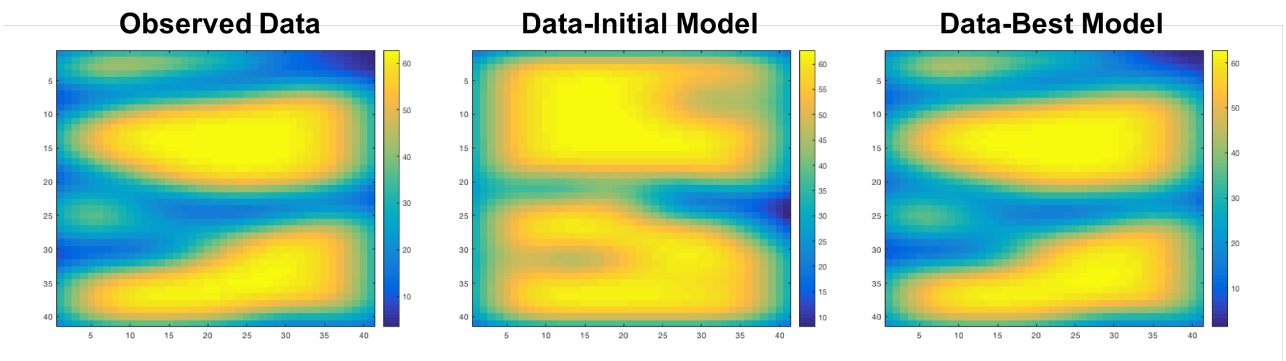

Figure 5: LHS: Observed data. Center: Synthetic data obtained from the initial model. RHS: Synthetic data obtained from the model which maximizes the posterior.

Figure 4 shows the convergence properties. The burn in time is presented around 2000 iterations. After that, the misfit is decreasing slowly.

Figure 5 presents the results of the inversion when the algorithm was run over fifty thousands iterations. In the left hand side is the observed data, in the center is the data obtained when convolving the initial model with the pointspread source, it is the first synthetic data we obtained. Finally, in the right hand side there is the synthetic data obtained from the best model. By the best model I mean the model which maximizes the posterior. As we can observe, by the blinked eye there is basically no difference between the observed data and the synthetic one. My model is a solution of the inverse problem and is able to describe the data correctly.

The figure 6 shows the models which generated the data in Figure 5. In the left hand side the true model used to create the observed data is displayed. In the center is the best model obtained and in the right hand side I showed the average over the accepted models (models which full-fill the metropolis acceptance condition) whose iteration number is higher than the burn in (in this way I include only models whose misfit is low, or at least acceptable, i.e. their synthetic data is not too different from the observed data).

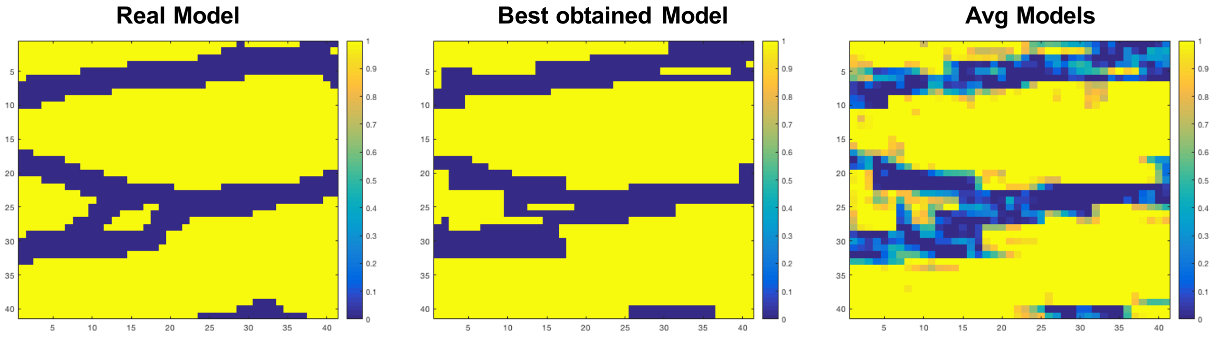

Figure 6: LHS: True model used to generate the observed data. Center: model which maximizes the posterior. RHS: Average over the accepted models after the bur-in period

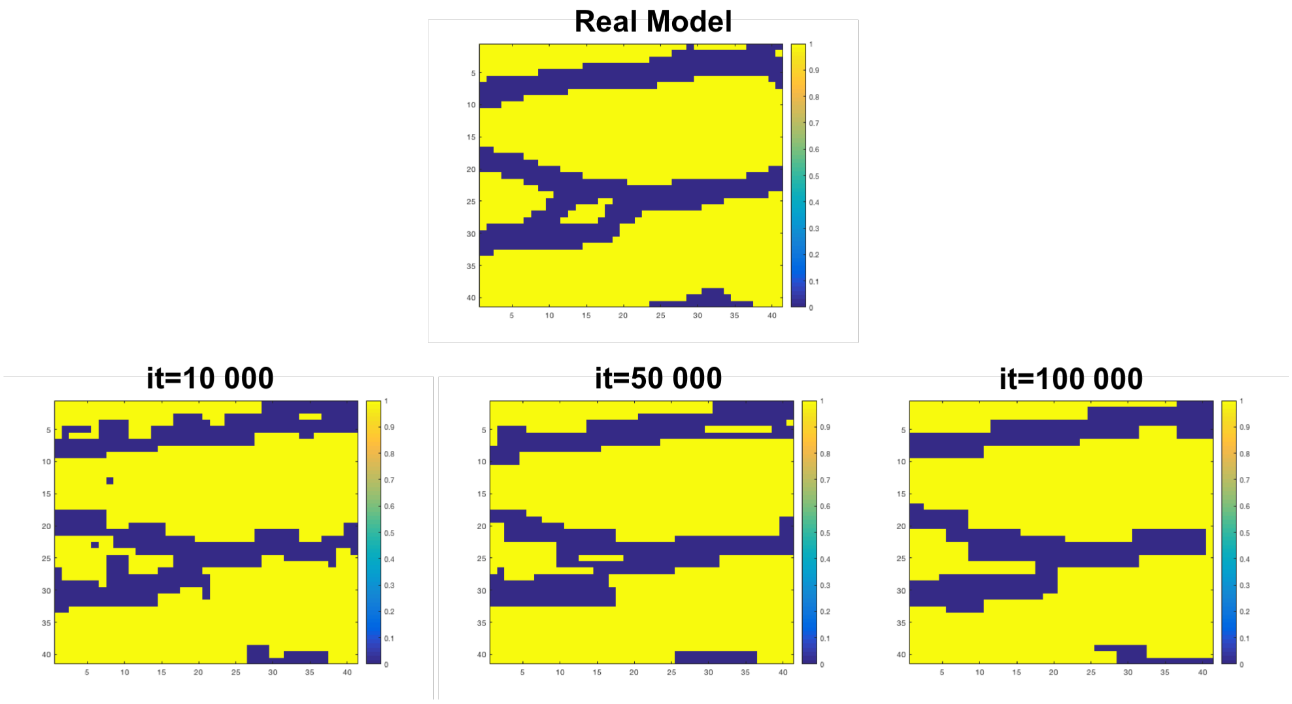

Figure 7: Improvement of the best models after various iterations.

As we can observe the obtained models have features very similar to the actual model. If I use either the best model or the average, both produce a synthetic data which is basically the same as the observed one.

Finally, figure 7 shows how the best obtained models are changing when I increase the iteration number. Due that the burn in is observed at around 2000 iterations, the difference between these three cases (iteration 10000, 50000 and 100000) is small. In fact, all three models will produce a synthetic data which is comparable with the observed one.