Robust full waveform inversion: In which domain should we measure the misfit?

F.J Herrmann, T. van Leeuwen, A.Y. Aravkin - UBC
H. Calandra - Total
Overview

- MAP estimation
- Outliers
- Students T
- Results
- Conclusions
MAP estimation

measurement model:
\[ d_i = F_i(m) + n_i \]

posterior likelihood:
\[ \pi_{\text{post}}(m) \sim \prod_{i=1}^{K} \pi_{\text{noise}}(F_i(m) - d_i) \pi_{\text{prior}}(m) \]
Maximization of the likelihood

\[ \max_m \pi_{\text{post}}(m) \]

is equivalent to

\[ \min_m - \log(\pi_{\text{post}}(m)) \]
MAP estimation

For Gaussian noise we have

$$\pi_{\text{noise}}(r) \sim \exp\left(-\|r\|^2_2\right)$$

which leads to the usual least-squares formulation

$$\min_m \sum_i \|F_i(m) - d_i\|_2^2$$
The use of alternative penalties can be interpreted as using a different noise model

$$\min_{\mathbf{m}} \sum_i \rho \left( F_i(\mathbf{m}) - \mathbf{d}_i \right)$$
MAP estimation

densities & penalties

Gaussian, Laplace and Students $T$
MAP estimation

data with 50% “bad traces”

true model

histogram of true noise
MAP estimation

least-squares penalty

recovered model

histogram of residual
MAP estimation

Huber penalty

recovered model

histogram of residual
MAP estimation

Students T penalty

recovered model

histogram of residual
MAP estimation

- Noise does not come from Students T distribution
- Use of Students T penalty may still be beneficial
- Noise has to be spiky
Outliers

What *is* an outlier?

- t,x ✔
- f,x ✔
- f,k ✗
Outliers

What *is* an outlier?

t, x    ✓

f, x    ❌

f, k    ✓
Outliers

Measure the misfit in a domain that \textit{sparsifies} the noise

\[
\min_{\mathbf{m}} \sum_{i} \rho \left( B \left( F_i(\mathbf{m}) - \mathbf{d}_i \right) \right)
\]

e.g., Fourier, Radon, Curvelets,...
Students T

The penalty is given by

\[ \rho(r) = \sum_j \log(1 + |r_j|^2 / \sigma^2) \]

where \( \sigma \) is a scale parameter. The corresponding adjoint source is given by

\[ (\nabla \rho)_j = \frac{2r_j}{|r_j|^2 + \sigma^2} \]
Students T

Scale parameter is used to separate outliers from good data
Students T

- *scale* parameter controls which residuals are ignored
- similar to a *weighted* least-squares approach
- how should we *choose* $\sigma$?
- what about *source* estimation?
Source estimation

Use variable projection approach on

$$\min_{m,w} \sum_i \rho (B (w_i F_i(m) - d_i))$$

solve source-weights as

$$\min_{w_i} \rho (B (w_i F_i(m) - d_i))$$
Auto-tuning

*Extended Students T penalty:*

\[ \rho_{\sigma}(r) = -N \log \left( \frac{\Gamma \left( \frac{\sigma^2+1}{2} \right)}{\Gamma \left( \frac{\sigma^2}{2} \right) \sqrt{\pi \sigma^2}} \right) + \frac{\sigma^2 + 1}{2} \sum_{j=1}^{N} \log \left( 1 + \frac{r_j^2}{\sigma^2} \right) \]

find optimal \( \sigma \) for a given residual by solving

\[ \min_{\sigma} \rho_{\sigma}(r) \]
Workflow

1. Forward modeling \( \mathbf{d}_{i}^{\text{pred}} = F_{i}(\mathbf{m}_{k}) \)
2. Estimate source weight (scalar optimization)
3. Compute residual \( \mathbf{r}_{i} = w_{i}\mathbf{d}_{i}^{\text{pred}} - \mathbf{d}_{i} \)

4. Estimate scale (scalar optimization)
5. Compute adjoint source \( \tilde{\mathbf{r}}_{i} = B^{*}w_{i}^{*}\nabla \rho(B\mathbf{r}_{i}) \)

7. Compute gradient \( \mathbf{g} = \sum_{i} \nabla F_{i}(\mathbf{m}_{k})^{*}\tilde{\mathbf{r}}_{i} \)

9. update \( \mathbf{m}_{k+1} = \mathbf{m}_{k} - \lambda \mathbf{g} \)
Results 1

- Marmousi model with *periodic* noise.
- inversion of *single* frequency (4 Hz) with 20 iterations
- Misfit measured in \((f,x)\) or \((f,k)\).
Results 1

no noise:

periodic noise:

\( f-k \)

\( f-x \)
Results 2

Acoustic inversion

- $v_p$ [m/s]
- $v_s$ [m/s]
- $v_0$ [m/s]
- Density [kg/m$^3$]
Results 2

Variable density data, no noise

least-squares

\( v_p \) [km/s]

Students T (f,x)

\( v_p \) [km/s]
Results 2

Data with bad traces

least-squares

$\nu_p \text{ [km/s]}$

Students T ($f, x$)

$\nu_p \text{ [km/s]}$
Results 2

Elastic data

least-squares

$v_p$ [km/s]

Students T (f,k)

$v_p$ [km/s]
Conclusions

• *Robust* inversion works best when noise is *localized*
• Measure misfit in domain in which noise is *sparse*
• *Source* and *scale* estimation can be done *automatically*
• ...