Parameter Estimation

Daniel Alexander

Estimation

- Fit the model to the data.
- Find values for the parameters so that predicted measurements from the model match measured data best.
- Ie, minimize the difference between predicted and measured data with respect to the model parameters.

Overview

- Fitting error
- Optimization
- Imaging application: Parametric mapping
- Example: diffusion imaging

Least squares fit

- Minimizes sum of square differences (SSD):

$$f(\mathbf{x}) = \sum_{k=1}^{K} (A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k))^2$$

- K is the number of measurements and $A(\mathbf{y}_k)$ is the k-th measurement, which corresponds to device settings $\mathbf{y}_k$.

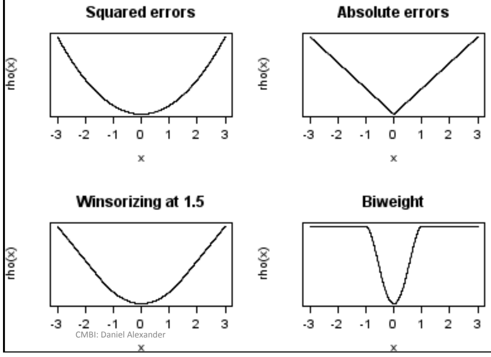
Maximum likelihood estimation

- Maximizes the likelihood of the data, $\mathbf{A} = (A(\mathbf{y}_1), \ldots, A(\mathbf{y}_K))^T$, under the noise model
$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}}\, p(\mathbf{A} \mid \mathbf{x})$$
- For additive noise
$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \prod_{k=1}^{K} p(A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k);\mathbf{z})$$
- For additive identically distributed Gaussian noise
$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{k=1}^{K} -(A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k))^2$$

Objective functions

- Sum of square differences $\sum_{k=1}^{K} (A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k))^2$
- L1 norm $\sum_{k=1}^{K} \left| A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k) \right|$
- Robust statistics and M-estimators
$$\sum_{k=1}^{K} \rho(A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k))$$

M-estimators



CMBI: Daniel Alexander 7

Noise model

- Sum of squares implies identically distributed zero-mean Gaussian noise
- Weighted sum of squares also simple to solve.
- Other noise models produce other M-estimators:
  - Laplacian distributed noise for L1 norm
  - t-Distributions for robust estimators

CMBI: Daniel Alexander 8

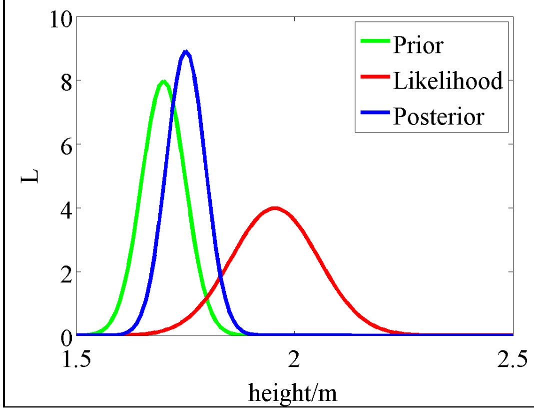
Bayesian estimation

- The maximum likelihood estimate maximizes $p(\mathbf{A} \mid \mathbf{x})$.
- Bayesian approach maximizes

$$\underbrace{\boxed{p(\mathbf{x} \mid \mathbf{A})}}_{\text{Posterior}} = \frac{\overbrace{\boxed{p(\mathbf{A} \mid \mathbf{x})}}^{\text{Likelihood}}\overbrace{\boxed{p(\mathbf{x})}}^{\text{Prior}}}{p(\mathbf{A})} = \frac{p(\mathbf{A} \mid \mathbf{x})p(\mathbf{x})}{\int p(\mathbf{A} \mid \mathbf{x})p(\mathbf{x})d\mathbf{x}}$$

- The maximum a-posteriori (MAP) estimate is
  $\tilde{\mathbf{x}} = \operatorname{argmax} p(\mathbf{x} \mid \mathbf{A}) = \operatorname{argmax}(\log p(\mathbf{A} \mid \mathbf{x}) + \log p(\mathbf{x}))$

9



Optimization

- Numerical techniques for locating the minima (or maxima) of a function.
  $\tilde{\mathbf{x}} = \operatorname{argmax} f(\mathbf{x})$
- f is the objective function.
- $\mathbf{x}$ is the vector of parameters.

CMBI: Daniel Alexander 11

Linear least squares

- A linear model has form $S(\mathbf{x};\mathbf{y}) = \sum_{i=1}^{N} g_i(\mathbf{y})x_i$
- Relate data and model parameters with a matrix equation, $G\mathbf{x} = \mathbf{A}$.
- Solve for $\mathbf{x}$: $\tilde{\mathbf{x}} = G^{+}\mathbf{A}$, where $G^{+} = (G^{T}G)^{-1}G^{T}$
- This is the least squares solution.

CMBI: Daniel Alexander 12

Weighted linear least squares

- Errors on each measurement are independent but have different variance

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{k=1}^{K} \frac{(A(\mathbf{y}_k) - S(\mathbf{x};\mathbf{y}_k))^2}{\sigma_k^2}$$

- The maximum likelihood estimate is

$$\tilde{\mathbf{x}} = (G^T W G)^{-1} G^T W \mathbf{A}$$

$$\text{where } W_{kk} = \frac{1}{\sigma_k^2}$$

Non-linear optimization

- Seeks a sequence $\mathbf{x}_0, \mathbf{x}_1, \ldots$ that converges to $\tilde{\mathbf{x}}$ where $\nabla f(\tilde{\mathbf{x}}) = 0$
- Gradient descent: $\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma \nabla f(\mathbf{x}_i)$
- Newton's method: $\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma H^{-1}(\mathbf{x}_i) \nabla f(\mathbf{x}_i)$

  where H is the Hessian of f.

An Introduction to the Conjugate Gradient Method Without the Agonizing Pain
Jonathan Richard Shewchuk
http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf

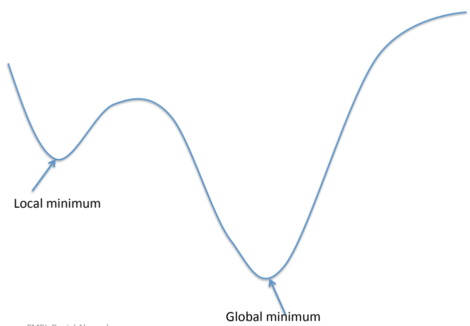
Non-linear optimization

- In `fminunc` in matlab:
  - Quasi-Newton methods compute H numerically from successive iterations
  - Gauss-Newton approximates H from first derivatives specifically for least squares.
  - Levenberg-Marquardt interpolates between gradient descent and Gauss-Newton for reliable convergence.
- Other deterministic techniques
  - Simplex method (`fminsearch` in matlab)
  - Powell's method
  - These do not require derivatives

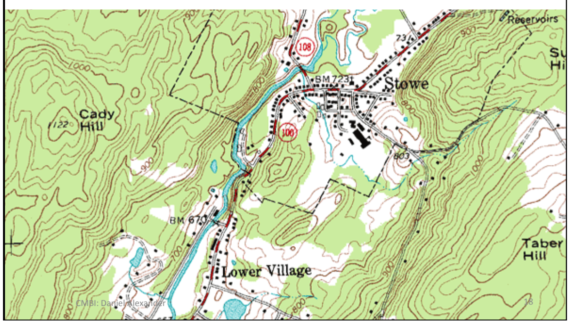
Constrained optimization

- Transformation method
- Soft or penalized constraints
- Legrange multipliers and active set methods (`fmincon` in matlab)

Global and local minima



Global and local minima



Stochastic techniques

- Repeated gradient descent
- Simulated annealing
- Population methods:
  - Genetic search
  - SOMA (self-organizing migratory algorithm)
  - Differential evolution