Effective Programming Practices for Economists

Numerical Optimization

Derivative-Free Trust Region Algorithms

Janoś Gabler, Hans-Martin von Gaudecker, and Tim Mensinger

Basic Idea (optimagic docs)

- Similar to derivative based trust region algorithm.

- Instead of Taylor approximation, use a surrogate model based on interpolation or regression.

  - Interpolation: Function is evaluated at exactly as many points as you need to fit the model.

  - Regression: Function is evaluated at more points than you strictly need. Better for noisy functions.

  - In general: Evaluation points are spread further out than for numerical derivatives.

- How the evaluation points are determined is complicated. It is also crucial for the efficiency of the algorithm.

Initial Evaluation

Actual $=$ expected $\Rightarrow$ accept, increase trust region radius

Iteration 1

Actual $=$ expected $\Rightarrow$ accept, increase trust region radius

Iteration 2

Actual $\approx$ expected, but step small $\Rightarrow$ accept, decrease trust region radius

Iteration 3

Actual $\approx$ expected, but step small $\Rightarrow$ accept, decrease trust region radius

Iteration 4

Actual $\approx$ expected, but step small $\Rightarrow$ accept, decrease trust region radius

Iteration 5

Converge because trust region radius shrinks to zero.

Some Remarks

- Within the trust region, the fit is generally better than the gradient based trust region algorithm

- By construction at the boundaries of the trust region for interpolation

- Choose between the two based on computation speed

  - If you have fast closed form derivatives, use the derivative based algorithm

  - If you only have numerical derivatives, use this instead

- It is intuitively very clear how this can work for noisy functions if enough evaluations are used for each surrogate model

A real algorithm: COBYLA