

Algorithms

Algorithms

1. Linear

- 1.1. Simplex Method
- 1.2. Interior Points

2. Quadratic

- 2.1. Simplified Simplex Method
- 2.2. Interior Point
- 2.3. Ellipsoid Algorithm

3. NLP without Constraints

- 3.1. Newton
 - 3.1.1. Original
 - 3.1.2. Levenberg-Marquardt algorithm
 - 3.1.3. Quasi-Newton (BFGS/DFP)
 - 3.1.4. Attention
- 3.2. Direction Determination+ Line minimization
 - 3.2.1. Steepest Descent Search (Gradient)
 - 3.2.2. Conjugate-Gradient Methods
 - 3.2.3. Perpendicular Search Methods
 - 3.2.4. Powell's perpendicular
 - 3.2.5. Parabolic Interpolation
 - 3.2.6. Golden Section
 - 3.2.7. Fibonacci method
- 3.3. Nelder-Meld
- 3.4. Newton Least Square

4. NLP with Constraints

- 4.1. Linear Equality (Elimination)
- 4.2. Nonlinear Equality (Lagrange)
- 4.3. Linear Inequality(Gradient Projection/SQP)
 - 4.3.1. Gradient Projection Method
- 4.4. Nonlinear Inequality (SQP/Barrier/Penalty)
- 4.5. SQP

5. Convex

- 5.1. Cutting Plane Method
- 5.2. Ellipsoid Algorithm
- 5.3. Interior Points

6. Global Optimal

- 6.1. Stopping Criterion
- 6.2. Genetic Algorithm

1. Linear

1.1. Simplex Method

1. Do not need multi-start
2. no stopping criterion is required:

Since the simplex algorithm finds the optimal solution in a finite number of steps, no stopping criterion is required.

However, in practice a maximum number of iterations is usually specified.

1.2. Interior Points

When Variable is too much

2. Quadratic

2.1. Simplified Simplex Method

only can be used when objective function is convex and quadratic

1. do not need multi-start
2. no stopping criterion is required:

Since the modified simplex algorithm finds the optimal solution in a finite number of steps, no stopping criterion is required.

However, in practice a maximum number of iterations is usually specified.

2.2. Interior Point

2.3. Ellipsoid Algorithm

3. NLP without Constraints

3.1. Newton

The gradient and Hessian Are needed

3.1.1. Original

The gradient and the Hessian of the objective function should easily be computed analytically.

1. need multi-start
2. stopping criterion

3.1.2. Levenberg-Marquardt algorithm

More general Newton that is able to jump out singularity

1. need multi-start
2. stopping criterion:

$$\|\nabla f(x_k)\|_2 \leq \epsilon$$

where x_k is the current iteration point and f is the objective function.

3.1.3. Quasi-Newton (BFGS/DFP)

Improve the speed of Newton Method if we need computed gradient/ Hessian numerically

1. need multi-start
2. stopping criterion (waiting test)

$$\|\nabla f(x_k)\|_2 \leq \epsilon$$

where x_k is the current iteration point and f is the objective function.

3.1.4. Attention

The Levenberg-Marquardt algorithm is to be preferred above the BFGS quasi-Newton method since the

Levenberg-Marquardt algorithm uses the exact Hessian (i.e., 2nd-order information) whenever possible, whereas

the BFGS method uses an approximation of the Hessian based on function values and gradients (i.e., 1st-order information)

If a choice has to be made between SQP and a penalty function approach with Levenberg-Marquardt - assuming that the Hessian can be computed easily, then SQP is in general preferred as both algorithms use second-order information but SQP can deal more accurately with the constraint.

3.2. Direction Determination+ Line minimization

3.2.1. Steepest Descent Search (Gradient)

The gradient of the objective function should be computed easily

1. need Multi-start
2. Stopping criterion:

$$\|\nabla f(x_k)\|_2 \leq \epsilon$$

where x_k is the current iteration point and f is the objective function.

- 3.

Attention: Because Steepest Descent Search only use Gradient but not Hessian, so when the gradient and Hessian are both easily to be computed, we prefer methods that take Hessian into consideration(like DFS)

3.2.2. Conjugate-Gradient Methods

Direction is similar with Newton

1. Need Multi-Start
2. Stopping Criterion (need test)

$$\|\nabla f(x_k)\|_2 \leq \epsilon$$

where x_k is the current iteration point and f is the objective function.

3.2.3. Perpendicular Search Methods

Do not need gradient and Hessian

1. Need Multi-Start
- 2.

3.2.4. Powell's perpendicular

Do not need gradient and Hessian

1. Need Multi-Start

3.2.5. Parabolic Interpolation

3.2.6. Golden Section

3.2.7. Fibonacci method

Fibonacci sequence $\{\mu_k\} = 0, 1, 1, 2, 3, 5, 8, 13, 21, \dots$

$$\mu_k = \mu_{k-1} + \mu_{k-2} \quad \mu_0 = 0, \quad \mu_1 = 1$$

Select n such that

$$\frac{1}{\mu_n}(b_0 - a_0) \leq \varepsilon$$

Next use

$$\lambda_l = \frac{\mu_{n-l}}{\mu_{n-l+1}}$$

3.3. Nelder-Meld

Do not need gradient and hessian

Method is less efficient than previous methods if more than 3–4 variables

1. need multi-start
2. stopping criterion:

$$\begin{aligned} |f(x_{k+1}) - f(x_k)| &\leq \epsilon_f \\ \|x_{k+1} - x_k\|_2 &\leq \epsilon_x \end{aligned}$$

where f is the objective function of the unconstrained problem (so if has penalty it also includes the penalty term)

3.4. Newton Least Square

For the Gauss-Newton or Levenberg-Marquardt nonlinear least squares method to work well, it is required that the error is small (see slide 7/23 of the unconstrained optimization slides). For a system identification problem this will in general be the case once the model is sufficiently accurate. However, for a reference tracking problem it will in general happen that the system cannot track the reference from the very start, and as a result of that the error will not be small. So then the Gauss-Newton or Levenberg-Marquardt nonlinear least squares method will not work well.

4. NLP with Constraints

4.1. Linear Equality (Elimination)

4.2. Nonlinear Equality (Lagrange)

When it is not possible to use the constraint to eliminate one of the variables

1. Stop Criterion

$$\|\nabla f(x_k) + \nabla h(x_k)\lambda_k\|_2 \leq \epsilon_1$$

$$\|h(x_k)\|_2 \leq \epsilon_2$$

where x_k is the current iteration point

f is the objective of the minimization problem

h is the equality constraint function (when the constraint is written in the form $h(x) = 0$) ϵ_1, ϵ_2 are small positive numbers

4.3. Linear Inequality(Gradient Projection/SQP)

4.3.1. Gradient Projection Method

Need Gradient

1.

Attention: When SQP and Gradient Projection Method are both OK, use SQP Method, because the Hessian can fast the calculation

4.4. Nonlinear Inequality (SQP/Barrier/Penalty)

For Barrier and Penalty, if use search method

the f in stopping criterion should change to f_{total}

OR use KKT

$$\|\nabla f(x_k) + \nabla g(x_k)\mu_k\|_2 \leq \epsilon_1$$

$$|\mu_k^T g(x_k)| \leq \epsilon_2$$

$$\mu_k \geq -\epsilon_3$$

$$g(x_k) \leq \epsilon_4$$

where x_k is the current iteration point, f is the objective of the minimization problem, and g is the inequality constraint function (written in the form $g(x) \leq 0$) of the simplified optimization problem.

It depends on the inequality, if is easy to get gradient, then KKT, if not, f_{total}

Using the stopping criterion based on KKT is actually better than the zero gradient condition due to two reasons: the penalty function weight might not be appropriate and the penalty function approach could yield an infeasible solution. By adjusting the weight and adopting the KKT-based stopping criterion a better solution can be obtained.

4.5. SQP

Need Hessian and Gradient

1. need multi-start
2. stopping criterion: KKT

$$\begin{aligned}\|\nabla f(x_k) + \nabla g(x_k) \mu_k\|_2 &\leq \varepsilon_1 \\ |\mu_k^T g(x_k)| &\leq \varepsilon_2 \\ \mu_k &\geq -\varepsilon_3 \\ g(x_k) &\leq \varepsilon_4\end{aligned}$$

where x_k is the current iteration point, f is the objective of the minimization problem, and g is the inequality constraint function (written in the form $g(x) \leq 0$) of the simplified optimization problem.

5. Convex

5.1. Cutting Plane Method

1. Stopping Criterion:

$$\begin{aligned}|f(x_k) - f(x^*)| &\leq \epsilon_f \\ g(x_k) &\leq \epsilon_g\end{aligned}$$

where x_k is the current iteration point

x^* is the (yet unknown) optimum of this optimization problem

f denotes the objective function

g denotes the constraint function g

5.2. Ellipsoid Algorithm

1. Stopping Criterion

$$\begin{aligned}|f(x_k) - f(x^*)| &\leq \epsilon_f \\ g(x_k) &\leq \epsilon_g \\ \|x^* - x_k\|_2 &\leq \epsilon_x\end{aligned}$$

where x_k is the current iteration point

x^* is the (yet unknown) optimum of this optimization problem

f denotes the objective function

g denotes the constraint function g

- 2.

5.3. Interior Points

1. stopping criterion\

$$|f(x_k) - f(x^*)| \leq \epsilon$$

6. Global Optimal

6.1. Stopping Criterion

Last resort:

$$\|x_{k+1} - x_k\|_2 \leq \varepsilon_x, \quad |f(x_{k+1}) - f(x_k)| \leq \varepsilon_f$$

6.2. Genetic Algorithm

A slight disadvantage of the basic version of the genetic algorithm is that it requires the feasible set to be discretized, but that can be overcome in more advanced versions of the genetic algorithm.