**Min-Max Reciprocity Algorithm to Find Out All Extremum Points of Function**

Loc Nguyen

Institute of Mathematics, Hanoi, Vietnam

**Abstract**

**1. Searching local extremum points procedure**

Suppose *f* is the target function that we need to find out its extremum points with regard that *f* is scalar-by-vector function.

Where is real number field.

Given input is the initial point ***x****0*, we use and modify descent gradient method to find out maximum and minimum points according to two opposite directions. Given the *ith* iteration in descent gradient method, we have some notations as below:

* Let and be the minimum-candidate and maximum-candidate points. If initial point is ***x****0*, we have . The upper numbers *i* and *j* denotes the indices instead of exponent factors as usual. Note that and are points in .
* Let be gradient of function *f*, descent direction and ascent direction are defined as and . Hence, and are evaluations of and at and , respectively. Note that and are vectors in .
* Let and be lengths of the descent direction and ascent direction , respectively. Note that and are real number.

At the *ith* and *jth* iterations when the indices *i* and *j* starts with *0*, the extremum-candidate points and are calculated as below:

Where,

* The direction length is minimum value of expression with note that is function of variable *t*.
* The direction length is maximum value of expression with note that is function of variable *t*.

The iteration process stops when both descent and ascent directions are equal to zero or extremum-candidate points and are out of pre-defined range. Let and be the minimum point and maximum point at the *ith* iteration when the process of descent gradient method stops. However, there are some attentions when we implement this procedure in practice.

* If *f* has second-order derivative denoted , we evaluate such second-order derivative at and . If , then is saddle point and the iteration process continues. If , then is saddle point and the iteration process continues.
* If *f* has not second-order derivative, we define a very small number *ε*. If , then is not minimum point and the iteration process continues. If , then is not maximum point and the iteration process continues.

If *f* has second-order derivative, we denote is the saddle point. So the set is output of searching local extremeum points procedure whose input is the initial point ***x****0*. It is easy to recognize that we search and in two opposite directions and when the traditional descent gradient method only search minimum point according to descend direction .

We have a very important convention that the searching local extremeum points procedure can be modified so as to find out only minimum points or only maximum points. Moreover, the procedure can be extended so that its input has more than one initial point.

**2. Min-Max reciprocity algorithm**

Min-Max reciprocity algorithm is iterative method having two steps with the important support of searching local extremum points procedure. Before going inside the algorithm, we define some notations:

* Input of the algorithm is an initial point ***x****0*.
* Outputs of the algorithm are a set of minimum points denoted ***MIN***, a set of maximum points denoted ***MAX*** and a set of saddle points denoted ***SADDLE***.
* Let and be starting points for searching minimum and maximum points, respectively. Both and are initialized to be equal to ***x****0*.
* Let *S* denote the searching local extremum points procedure mentioned in previous section.
* Let *Pm* and *PM* denote processes of searching minimum and maximum points, respectively.

**Step 1**

If , then the searching procedure *S* is used to find out minimum and maximum points. If , then this procedure is used to find out minimum and maximum points separately.

* Firstly, the procedure *S* is used to find out only minimum points with initial point if the process *Pm* is not stopped.
* Secondly, the procedure *S* is used to find out only maximum points with initial point if the process *PM* is not stopped.

Let be the output of procedure *S* where , , and are minimal, maximal and saddle point, respectively.

**Step 2**

We survey extremum points , , and which are results of the searching procedure *S* in step *1*. There are two cases:

* *Case 1*: The process *Pm* is not stopped. If the set ***MIN*** has already contained or the set ***SADDLE*** has already the respective , then *Pm* is stopped; otherwise is added into ***MIN*** and is added into ***SADDLE***.
* *Case 2*: The process *PM* is not stopped. If the set ***MAX*** has already contained or the set ***SADDLE*** has already the respective , then *PM* is stopped; otherwise is added into ***MAX*** and is added into ***SADDLE***.

If both *Pm* and *PM* are stopped or both extremum points and are out of pre-defined range, then the algorithm is stopped. Otherwise, it is necessary to calculate starting points and . Suppose and are descent and ascent directions at the *ith* and *jth* iterations of procedure *S* which produces and , respectively. We define a very small number *δ* so as to calculate starting points and as following:

There are two notes when domain of *f* is the subset of two-dimension or one-dimension space:

* If the derivative of *f* at is equal ***0***, , then is also a minimum point and added into the set ***MIN***; after that is increased again by interval . The addition-increase task is continued until . If the cosine of angle between of the derivative and is positive, then becomes , which means that . In this case, the input of searching procedure *S* may have more than one initial point.
* If the derivative of *f* at is equal ***0***, , then is also a maximum point and added into the set ***MAX***; after that is increased again by interval . The addition-increase task is continued until . If the cosine of angle between of the derivative and is negative, then becomes , which means that . In this case, the input of searching procedure *S* may have more than one initial point.

We recognize that the maximum point is used to search next minimum points and the minimum point is mutually used to search next maximum points; this is main ideology of the Min-Max reciprocity algorithm. After starting points and are determined, the algorithm goes back step *1*.

**Reference**