PD 算法

Yummy

2018 年 1 月 2 日

1 Positive Definite 算法

Positive Definite (PD) 算法是用来判断一类特殊多项式正定的算法. 其本质是利用多项式正定的某些充分条件, 在多项式满足这些充分条件的情况下, 可以得到该多项式是正定的.

下面介绍我们用来判定多项式正定与半正定的充分条件.

引理 1 关于变量 $x = (x_1, \ldots, x_n)$ 的多项式

$$f(x_1, \ldots, x_n; u_1, \ldots, u_m) = \sum_{\alpha} f_\alpha(u_1, \ldots, u_m) x^\alpha.$$

其中 $u_1, \ldots, u_m > 0$ 为参数, 同时变量 $x_1, \ldots, x_n > 0$. 单项式 $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, α 称为单项式 x^α 的次数, $f_\alpha(u_1, \ldots, u_m)$ 为单项式 x^α 的系数多项式.

若对任意次数 α, 在 $\mathbb{R}_m^+ = \{(u_1, \ldots, u_m) \mid u_i > 0,\ i = 1, \ldots, m\}$ 上都有 $f_\alpha \geq 0$. 那么 $f(x_1, \ldots, x_n; u_1, \ldots, u_m)$ 在 $\mathbb{R}_n^+ = \{(x_1, \ldots, x_n) \mid x_i > 0,\ i = 1, \ldots, n\}$ 上半正定. 此时称多项式 f 为 "系数半正定多项式".

若对任意次数 α, 在 $\mathbb{R}_m^+$ 上都有 $f_\alpha \geq 0$, 且其中至少有一个严格大于零. 那么 $f(x_1, \ldots, x_n; u_1, \ldots, u_m)$ 在 $\mathbb{R}_n^+$ 上正定. 此时称多项式 f 为 "系数正定多项式".

引理 2 如果多项式 $g(u_1, \ldots, u_m)$ 可以因式分解为

$$g(u_1, \ldots, u_m) = \prod_i g_i^{\beta_i}(u_1, \ldots, u_m).$$

其中 g_i 的次数 β_i 为偶数或 g_i 为关于变量 $u_1, \ldots, u_m$ 系数正定多项式, 那么多项式 g 在 $\mathbb{R}_m^+$ 上半正定. 此时称多项式 g 为 "半正定多项式之积".

引理 3 如果多项式 $h(u_1, \ldots, u_m)$ 可以记为

$$h(u_1, \ldots, u_m) = \sum_i h_i(u_1, \ldots, u_m),$$

其中 h_i 为关于变量 $u_1, \ldots, u_m$ 系数正定多项式或为半正定多项式之积, 那么多项式 h 在 $\mathbb{R}_m^+$ 上半正定, 称之为 "半正定多项式之和".

根据引理 1, 判断关于变量 x 的参系数多项式 $f(x;u)$ 在 $\mathbb{R}_n^+$ 上是否为正定多项式, 其充分条件为该参系数多项式是否为系数正定多项式. 即, 判断其系数多项式 $f_\alpha(u)$ 在 $\mathbb{R}_m^+$ 上半正定且至少有一个正定. 这样, 就将判断原多项式正定性的问题转化为判断变量和项数较少的系数多项式正定性或半正定性的问题. 即, 判断这些系数多项式满足上述系数正定多项式、半正定多项式之积或半正定多项式之和的条件.

判断系数多项式 $f_\alpha(u)$ 在 $\mathbb{R}_m^+$ 上是否为系数正定多项式的过程十分直接. 由于其所有系数多项式都为非零常数多项式, 所以我们只需判断这些常数全大于零即说明系数多项式 $f_\alpha(u)$ 在 $\mathbb{R}_m^+$ 上系数为正定多项式.

而判定多项式是否为半正定多项式之积的过程如下: 首先, 将多项式因式分解. 然后对每个因子, 判断其次数是否为偶数. 如果是偶数, 则将其放入移除列表中; 如果是奇数, 则判断该因子的底数是否为系数正定多项式. 如果为系数正定多项式, 则放入移除列表中; 如果判断为非系数正定多项式, 则放入输出列表, 等待后续判断. 如果输出列表为空, 则输入多项式为半正定多项式之积.

我们将筛选出既不是系数正定多项式、又不是半正定多项式之积的多项式来进行下一步判断. 其过程由程序 `PDFilter` 来进行, 其伪代码如算法 1.

算法 1. `PDFilter`.

Input: 多项式或多项式列表.

Output: 输入多项式中非系数正定且非半正定多项式之积的部分.

```
begin
    提取非系数正定多项式;
    将非系数正定多项式因式分解, 提取奇数次数因子;
    提取奇数次数因子中的非系数正定多项式;
    将非系数正定多项式因子相乘, 形成候选输出多项式;
    除去上述过程中产生的系数正定多项式;
    return 剩余的多项式.
end
```

在无法判断多项式是系数正定的或是半正定多项式之积后, 我们来判断其是否为半正定多项式之和. 平方和 (Sum of Squares, SOS) 方法是用来判定多项式为半正定多项式之和的有效算法. 但对于该系统的特殊多项式, 我们找到了将其变为半正定多项式之和的特殊规律.

例如, 多项式

$$\begin{aligned} &h(u_1,u_2,u_3,u_5,v_1,v_2,v_3) = \\ &{u_1}^3{u_2}^3{v_3}^3 - {u_1}^3{u_2}^2u_3v_2{v_3}^2 - {u_1}^3u_2{u_3}^2{v_2}^2v_3 + {u_1}^3{u_3}^3{v_2}^3 + {u_2}^3{u_5}^3{v_1}^3 \\ &+ {u_2}^3{u_5}^3{v_3}^3 - {u_2}^2u_3{u_5}^3v_2{v_3}^2 - u_2{u_3}^2{u_5}^3{v_2}^2v_3 + {u_3}^3{u_5}^3{v_1}^3 + {u_3}^3{u_5}^3{v_2}^3, \quad (1) \end{aligned}$$

既非系数正定多项式, 又无法被因式分解. 但经过如下操作, 可以将其写为半正定多项式之和的形式,

$$h = {u_1}^3 \left(u_2 v_3 + u_3 v_2\right) \left(u_2 v_3 - u_3 v_2\right)^2 + {u_5}^3 \left(u_2 v_3 + u_3 v_2\right) \left(u_2 v_3 - u_3 v_2\right)^2 + {u_5}^3 {v_1}^3 \left(u_2^3 + u_3^3\right).$$

将多项式 (1) 写为半正定多项式之和的过程可以分为两个步骤: 第一步, 对多项式的各项进行分类; 第二步, 对每类分别进行因式分解.

现在的问题便是如何对多项式的各项进行分类. 其主要思想就是根据各项所含变量的不同进行分类. 但仅仅根据所含变量的个数或者变量的符号不同进行分类是不够的, 那么此时就需要根据项的 “变量特征” 来确定分类. 下面我们就来确定某些项的变量特征. 首先, 选出多项式 (1) 中变量最多的项, 即

$$-{u_1}^3 {u_2}^2 u_3 v_2 {v_3}^2,\ -{u_1}^3 u_2 {u_3}^2 {v_2}^2 v_3,\ -{u_2}^2 u_3 {u_5}^3 v_2 {v_3}^2,\ -u_2 {u_3}^2 {u_5}^3 {v_2}^2 v_3.$$

然后根据其含有变量的不同分为两类, 即

$$[-{u_1}^3 {u_2}^2 u_3 v_2 {v_3}^2,\ -{u_1}^3 u_2 {u_3}^2 {v_2}^2 v_3],\ [-{u_2}^2 u_3 {u_5}^3 v_2 {v_3}^2,\ -u_2 {u_3}^2 {u_5}^3 {v_2}^2 v_3].$$

对上述两类的元素求和后再因式分解, 可以得到

$$-{u_1}^3 u_2 u_3 v_2 v_3 \left(u_2 v_3 + u_3 v_2\right),\ -u_2 u_3 {u_5}^3 v_2 v_3 \left(u_2 v_3 + u_3 v_2\right).$$

这时, 我们发现, 每个式子中都有一个和式因子, 且该和式因子包含两个项, 分别含有变量,

$$\{u_2, v_3\},\ \{u_3, v_2\}.$$

除了这两组变量外, 第一类还含有变量 $\{u_1\}$, 第二类还含有变量 $\{u_5\}$. 综合上述变量集合, 两个类分别含有如下变量集合,

$$[\{u_2, v_3\},\ \{u_3, v_2\},\ \{u_1\}],\ [\{u_2, v_3\},\ \{u_3, v_2\},\ \{u_5\}].$$

上述两个变量集合列表便称为对应类的变量特征.

在得到这两个类的变量特征后, 接下来就是判断剩余项是否属于其中某类, 进而与该类元素构成和式并因式分解为半正定多项式之积的形式. 多项式 (1) 的剩余项为,

$${u_1}^3 {u_2}^3 {v_3}^3,\ {u_1}^3 {u_3}^3 {v_2}^3,\ {u_2}^3 {u_5}^3 {v_1}^3,\ {u_2}^3 {u_5}^3 {v_3}^3,\ {u_3}^3 {u_5}^3 {v_1}^3,\ {u_3}^3 {u_5}^3 {v_2}^3.$$

对于第一余项 ${u_1}^3 {u_2}^3 {v_3}^3$, 其含有的变量集合为 $V = \{u_1, u_2, v_3\}$. 通过下述操作来判断其是否属于第一类. 首先, 判断第一类的变量特征的第一个变量集合 $\{u_2, v_3\}$ 是否属于 V. 可以知道其确实属于 V, 那么从集合 V 中除去变量集合 $\{u_2, v_3\}$, V 变为 $\{u_1\}$. 然后判断第一类的变量特征的第二个变量集合 $\{u_3, v_2\}$ 是否属于 V. 我们知道其不属于, 那么接下来判断第三个变量集合 $\{u_1\}$ 是否属

于 V. 可以知道其确实属于 V, 那么再从 V 中除去变量集合 $\{u_1\}$. 此时该余项的变量集合 $V=\emptyset$, 那么我们称余项的变量集合属于第一类的变量特征, 从而将第一余项划分到第一类去.

同样地, 可以知道第二余项 ${u_1}^3{u_3}^3{v_2}^3$ 的变量集合在经历上述过程后也为空集, 所以其也属于第一类. 而第三余项 ${u_2}^3{u_5}^3{v_1}^3$ 的变量集合 $\{u_2,u_5,v_1\}$ 既不包含第一类的变量特征中的 $\{u_2,v_3\}$, 又不包含 $\{u_3,v_2\}$ 和 $\{u_1\}$, 因此其不属于第一类. 同样的其也不属于第二类, 所以将其单独成类. 同理对于第五余项, 其也归属第三类. 而对于第四和第六余项, 容易验证其属于第二类. 从而我们得到多项式 (1) 各项的如下划分,

$$\begin{aligned}&[[-{u_1}^3{u_2}^2u_3v_2{v_3}^2,-{u_1}^3u_2{u_3}^2{v_2}^2v_3,{u_1}^3{u_2}^3{v_3}^3,{u_1}^3{u_3}^3{v_2}^3],\\&[-{u_2}^2u_3{u_5}^3v_2{v_3}^2,-u_2{u_3}^2{u_5}^3{v_2}^2v_3,{u_2}^3{u_5}^3{v_3}^3,{u_3}^3{u_5}^3{v_2}^3],\\&[{u_2}^3{u_5}^3{v_1}^3,{u_3}^3{u_5}^3{v_1}^3]].\end{aligned}$$

综上, 我们就是通过判断余项的变量集合是否属于某类的变量特征来划分各项所属的类别. 该判断过程的伪代码如算法 2.

算法 2. `IsInTheItemClass`.

Input: 某项, 某类的变量特征.

Output: `True`: 如果某项的变量集合属于变量特征, `False`: 如果变量集合不属于变量特征.

```
begin
    V := 提取某项的变量集合;
    for i in 变量特征 do
        if i 属于 V then
            V := V \ i;
        end
        if V = ∅ then
            return True.
        end
    end
    return False.
end
```

在完成对多项式各项的分类后, 接下来对每类分别求和后因式分解、再求和便得到了我们之前给出的半正定多项式之和的形式. 当然, 该过程可能需要进行多次才能将多项式写成半正定多项式之和的形式. 对多项式各项的分类由程序 `ItemClassByGreat` 实现, 其伪代码如算法 3.

通过上述半正定多项式之和的处理, 对每个部分和再进行既非系数正定多

算法 3. `ItemClassByGreat`.

Input: 多项式.

Output: 多项式项的分类.

```
begin
    提取多项式的各项;
    选出各项中变量最多的项;
    根据含有变量的不同对变量最多项进行分类;
    求取上述各类的变量特征;
    for i in 余项列表 do
        for j in 分类列表 do
            if IsInTheItemClass(i, j 的变量特征) then
                加入该分类;
            else
                留在余项类;
            end
        end
    end
    return 所有分类与余项类.
end
```

项式、又非半正定多项式之积的筛选, 最后输出未通过筛选的部分和. 结合上述程序 `PDFilter` 和 `ItemClassByGreat`, 我们对输入多项式作筛选和化简, 最终输出那些无法判定为系数正定多项式、半正定多项式之和或积的多项式. 该过程由程序 `PDSimplify` 实现, 其伪代码为算法 4.

最终, 判断关于变量 x 的参系数多项式 $f(x;u)$ 在 $\mathbb{R}_n^+$ 上是否为系数正定多项式, 我们需要判断其系数多项式 $f_\alpha(u)$ 在 $\mathbb{R}_m^+$ 上半正定且至少有一个正定. 因此, 根据上述判定正定性算法, 我们编写了一个程序, 其输入为参系数多项式 $f(x;u)$ 与变量 x. 在提取其指定变量 x 的系数多项式 $f_\alpha(u)$ 之后, 我们判断这些系数多项式的正定性, 筛选出那些无法通过前述过程判断其正定性的系数多项式. 如果没有任何系数多项式的输出, 则该参系数多项式 $f(x;u)$ 在 $\mathbb{R}_n^+$ 上正定. 这种判定多项式正定的过程我们称之为 Positive Definite (`PD`) 算法, 其伪代码如算法 5.

A Maple 程序

上述程序可在 Github 网站获取.

https://github.com/woshiYummy/IsDDLVSysGloballyStable.git

算法 4. PDSimplify.

Input: 多项式或多项式列表.

Output: 既非系数正定多项式, 又非半正定多项式之和或积的多项式.

begin
- 筛选出既非系数正定多项式、又非半正定多项式之积的多项式 (PDFilter);
- 对筛选出的每个多项式进行项的分类 (ItemClassByGreat);
- 对每类里的项求和后进行既非系数正定多项式、又非半正定多项式之和的筛选 (PDFilter);
- 将筛选出来的每类的部分和汇总求和, 形成候选多项式;
- 除去上述过程中产生的系数正定多项式 (PDFilter);
- **return** 剩余的多项式.

end

算法 5. PD.

Input: 参系数多项式 $f(x;u)$, 指定变量 x.

Output: 不能判定正定性的系数多项式.

begin
- 提取参系数多项式 $f(x;u)$ 关于变量 x 的系数多项式 $f_\alpha(u)$;
- 对系数多项式 $f_\alpha(u)$ 形成的多项式列表, 使用两次 (或更多次) PDSimplify 对其进行筛选和化简;
- **return** 不能判定正定性的系数多项式.

end