**Content**

[1. Define program in Linear Algebra 1](#_Toc505021885)

[2. Improvement method for computing the least model 4](#_Toc505021886)

[References 7](#_Toc505021887)

# 1. Define program in Linear Algebra

A *define program* is a finite set of *rules* of the form:

 (1)

Where h, bi are propostional variables.

Let Pbe a define program.

A set *I* ⊆ BPis an *interpretation* of P.

A mapping is defined as TP(*I*) = 

A rule r is called *d-rule* iff r has the form:

 (2)

where h, bi are propostional variables.

For each rule r of the form (1) and (2), the left-hand side of ← is the *head* and the right-hand side is the *body*:

*head*(*r*) = *h* and *body*(*r*) = *{b*1*,…, bm}*

In particular, the rule (1) is a *fact* if body (r) = {}.

A program P is called singly defined (or SD-program):

*head(r1)≠ head(r2) for any two rules r1 and r2 in P*

The set BP is the *Herbrand base* of P.

**Proposition 1 [1]:** Any define program P with Herband base BP = {*p*1*,…,pn*} can be transformed to the program P’ = Q ∪ D with Q is a SD-program and D is a set of d-rules.

Let BP’ = {*p*1*,…,pn, pn*+1*,…,pn’*}

We call program P’ as the *positive program.*

**Definition 1.1:** **(vector representation of interpretations)**

Let P be a define program and BP = {p1,…, pn} be the Herband base. Then an interpretation I of Pis represented by a column vector *v*= (*a*1*, . . . , an*)T ∈ n, where each element *ai* represents the truth value of the proposition pisuch that *ai* = 1 if pi ∈ *I* (1 *≤ i ≤ n*); otherwise, *ai* = 0.

We write row*i*(*v*) = *pi*.

A vector *v*0 representing a facts in P is defined as ***v*0**= (*a*1*, . . . , an*)T where *ai* = 1 iff pi ← (1 *≤ i ≤ n*); otherwise, *ai* = 0.

We denote *v*[1…*k*]= (*a*1*, . . . , ak*)T ∈ *k*

**Definition 1.2: (matrix representation of positive programs)**

Let a define program P with Herband base BP = {*p*1*,…,pn*} and P is transformed to positive program *P’* = Q ∪ D with Q is a SD-program and D is a set of d-rules, BP’ = {*p*1*,…,pn, pn*+1*,…,pn’*}. Then *P’* is represented by a matrix such that for each element *aij* (1 *≤* *i, j* *≤* *n’*) in *MP’*,

1. 

2. 

3. *aii* = 1 if *pi* ← is in Q.

4. Otherwise, *aij* = 0.

We write rowi(*MP’*) = pi and colj(*MP’*) = pj.

**Definition 1.2: (thresholding function)**

Given a vector *v*= (*a*1*,…, an*)T, define a vector *w*= θ(*v*) = (*w*1*,…, wn* )T where:

• if *ai* < 1 (1 ≤ *i* ≤ *n*): *wi* = 0;

• if *ai* ≥ 1 (1 ≤ *i* ≤ *n*): *wi* = 1.

θ is called a thresholding function.

Given a matrix *MP’* ∈ n’×n’, define *v*1 = θ(MP’.*v*0)

*v*2 = θ(MP’.*v*1)

*v*k+1 = θ(MP’.*v*k)

**Proposition 1.2:** ∃k, *v*k+1 = *v*k for some k ≥ 1

When *v*k+1 = *v*k we write *v*k = FP(MP.*v*0)

**Theorem 1 [2]**: Let P’ be a positive program and  its matrix representation. Then  is a vector representing the least model of P iff *m* = FP(MP.*v*0) where is the vector representing facts in P’.

Example 1: Consider *P* = {*p* ← *q*, *p* ← *r* ∧ *s*, *r* ← *s*, *s* ← } with *BP* = {*p*, *q*, *r*, *s*} then:

P will be transformed to positive program P’ = Q ∪ D with *BP’* = {*p,q,r,,s,t, u*}:

Q = { *t* ← *q*, *u* ← *r* ∧ *s, r* ← *s*, *s* ← }

 D = { *p* ← *t* ∨ *u*}

We have the matrix *MP’* ∈ 6×6represents program P’:

We have *v*0 = (0 0 0 1 0 0)T is a vector represents its facts.

*v*1 = θ(MP’.*v*0) = (0 0 1 1 0 0)T

*v*2 = θ(MP’.*v*1) = (0 0 1 1 0 1)T

*v*3 = θ(MP’.*v*2) = (1 0 1 1 0 1)T

*v*4 = θ(MP’.*v*3) = (1 0 1 1 0 1)T = *v*3

Then vector *v*3 is a vector represents the least model of P’, it represents {*p, r, s, u*}

# 2. Improvement method for computing the least model

**Definition 2.1: (matrix representation of positive programs)**

Let a define program P with Herband base BP = {*p*1*,…,pn*} and P is transformed to positive program *P’* = Q ∪ D with Q is a SD-program and D is a set of d-rules, BP’ = {*p*1*,…,pn, pn*+1*,…,pn’*}. Then *P’* is represented by a matrix such that for each element *aij* (1 *≤* *i, j* *≤* *n’*) in *MP’*,

1. 

2. *aii* = 1 if *pi* ← is in Q.

3. Otherwise, *aij* = 0.

We write rowi(*MP’*) = pi and colj(*MP’*) = pj.

**Definition 2.2: (improved thresholding function)**

Let a define program P with Herband base BP = {*p*1*,…,pn*} and P is transformed to positive program *P’* = Q ∪ D with Q is a SD-program and D is a set of d-rules, BP’ = {*p*1*,…,pn, pn*+1*,…,pn’*}.

Given a vector *v*= (*a*1*,…, an’*)T Define a vector *w*= θD(*v*) = (*w*1*,…, wn’* )T where:

• if *ai* < 1 (1 ≤ *i* ≤ *n’*): *wi* = 0;

• if *ai* ≥ 1 (1 ≤ *i* ≤ *n’*): *wi* = 1

And if ∃*d*∈D, row*i*(*w*) ∈ body(*d*) then:

*wj* = 1 with head(*d*) = *pj*

θD is called a thresholding function on positive program P’.

By definition of θD, we have: θD(*v*) = θD(θ(*v*)).

**Theorem 2**: Let a define program P with Herband base BP = {*p*1*,…,pn*} and P is transformed to positive program *P’* = Q ∪ D with Q is a SD-program and D is a set of d-rules, BP’ = {*p*1*,…,pn, pn*+1*,…,pn’*}.

Give *v* ∈ *n* is a vector represents an interpretation *I* of *P*, and *u*:= θD(*M1*.*v*) ∈ *n’*.

Then *u* is a vector represents an interpretation *J* of *P’* and *J* ∩ BP = TP(*I*)

*Proof:*

Let then with  (1 ≤ *k* ≤ *n’*)

+ Suppose 

and *u* = θD(*M1*.*v*) = θD(*w*) = (*u*1*, …, un’*)T

• Prove: J ∩ BP ⊆ TP(I)

Let *uk* = 1 (1≤ *k* ≤ *n*) and *pk* = row*k*(*u*). We prove: *pk* ∈ T*p*(*I*)

1. Case 1: If *wk* = *uk* = 1: Because *wk* = 1, so 

Let  such that *bi* ≠ 0 (1 ≤ *m* ≤ *n*). We have:

*bi* = 1/*m* (1 ≤ *i* ≤ *m*) and 

Then there is a rule: *pk* ← *pk1* ∧ … ∧ *pkm* in P such that *pki* = col*j*(*M*1) for *bi* = *akj* (1 ≤ *i* ≤ *m*) and implies *pk* ∈ T*P*(*I*)

We have *pk* ∈ T*P*(*I*)

1. Case 2: if *uk* ≠ *wk* : then *wk* = 0

Because *wk* = 0, by definition of θD, so:

∃ *ko*, *n* + 1 ≤ *ko* ≤ *n’*, *wko* = 1

and ∃*dko* ∈ D, row*ko*(*w*) = *pko* ∈ body(*dko*) and head(*dko*) = *pk*

. *dko*has the form:

*pk* ← *pko1* ∨…∨ *pkoq* with *pko* ∈{*pko1*,…, *pkoq*} ⊆ BP’\BP

. *wko* = 1 (*n* ≤ *ko* ≤ *n’*): 

Let  such that *bi* ≠ 0 (1 ≤ *m* ≤ *n*). We have:

*bi* = 1/*m* (1 ≤ *i* ≤ *m*) and 

Then there is a rule: *pko* ← *pk1* ∧ … ∧ *pkm* in Q

such that *pki* = col*j*(*M*1) for *bi* = *akj* (1 ≤ *i* ≤ *m*) and 

By transforming definite program P to positive program P’, we have:

*pk* ← *pk*1 ∧ … ∧ *pkm* ∈ P and {*pk1*,…, *pkm*} ⊆ I

Hence, *pk* ∈ Tp(*I*)

In both case 1 and case 2, we have:

*pk*= row*k*(*u*) ∈ Tp(*I*) if *uk*= 1 (1 ≤ *k* ≤ *n*)

Then J ∩ BP ⊆ TP(*I*)

• Prove: TP(*I*) ⊆ J ∩ BP

We prove: if *pk*∈ TP(*I*) then *uk*= 1 (1 ≤ *k* ≤ *n*)

*pk*∈ TP(*I*), so ∃ {*pk1*,…, *pkm*} ⊆ *I* and *pk* ← *pk*1 ∧ … ∧ *pkm* ∈ P (1 ≤ *m* ≤ *n*)

{*pk1*,…, *pkm*} ⊆ *I* , so *vkj* = 1 (1 ≤ *j* ≤ *m*)

+ If *pk* ← *pk*1 ∧ … ∧ *pkm* ∈ Q then:

*pki* ∈ BP ⇒ ∃ *j*, *pki* = col*j*(*M*1) (1 ≤ *i* ≤ *m*) and *akj*= 1/*m* (1 ≤ *j* ≤ *n*)

Hence,  = 1 ⇒ *wk* = 1 = *uk*

+ If *pk* ← *pk*1 ∧ … ∧ *pkm* ∉ Q then:

∃ *ko*, *n* + 1 ≤ *ko* ≤ *n’*, *pko* ← *pk1* ∧ … ∧ *pkm* ∈ Q

And *pk* ← *pko1* ∨…∨ *pkoq* ∈ D with *pko* ∈{*pko1*,…, *pkoq*}

. *pko* ← *pk1* ∧ … ∧ *pkm* ∈ Q then:

*pki* ∈ BP ⇒ ∃ *j*, *pki* = col*j*(*M*1) (1 ≤ *i* ≤ *m*): *akoj*= 1/*m* (1 ≤ *j* ≤ *n*)

So,  = 1 ⇒ *wko* = 1

Because *pk* ← *pko1* ∨…∨ *pkoq* ∈ D and row*ko*(*w*) = *pko* ∈{*pko1*,…, *pkoq*}

So *uk* = 1 (by definition of θD)

We have: if *pk*∈ TP(*I*) then *uk*= 1 (1 ≤ *k* ≤ *n*) ⇒ TP(*I*) ⊆ J ∩ BP

Hence: *J* ∩ BP = TP(*I*).

Given a matrix *M1* ∈ n’×n, define *v*1 = θD(*M1*.*v*0[1… *n*])

*v*2 = θD(*M1*.*v*1[1… *n*])

*vk*+1 = θD(*M1*. *vk*[1… *n*])

**Proposition 2.1:** ∃*k*, *vk*+1 = *vk* for some *k* ≥ 1

When *vk*+1 = *vk* we write *vk* = FP(*M1*.*v*0[1…*n*])

Note: With the same positive program *P’*, The *k* in prosition 2.1 will not greater than the *k* in proposition 1.2

Example 2: With define program P as example 1 we have:

P’ = Q ∪ D with *BP’* = {*p,q,r,,s,t, u*}:

 Q = { *t* ← *q*, *u* ← *r* ∧ *s, r* ← *s*, *s* ← }

D = { *p* ← *t* ∨ *u*}

We have the matrix *M1* ∈ 6×4represents program P’:

We have *v*0 = (0 0 0 1 0 0)T is a vector represents its facts.

*v*1 = θD(*M1*.*v*0[1…4]) = (0 0 1 1 0 0)T

*v*2 = θD(*M1*.*v*1[1…4]) = (1 0 1 1 0 1)T

*v*3 = θD(*M1*.*v*2[1…4]) = (1 0 1 1 0 1)T = *v*2

Then vector *v*2 is a vector represents the least model of P’, it represents {*p, r, s, u*}



In matrix *M1,* we can remove *p*-row and *q*-row:

# References

[1] Chiaki Sakama, *Some techniques for matrix computation*, Jan. 2018.

[2] C. Sakama, K. Inoue, and T. Sato. *Linear algebraic characterization of logic programs*. Proc. KSEM, LNAI 10412, pp. 520–533 (2017)