

คู่มือการใช้งานโปรแกรม XCS

1 การคอมไพล์โปรแกรม

ในการ compile บนระบบปฏิบัติการ Unix นั้นจะใช้คำสั่งดังนี้

`icc -o <ชื่อไฟล์ที่ต้องการใช้ run> *.cpp`

```
pornthep@barossa:~/program/XCS_IDS
[pornthep@barossa XCS_IDS]$ ls
Action.cpp      crossover.h      declare.h        HybRep.cpp      lcs.cpp          population.h     t
action.h        DatafromFiles.cpp environment.cpp   hybRep.h        lcs.h            Random.cpp      U
batch_command_off2.dat datafromfiles.h ftp-2000         init.h          main.cpp         random.h        u
classifier.h     data.h          ftp-800         input1.dat      main.h           result          u
compile.bat     DataOnfly.cpp   GA.cpp          input.cpp       param.h          runu.bat        u
Crossover.cpp   dataonfly.h     ga.h            input.h         Population.cpp    submit.dat      U
[pornthep@barossa XCS_IDS]$ icc -o main.exe *.cpp
[pornthep@barossa XCS_IDS]$ ls
Action.cpp      crossover.h      declare.h        HybRep.cpp      lcs.cpp          Population.cpp   submi
action.h        DatafromFiles.cpp environment.cpp   hybRep.h        lcs.h            population.h     test.
batch_command_off2.dat datafromfiles.h ftp-2000         init.h          main.cpp         Random.cpp      UCSCl
classifier.h     data.h          ftp-800         input1.dat      main.exe        random.h        ucs_c
compile.bat     DataOnfly.cpp   GA.cpp          input.cpp       main.h           result          ucs_c
Crossover.cpp   dataonfly.h     ga.h            input.h         param.h          runu.bat        ucs.h
[pornthep@barossa XCS_IDS]$
```

รูปที่ 1 การ compile โปรแกรม XCS

2 คำสั่งในการใช้งาน

ในการใช้งานโปรแกรมจะเรียกผ่าน command โดยมีพารามิเตอร์ต่างๆ ดังนี้

1. จำนวนของกฎใน Population
2. Crossover rate
3. Mutation rate
4. Covering spread (สำหรับ parameter ที่เป็น real value)
5. mutation spread (สำหรับ parameter ที่เป็น real value)
6. representation type 2 เป็น hyper re
7. condition noise rate
8. ation noise rate
9. มี noise หรือไม่ (0 ไม่มี noise และ 1 มี noise)

10. รูปแบบของการเรียนรู้ เป็นการเรียนรู้แบบ online หรือ off-line
11. รูปแบบของปัญหา (onfly เป็นการ generate ปัญหาจากฟังก์ชัน, files เป็นการอ่านข้อมูลจากไฟล์ที่มีอยู่แล้ว)
12. จำนวนรอบการเรียนรู้
13. ชนิดของปัญหาในที่นี้คือ Multiplexer
14. ลักษณะของปัญหา เป็น binary หรือ real
15. จำนวน address bit ของ Multiplexer
16. ไฟล์ที่กำหนดพารามิเตอร์ต่าง

```
pornthep@barossa: ~/program/XCS_IDS
[pornthep@barossa XCS_IDS]$ ./main.exe 1000 0.8 0.04 0.6 0.1 2 0.1 0 0 online onfly 10000 MUX binary 4 input1.dat
Write to file 0 0
Rd: 0, Sum: 0, 1 1
Cl[0]: nan, Cl[1]: 0,
Rd: 50, Sum: 0.14, 45 45
Cl[0]: 0.153846, Cl[1]: 0.12,
Rd: 100, Sum: 0.36, 86 91
Cl[0]: 0.282609, Cl[1]: 0.218182,
Rd: 150, Sum: 0.26, 124 138
Cl[0]: 0.28125, Cl[1]: 0.229885,
Rd: 200, Sum: 0.5, 160 177
Cl[0]: 0.356322, Cl[1]: 0.280702,
Rd: 250, Sum: 0.48, 189 221
Cl[0]: 0.409091, Cl[1]: 0.297872,
Rd: 300, Sum: 0.6, 222 268
Cl[0]: 0.481752, Cl[1]: 0.310976,
Rd: 350, Sum: 0.44, 248 308
Cl[0]: 0.503145, Cl[1]: 0.307292,
Rd: 400, Sum: 0.54, 280 354
```

รูปที่ 2 การใช้งานโปรแกรม XCS

3 ผลลัพธ์ที่ได้จากโปรแกรม

จากตัวอย่าง จะแสดง

ลำดับของการเรียนรู้ <Rd: >, ค่าความถูกต้องโดยรวมของระบบ <Sum: >,

จำนวนของกฎที่มีโดยไม่ซ้ำ, จำนวนของกฎที่มีอยู่จริง,

ค่าความถูกต้องของ class i <Cl[i]: > (i เป็นจำนวนนับที่แสดงถึงลำดับของ class ต่างๆ ของปัญหา)