

C4.5 Source Code


안재원

목차

- header.h
- getopt.h
- getnames.h
- getdata.h
- stats.h
- prune.h
- besttree.h

01

header.h

- `void PrintHeader(char *Title)`
 - C4.5 프로그램의 타이틀을 출력합니다.
 - 출력내용
C4.5 [release 버전] 타이틀 시간 -----....
- 

02

getopt.h

- `int getopt(Argc,Argv,Str) int Argc; char **Argv,*Str`

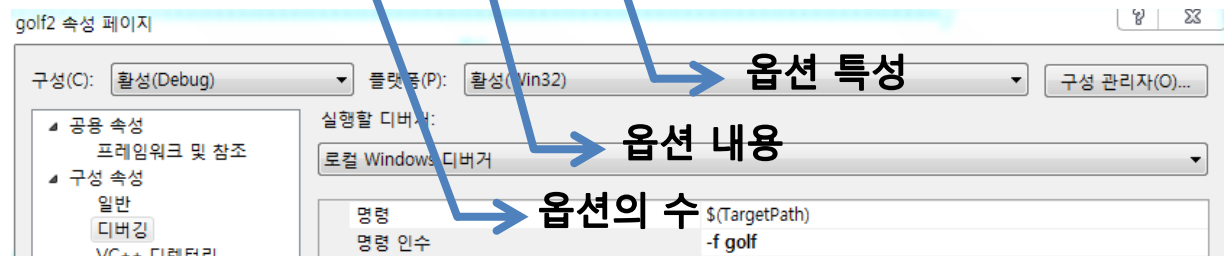
- C4.5 동작을 위한 옵션과 내용을 확인합니다.

- C4.5.c에서 호출.

전역변수 `char *optarg` 사용.

`int`형으로 반환.

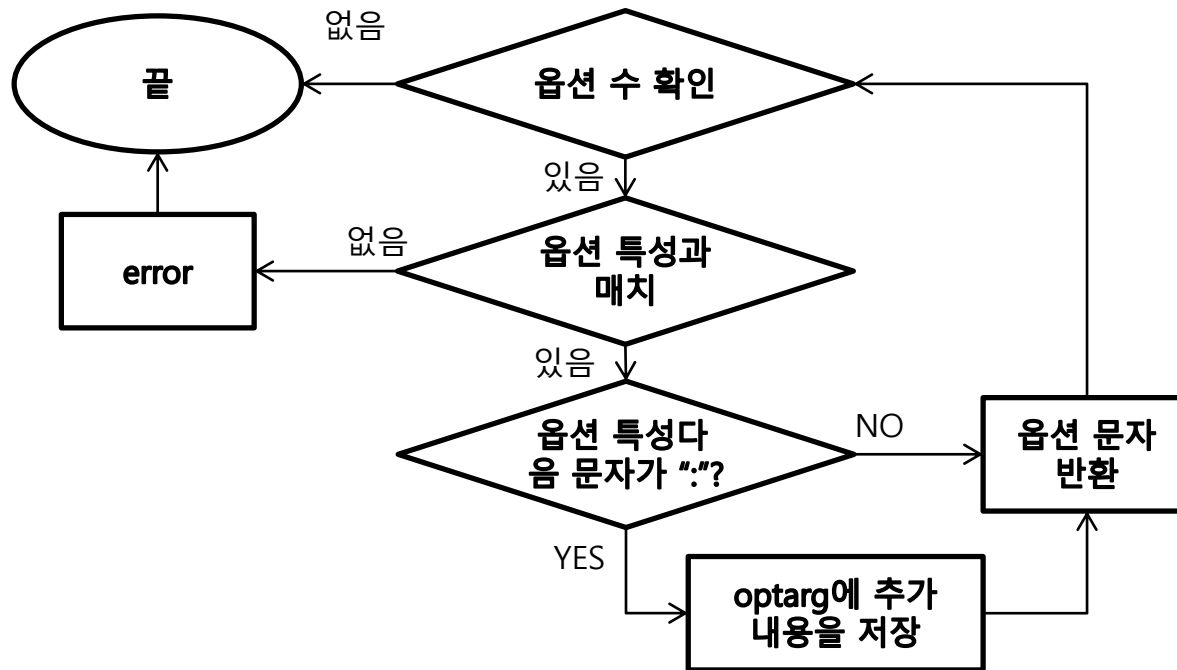
`O = getopt(Argc, Argv, "f:bupv:t:w:i:gsm:c")`



02

getopt.h

- `int getopt(Argc,Argv,Str) int Argc; char **Argv,*Str`



03

getnames..h

- **Boolean ReadName**(File *f, String s)
 - *.names파일의 내용을 단위 별로 읽는 함수 입니다.
- **void GetNames**()
 - *.names파일의 전체 내용을 읽는 함수 입니다.
- **String CopyString**(String x)
 - x의 내용을 한 사이즈 큰 메모리에 저장하는 함수 입니다.
- **int Which**(String Val, String List[], Short First, Short Last)
 - Val의 값이 List[]안에 있는지 없는지 찾는 함수 입니다.
- **void Error**(Short n, String s1, String s2)
 - 파일을 읽는 과정에서 발생하는 에러결과를 출력하는 함수 입니다.

03 getnames..h

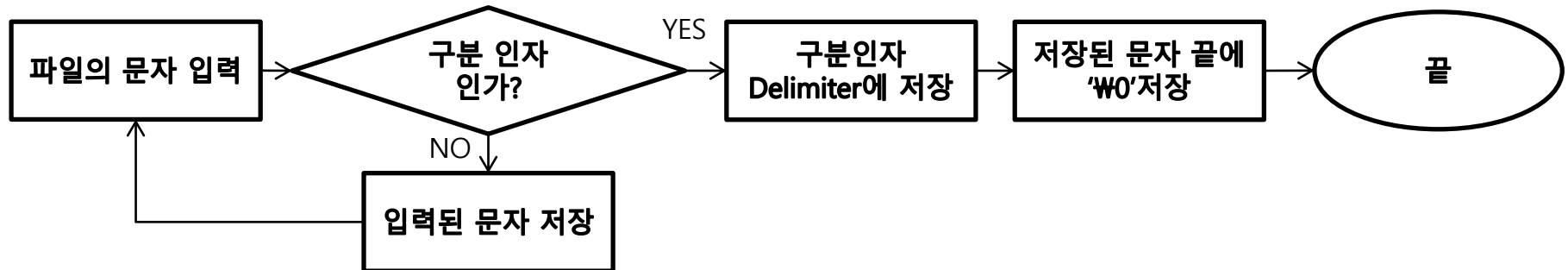
- **Boolean ReadName**(File *f, String s)
 - 전역변수 **char** Delimiter에 구분인자를 저장하고, s에는 name이 저장됩니다.

*.names파일 형식

Class name 1, Calss name 2.

Attribute name 1: Value name 1, Value name 2, Value name 3.

Attribute name 2: continuous.



03 getnames..h

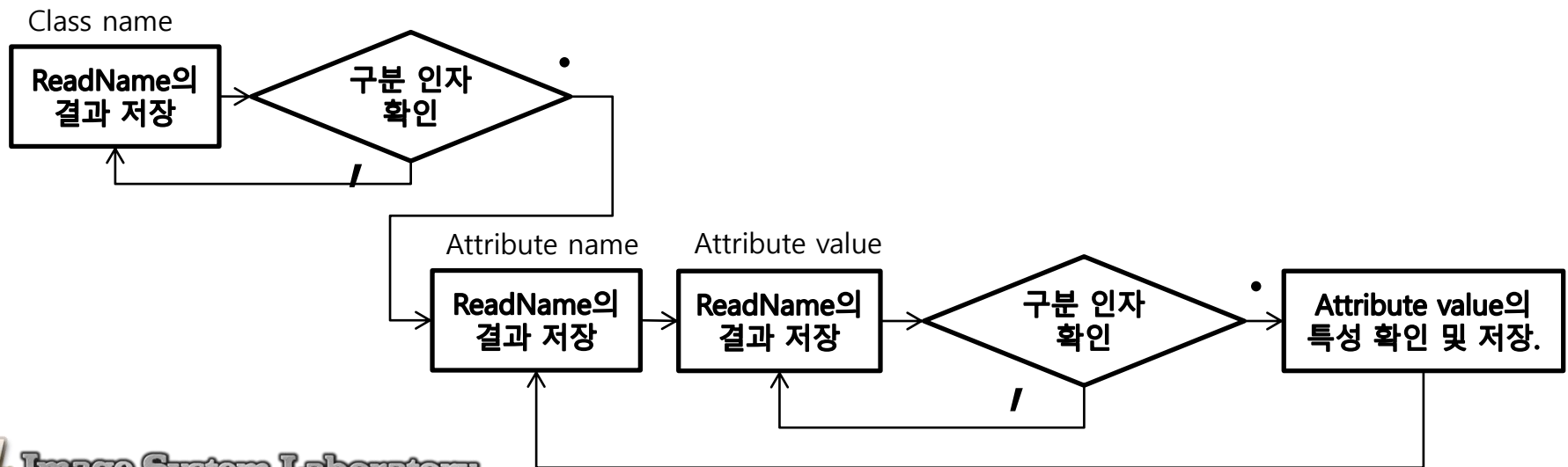
- **void GetNames()**
 - *.names파일의 내용을 전역변수 ClassName, MaxClass, AttName, MaxAttVal, AttValName, SpecialStatus, MaxAtt, MaxDiscrVal에 저장하는 함수 입니다.

*.names파일 형식

Class name 1, Calss name 2.

Attribute name 1: Value name 1, Value name 2, Value name 3.

Attribute name 2: continuous.



04 getdata.h

- **Description** **GetDescription(FILE *Df)**
 - 실질적인 data를 읽는 함수로 Attribute특성에 따라 Description 값을 반환하는 함수 입니다.

*.data파일

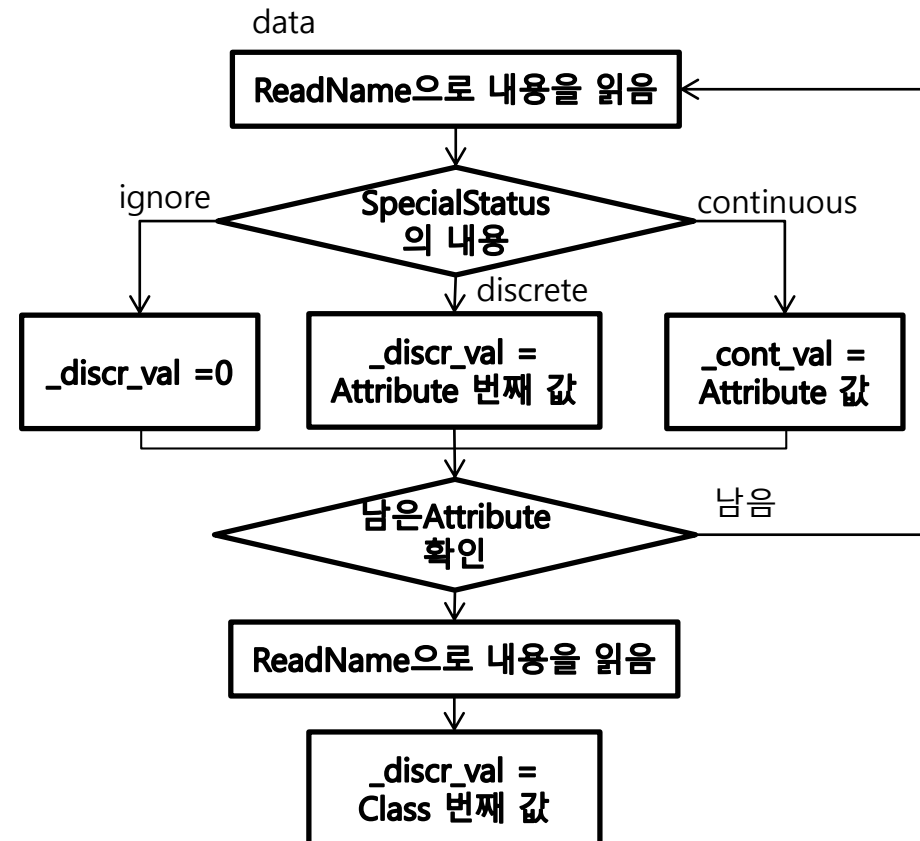
Value name 2, 55, Class name 1

Value name 3, 60, Class name 1

Value name 2, 66, Class name 2

Description

```
typedef union _attribute_value
{
    DiscrValue_discr_val;
    float_cont_val;
}
AttValue, *Description;
```



04 getdata.h

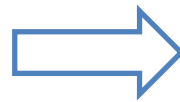
- **void GetData(String Extension)**
 - Extension 확장자를 갖는 파일의 내용을 읽는 함수 입니다.
 - 읽어온 data를 전역변수 Item과 MaxItem에 저장합니다.

*.data파일

Value name 2, 55, Class name 1

Value name 3, 60, Class name 1

Value name 2, 66, Class name 2



Item[0] -> _discr_val = 1
_cont_val = 55
_discr_val = 0

Item[1] -> _discr_val = 2
_cont_val = 60
_discr_val = 0

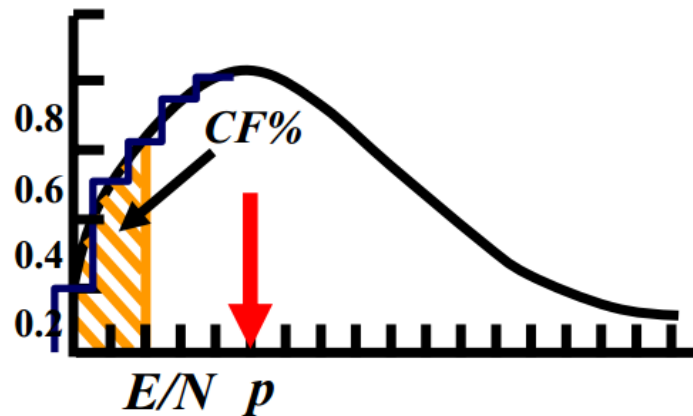
Item[2] -> _discr_val = 1
_cont_val = 66
_discr_val = 1

Description

```
typedef union _attribute_value
{
    DiscrValue_discr_val;
    float_cont_val;
}
AttValue, *Description;
```

05 stats.h

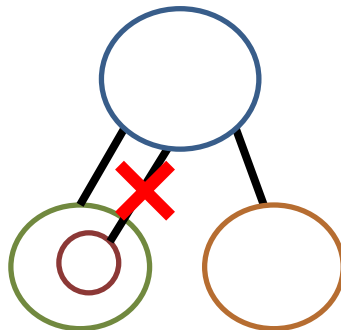
- `float AddErrs(N, e) itemCount N, e ;`
 - Additional errors을 연산하는 함수 입니다.
 - ExtraLeafErrors를 구할 때 사용합니다.
 - Confidence Level에 따라 다른 계수를 사용해 연산합니다.



06

prune.h

- **Boolean Prune**(Tree T)
 - Tree의 에러를 추정하고, 변경여부를 반환 합니다.
- **float EstimateErrors**(Tree T, ItemNo Fp, ItemNo Lp, short Sh, Boolean UpdateTree)
 - Tree의 에러를 추정하는 함수 입니다.
- **void CheckPossibleValues**(Tree T)
 - 불필요한 subset tests를 제거하는 함수 입니다.

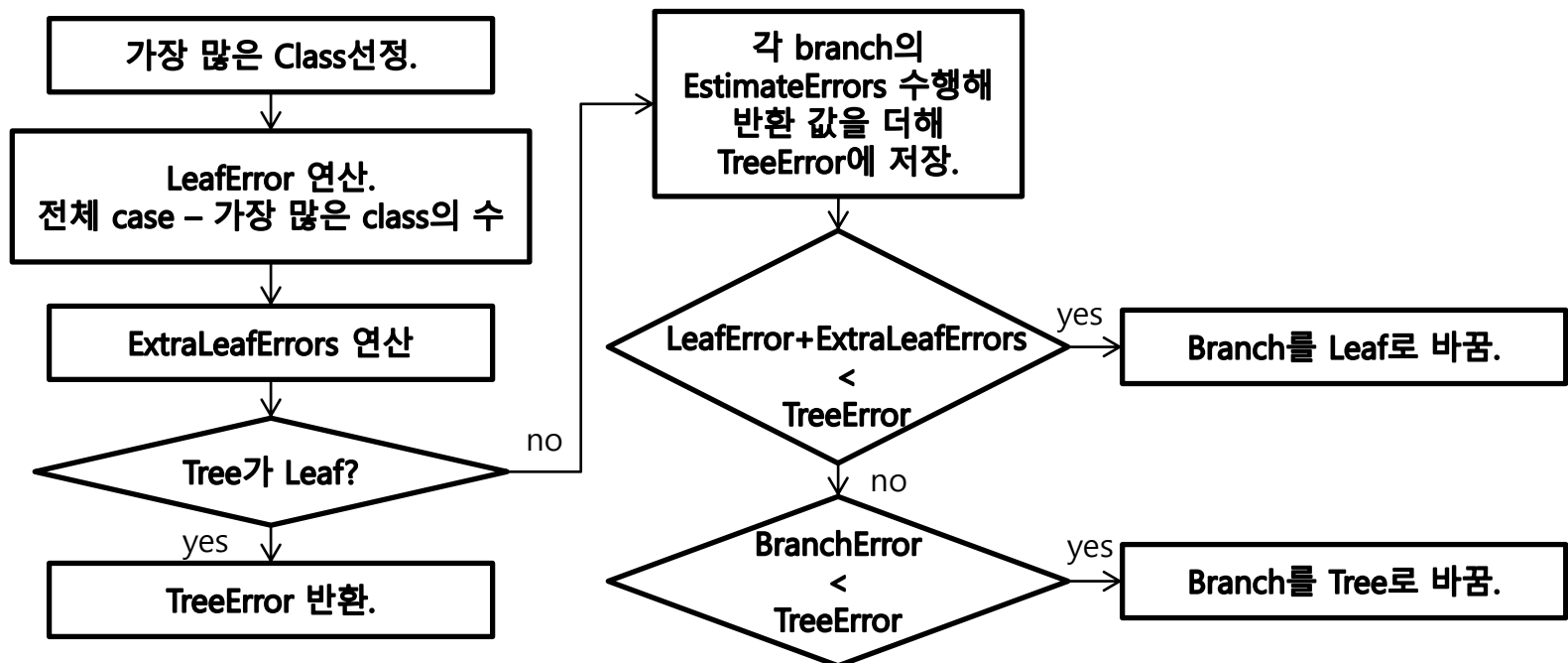


06

prune.h

- `float EstimateErrors(Tree T, ItemNo Fp, ItemNo Lp, short Sh, Boolean UpdateTree)`

- 최종적으로 정해지는 TreeError는 LeafError과 ExtraLeafErrors의 합 입니다.



07

besttree.h

- `void OneTree()`
 - 하나의 Tree를 생성하고 출력 및 저장하는 함수입니다.
- `short BestTree()`
 - 가장 적은 error값을 갖는 Tree를 선택하는 함수 입니다.
- `Tree Iterate(ItemNo Window,
ItemNo IncExceptions)`
 - BestTree()에서 Tree를 읽을 때 사용하는 함수입니다.

07

besttree.h

- **void Shuffle()**
 - data내용이 들어있는 Item의 순서를 임의로 섞는 함수 입니다.
- **void FormTarget(ItemNo Size)**
 - 각 class의 TargetClassFreq를 구하는 함수 입니다.
- **void FormInitialWindow()**
 - Item을 같은 class끼리 붙어 있도록 섞는 함수입니다.
- **void Evaluate(Boolean CMInfo, short Saved)**
 - 각 trials의 errors 리포트를 출력하는 함수입니다.

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
8	0< 0.0%>	8	0< 0.0%>	<38.5%> <<

감사합니다.
