ID3 does not guarantee to construct an optimal Decision Tree

Md Abdul Kader

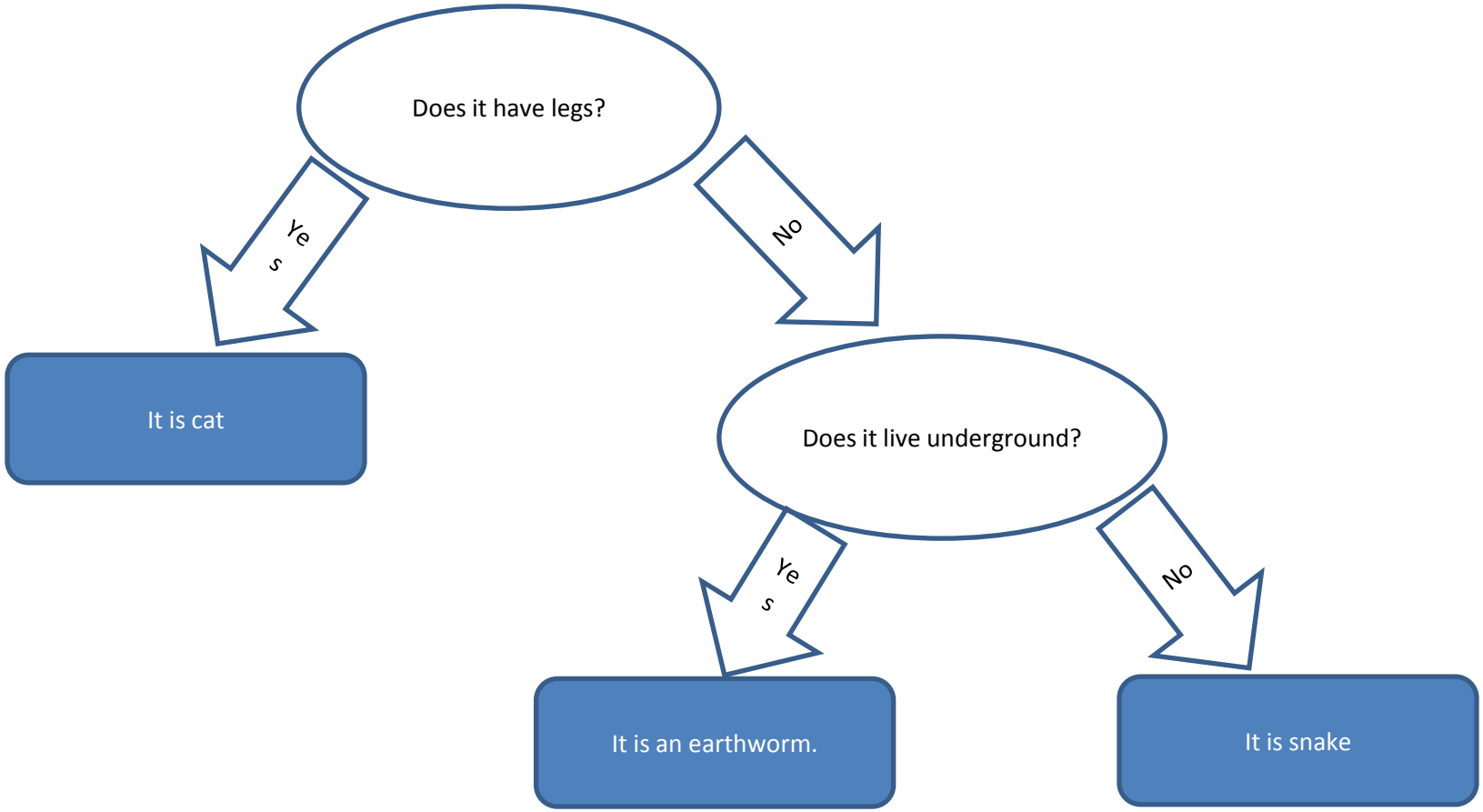
Machine Learning Problem

- Given example Dataset.
- Learn a system (i.e., Decision Tree) to predict unknown instance.

Example Data set (Animal)

Has Leg	Lives underground	Prediction
YES	NO	Cat
NO	YES	Earthworm
NO	NO	Snake

Decision tree

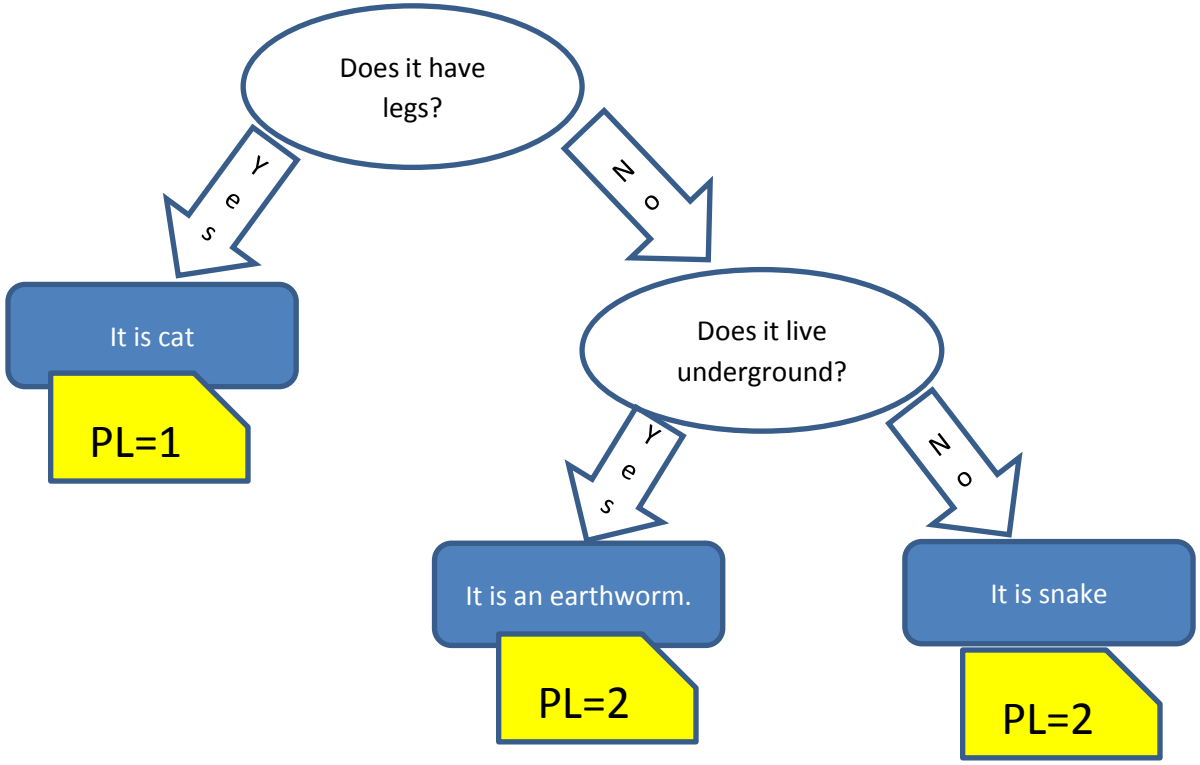


Big Question

- How can we construct the Decision Tree from dataset so that we can predict any instance (vector of attributes value) as quickly as possible?
- **Answer:** we need optimal decision tree (DT).

Cost of a Tree

- Sum of all length of path.
- PL=Path Length.



- Sum(PL_i)=5

Optimal Decision Tree (DT)?

- Generated DT from dataset that gives minimum cost.
- Selecting the best DT from all possible trees.

- **BUT** finding the minimum cost (best) tree is ***NP-Complete*** [1] problem. [Proved in May 1976]

ID3?

- ID3 is an algorithm invented by Ross Quinlan used to generate a decision tree from a dataset.
- It uses good Heuristic (Information gain) to construct DT with less cost.
- Iterative Dichotomiser 3
- Typically used in the **machine learning** and **natural language processing** domains.

Advantages of ID3

- It uses greedy approach. So it is really fast.
- It constructs optimal Decision Tree for more than 50% of all datasets.
- [*According to my experiments*].

Goals of my project

- *First goal*: To show that ID3 does not always construct an optimal DT.
- *Second goal*: How frequent does it produce optimal Decision Tree?

Experiment

- I wrote three programs
  - Calculate the cost of generated DT by ID3
  - Calculate the cost of optimal DT
    - Generating all possible trees and selecting best one.
  - Random Dataset generator for small number of attributes

Calculation of cost of DT by ID3

```
110  ii ID3length(vii db,int mask){
111      if(isAllSameClass(db)|| mask==-1+(1<<(C-1))){
112          return ii(0,1); //sum , no of leaf
113      }
114      //cout<<mask<<endl;
115      double mg=-1,t;
116      int at=0;
117      for(int i=0;i<C-1;i++)// for each attributes;
118      if(!(mask&(1<<i)))
119      {
120          t=IG(db,i);
121
122          if(t>mg){
123              mg=t; at=i;
124          }
125      }
126      //cout<<at<<" .";
127      ii a,b;
128      a=ID3length(project(db,at,0),mask|(1<<at)); a.first+=a.second;
129
130      b=ID3length(project(db,at,1),mask|(1<<at));b.first+=b.second;
131
132      ii r;
133      r.first=a.first+b.first;
134      r.second=a.second+b.second;
135
136      return r;
137
138  }
```

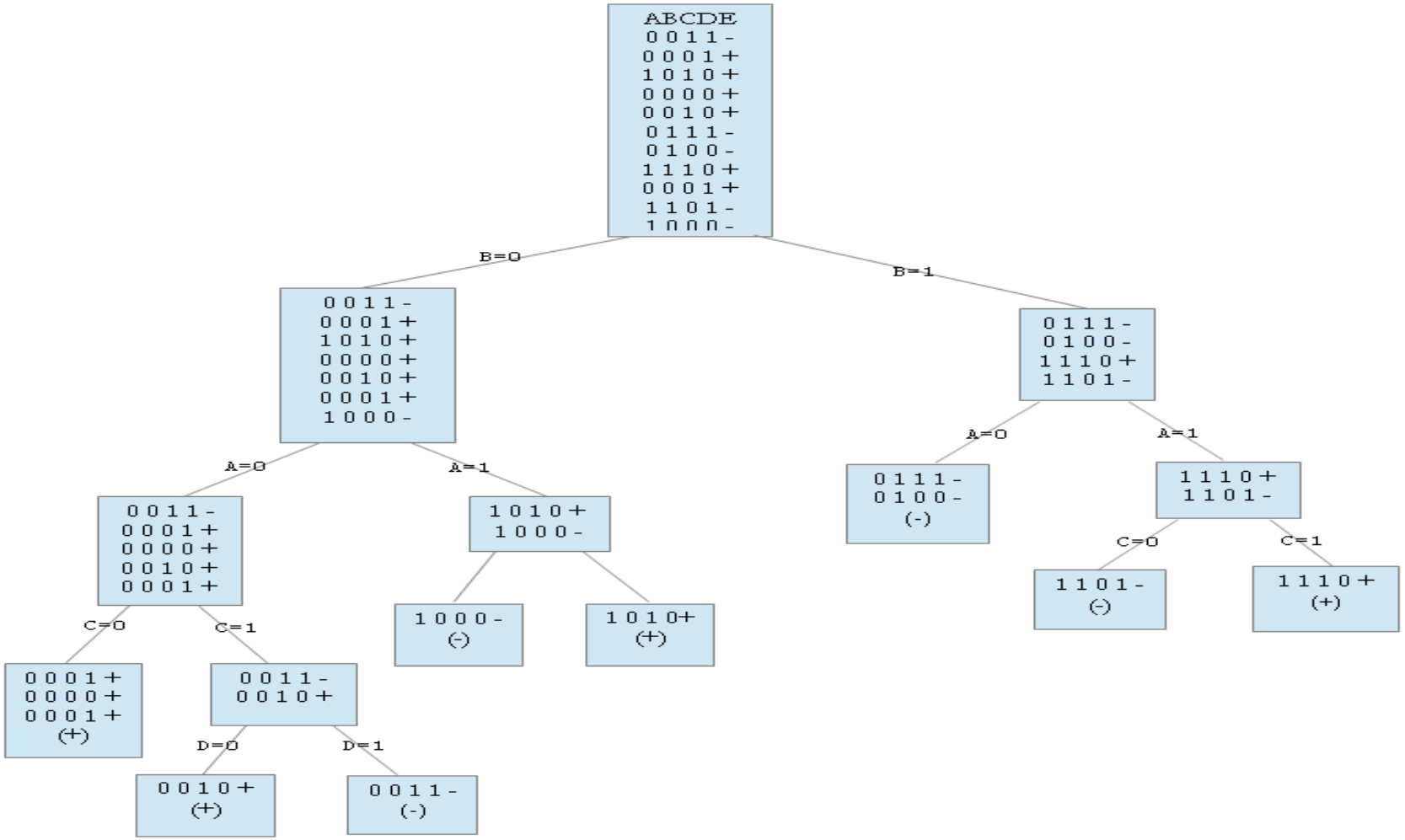
Calculation of cost of optimal DT

```
143  ii minCostTree(vii db,int mask,int lb){
144
145      if(isAllSameClass(db)|| mask==-1+(1<<(C-1))){
146
147          return ii(0,1); //sum of len , no of leaf
148      }
149      //showDB(db);
150      int na=db[0].size()-1;
151      int att=-1;
152      ii a,b,r,mn(INF,0);
153      for(int i=0;i<na;i++)
154      if(!(mask&(1<<i))){
155
156          a=minCostTree(project(db,i,0),mask|(1<<i),lb+1);     a.first+=a.second;
157
158          b=minCostTree(project(db,i,1),mask|(1<<i),lb+1);     b.first+=b.second;
159
160          r.first=a.first+b.first;
161          r.second=a.second+b.second;
162          if(mn.first>r.first){
163              att=i;
164          }
165          mn=min(r,mn);
166
167      }
168
169      return mn;
170
171
172  }
```

Randomly generated dataset

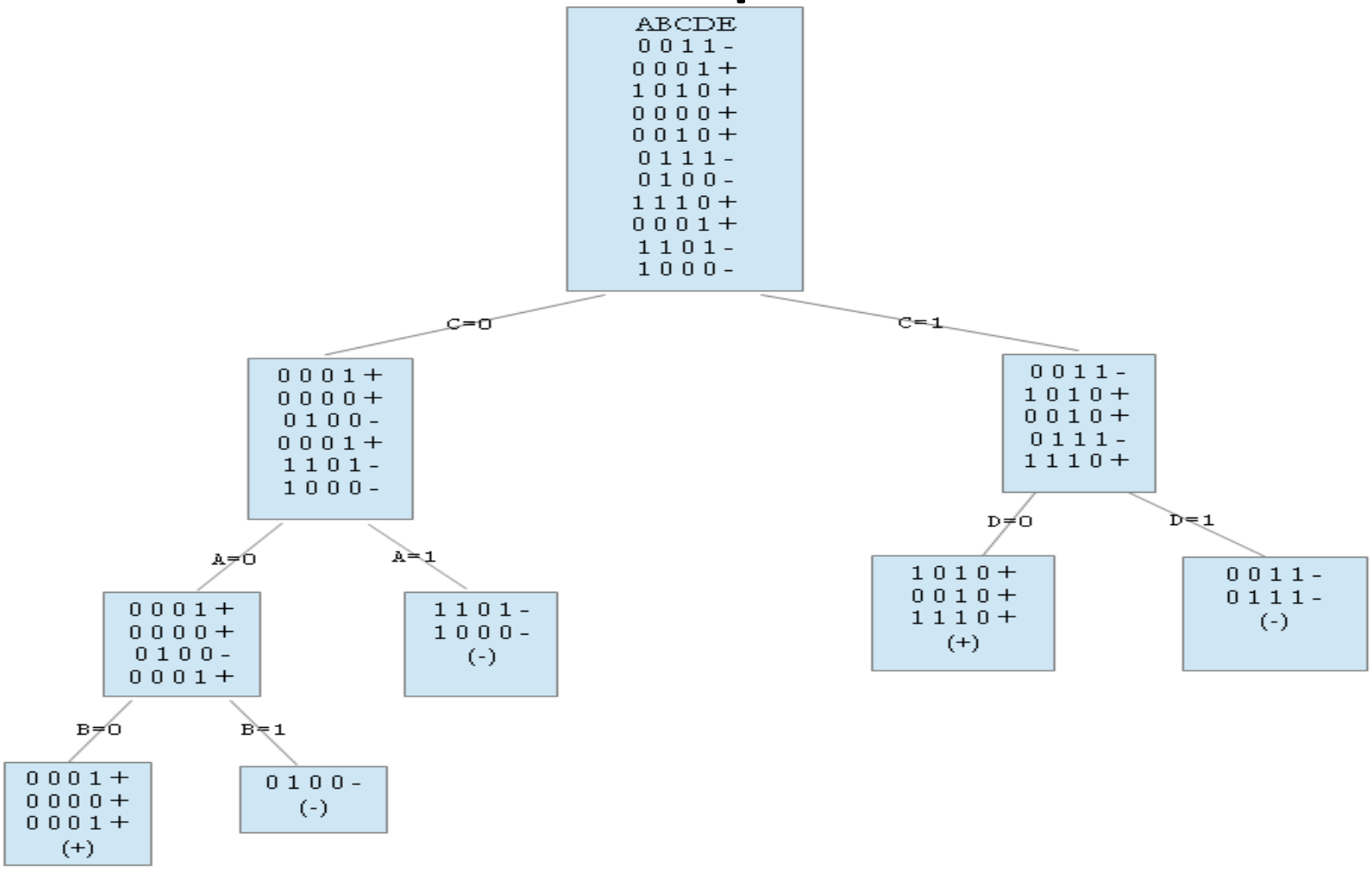
A	B	C	D	E
0	0	1	1	-
0	0	0	1	+
1	0	1	0	+
0	0	0	0	+
0	0	1	0	+
0	1	1	1	-
0	1	0	0	-
1	1	1	0	+
0	0	0	1	+
1	1	0	1	-
1	0	0	0	-

Generated DT by ID3



Cost =Total length of path= 25

Generated optimal DT



Cost =Total length of path= 12

- Now we know ID3 does not produce the optimal decision tree.
- **BUT** how frequent does it produce optimal decision tree?

Number of dataset	ID3 is optimal	ID3 is not optimal	Optimal in %
50	27	23	54%
50	32	18	64%
50	33	17	66%
100	61	39	61%
100	52	49	52%
150	83	67	55%

*Note: Number of attributes is less than 10

Conclusion

- ID3 does not produce optimal decision tree but gives a good approximation.
- For more than 50% of datasets, it produces optimal Decision tree.

References

- [1] Laurent Hyafil, Ronald L. Rivest. "**Constructing optimal binary decision trees is NP-complete**"

Q&A