```
%create sampling rec
%(+0,+ListofRandkey,+Attribute)

create_rec(_,[],_):-true.
create_rec(N,[H|T],A):- instance(H,R1,R2),
              include(filter(A),R2,R22), N1 is N+1,
              assert(instanceR(N1-H,R1,R22)),!,
              create_rec(N1,T,A).

%filter(+AttrList,+Element)
% true or false -- filter for selected attri
filter([],_):-false.
filter([H|_],(H=_)):-true,!.
filter([H|T],(M=V)):-M\==H,filter(T,(M=V)). %<<<<<<< HERE density

%TLL=[[outlook+sunny+3, outlook+overcast+1, outlook+rainy+3],...]
%tally(-TLL)

tally(TLL):- findall(A+VL,attributeR(A,VL),L),
              maplist(map,L,LL),tallyAtt(LL,TLL).

tallyAtt(LL,TLL):-maplist(tallyEach,LL,TLL).
tallyEach(L,TL):-maplist(finda,L,TL).

finda(A+V,(A+V+N)):-
       findall(A+V,(instanceR(_,class=C,L),(member(A=V,L);(A=class,V=C))),Res),
       length(Res,N).

map(A+VL,EL):-maplist(add(A),VL,EL).
add(A,B,A+B).

% called from MENU GUI.
%     +'outlook_out',+'[3,0.23]',+2,+all_cluster.pl

mainp(Dl,ParaDens1,K,Fout):- init,
       reconsult(Dl),Per=100, S=1,
       findall(X,(attribute(X,_),X\=class), AL),
       term_to_atom(ParaDL,ParaDens1),
       [M,D]=ParaDL, all_rec_dens(M,D),create_attr(AL),
       choose_sampling(Per,S,AL),
       tally(TLL),writeln(TLL),writeln(end+main),
       no_rec(Want,Actual),
       tell('ood.pl'),  % save densed instance --extra file
       format('~n%Density Parameter=[~a,~a]
                    Sampling[Percent,Type]=[~a,~a]~n',[M,D,Per,S]),
       format('~n%Want ~a records, but has ~a records~n',
                    [Want,Actual]),
                    (attributeR(X,Y),write(attribute(X,Y)),writeln('.'),
                    fail;true),
       (instanceR(N4-_,K4,L4),  write(instance(N4,K4,L4)),
         writeln('.'),  fail ; true),   !,told,  %   InFile='ood.pl',
       format('~ncaling ...cluster precess....~n'),
       tell(Fout),
       format('%--file:~w~n% output of clustering result',[Fout]),
       format('%density parameter=~w ,No of cluster= ~w',[ParaDens1,K]),
       findall(_,(attributeR(X,Y),
                    format('~nattribute(~w,~w).',[X,Y])),_),nl,
       mainClust(K,AllCluster),
       format('~n~nallCluster(~w).~ndescription(''~w'',~w).~n',
                    [AllCluster,Fout,K]),
       told.
```

```
%---------------------MENU---------------------
dens_clust_menu:-
  new(Dialog,dialog('Density Biased Clustering')),send_list(Dialog, append,
  [ new(D1, text_item(input_datafile,'outlook_out')),
    new(Dens, text_item(density_parameter,'[3,0.143]')),
    new(Per, int_item('number_cluster', low := 2, high := 80)),
    new(D2, text_item(output_file,'all_cluster1.pl')),
    button(cancel, message(Dialog, destroy)),
    button(enter, and(message(@prolog,mainp,
                              D1?selection,
                              Dens?selection,
                              Per?selection,
                              D2?selection
                             ),
          message(Dialog, destroy)))  % enter&destroy

  ]),
  send(Dialog, default_button, enter),
  send(Dialog, open).

%   working-- for discrete attributes
%+Kcluter,-AllClust

makeInitCluster(K,AllClust):-initClust(K,1,AllClust).

%+Kcluster,+L0startK,-AllClust
%4=Kcluster,1=start,[1*center,2*center]

initClust(K,L0,[]):-L0>K ,! .

initClust(K,L0,[L0*L|T]):- instanceR(L0-_,_,L),
                           L1 is L0+1,
                           initClust(K,L1,T).

%+recNum,+clustNo*[allAttri],-freqMatch)
% 3     , 2*[...]           , 2*15 )

freq(X,N*Y,N*F) :-  instanceR(X-_,_,L1),
                    intersection(L1,Y,I),
                    length(I,F).

cvalue(_*V,V).
cmax(L,A*V):-maplist(cvalue,L,L2),max_list(L2,V),member(A*V,L),!.

%+AllClust,+MaxInstanceN0,+StartAt,-AllFormatedPointList).
% AllClust,14           ,1    ,[1-15-2,2-11-1,...point-maxFreq-clustNo])

assignPoint(_,U,M,[]):-M>U,!.

assignPoint(AllClust,U,M,[M-V-A|T]):- maplist(freq(M),AllClust,Res),
                                      cmax(Res,A*V),
                                      M1 is M+1,
                                      assignPoint(AllClust,U,M1,T).

%+Kclusters,+StartAt,+AllFormatedPoint,-AllCluster
%2,         1,       AllPoint,[1*[outlook=sunny,temp=hot,..],2*[..]]  )

reComputeCenter(K,S,_Allpoint,[]):-S>K,!.
reComputeCenter(K,S,AllPoint,[S*NewCenter|T]):-
       findall(P,member(P-_-S,AllPoint),Z),%Z is AllP
       allPointAtAllAttr(Z,NewCenter),
       S1 is S+1,
       reComputeCenter(K,S1,AllPoint,T).
```

```
%+AllPoints,-AllClust
%[1,3,...],[1*[outlook=sunny,temp=hot,...],2*[...]]  )

allPointAtAllAttr(AllP,NewClusters) :-
       findall(AttName,(attribute(AttName,_),AttName\==class),AttNameL),
                                          % find all AttName
       maplist(allPoint(AllP),AttNameL,NewClusters).

%+AllPoints,+AttrName,-Attr=AttrVal
%[1,3,4],outlook,outlook=sunny

allPoint(AllP,Att,A):-
       findall(Att=V,(instanceR(X-_,_,K),member(X,AllP),member(Att=V,K)),Z),
       maxFreq(Z,A*V).

%+List,-MaxElement*Freq
%[a,a,b],a*2.

maxFreq(L,A*V):- findall(X*C,(member(X,L),count(X,L,C)),Z),cmax(Z,A*V).

write_cluster_member(K,AllPoint,NewClust):-
       numlist(1,K,KList),
       findall(Z-X,member(X-_-Z,AllPoint),R1),
       maplist(filter1,NewClust),
       maplist(filter11(R1),KList,_).

filter1(N*L):-format('~ncluster(~w,~w).',[N,L]).

filter11(L,X,Res):- findall(Y,member(X-Y,L),Res),
                    format('~ncluster_member(~w,~w).',[X,Res]).

%+Kcluster,+AllPoints,+AllClust
%2,       [1,3,4,5], OldClust

repeatCompute(K,AllPoint,OldClust,Ret) :-
       reComputeCenter(K,1,AllPoint,NewClust),
       (   OldClust==NewClust -> (
                           format('~nallPoint(~w).~n',[AllPoint]),
                           write_cluster_member(K,AllPoint,NewClust),
                           Ret=NewClust);% exit when Old=New
             (   %% write New Cluster
                 dens_rec(Rec),length(Rec,LastItemNo),
                 assignPoint(NewClust,LastItemNo,1,AllPoint2),
                          % assign points to new clusters
                        %writeln(allNewPoint-AllPoint2),
                 repeatCompute(K,AllPoint2,NewClust,Ret))).

%+Element,+List,-Count
%a,[a,b,a,a],3
count(_,[],0).
count(H,[X|T],C):-H\==X,!,count(H,T,C).
count(H,[H|T],C):-count(H,T,C1),!,C is 1+ C1.

%--------- main----
% +K cluster, -Ret ReturnedCluster.

mainClust(K,Ret):- findall(X,instanceR(X-_,_,_),InsL),
       length(InsL,Len1),
       ( Len1<K -> format('~n%_not_working Instance(~w)< K(~w)~n',[Len1,K]);
             (      makeInitCluster(K,AllClust),
                    assignPoint(AllClust,Len1,1,AllPoint),
                    OldClust=AllClust,
                    repeatCompute(K,AllPoint,OldClust,Ret)  ) ).
```

```
%------------ MERGE CLUSTER MENU ----
merge_clust_menu:-
       new(Dialog,dialog('Merge Clusters')),
       send_list(Dialog, append,
       [ new(D1, text_item(file1,'all_cluster1.pl')),
         new(D2, text_item(file2,'all_cluster2.pl')),
         new(D3, text_item(output_file,'all_cluster.pl')),
         new(K, int_item(numberOfCluster, low := 2, high := 20)),
         button(cancel, message(Dialog, destroy)),
         button(enter, and(message(@prolog,mergeClust,
                           D1?selection,
                           D2?selection,
                           D3?selection,
                           K?selection
                          ),
          message(Dialog, destroy)))  % enter&erase
  ]),
  send(Dialog, default_button, enter),
  send(Dialog, open).

%%     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%+inputfile,+inputfile

mergeClust(D1,D2,Fout,K):-
       format('~nmerging...from~w ~w~n',[D1,D2]),
       tell('tmp.pl'),
       format('%file tmp.pl~n:-dynamic attribute/2, instance/3.'),
       consult(D1),description(_,K1),
       findall(_,(attribute(X,Z),
                  format('~nattribute(~w,~w).',[X,Z])),_),
       findall(_,(cluster(X,Z),
                  format('~ninstance(~w,class=yes,~w).',[X,Z])),_),
       consult(D2),
       description(_,K2),nl,
       KLast is K1+K2,
       findall(_,(cluster(X,Z),
       N is X+K1,
       format('~ninstance(~w,class=no,~w).',[N,Z])),_),
       told,
       mainp('tmp.pl','[0,0]',K,'x.pl').


% ========= End of Data Clustering Program ===========
```

```
/* ============  Classification : main program ========= */
%
% To run the program, call this procedure:
%                                              id3menu.
%
%--------------  MENU ----------------

id3menu :-
       new(Dialog,dialog('Create Rules from Id3')),
       send_list(Dialog, append,
              [ new(D1, text_item(datafile,'post-operative.pl')),
                new(Per,text_item(minProb,'0.0')),
                button(cancel, message(Dialog, destroy)),
                button(enter, and(message(@prolog,callId3,
                           D1?selection,
                           Per?selection ),
                message(Dialog, destroy)))  % enter&destroy
               ]),
       send(Dialog, open).
%
callId3(Dfile,Per):-
       term_to_atom(Per1,Per),
       consult(Dfile),
       mainId3(Per1).
%
%-------------- ID3 -------------------

:- dynamic current_node/1,node/2,edge/3,hasClass/2.

mainId3(Min):-
       init(AllAttr,EdgeList),
       getnode(N),
       create_edge_onelevel(N,AllAttr,EdgeList),
       addKnowledge,
       selectRule(Min,Res),
       maplist(writeln,Res).

init(AllAttr,[root-nil/PB-NB]):-
       writeln(limitUpto400nodes),
       retractall(hasClass(_,_)),
       attribute( class,[ Y1, Y2]),
       assert(hasClass(Y1,Y2)),
       retractall(node(_,_)),
       retractall(current_node(_)),
       retractall(edge(_,_,_)),
       assert(current_node(0)) ,
       hasClass(C1,C2),
      findall(X,attribute(X,_),AllAttr1),
       delete(AllAttr1,class,AllAttr),
       findall(X2,instance(X2,class=C1,_),PB),
       findall(X3,instance(X3,class=C2,_),NB),
       length(PB,N1),
       length(NB,N2),
       N is N1+N2,
       retractall(total+_),
       apply(assert,[total+N]).

getnode(X):-
       current_node(X),
       X1 is X+1,
       retractall(current_node(_)),
       assert(current_node(X1)),
       X1 <400. % limit at 400 nodes
```

```
create_edge_onelevel(_,_,[]):-!.
create_edge_onelevel(_,[],_):-!.

create_edge_onelevel(N,AllAttr,EdgeList):-
       create_nodes(N,AllAttr,EdgeList).

create_nodes(N,AllAttr,[H1-H2/PB-NB|T]):-
              getnode(N1),
              assert(edge(N,H1=H2,N1)),
              assert(node(N1,PB-NB)),
              append(PB,NB,AllInst),
              ( (PB\==[], NB\==[])->
                    (cand_node(AllAttr,AllInst,AllSplit),
                     min_cand(AllSplite,[V,MinAttr,Split]),
                     delete(AllAttr,MinAttr,Attr2),
                     create_edge_onelevel( N1,Attr2,Split))
                     ;   true ),
              create_nodes(N,AllAttr,T).

create_nodes(_,_,[]):-!.
create_nodes(_,[],_):-!.

%
% --------------------------------
check1 :- node(X,Y),write(node/X+Y),fail ; true.
check2 :- edge(X,Y,Z),node(Z,N),writeln(edge/X+Y+Z*N),fail ; true.
check3 :- edge(X,Y,Z),node(X,L),node(Z,N),
              writeln(edge/X*L+Y+Z*N),fail ; true.
check4 :- X>>Class>>N,writeln(X>>Class>>N),fail ; true.

addKnowledge :-
       findall([A],pathFromRootToLeaf(A,_),Res),
       retractall(_>>_>>_),
       maplist(apply(assert),Res),
       write(addToKNB).

selectRule(V,Res):-
       findall(N>>X>>Class,(X>>Class>>N,N>=V),Res1),
       sort(Res1,Res2),
       reverse(Res2,Res).

path(A,[H|T],C):-
       edge(A,H,B),
       path(B,T,C).

path(C,[],C):-!.
pathFromRootToLeaf(V>>Class>>Num,C):-
       path(1,V,C),
       node(C,Value1-Value2),
       (Value1=[];Value2=[]),
       (Value1=[]->length(Value2,Numb);length(Value1,Numb)),
       total+Total,
       Num is Numb/Total,hasClass(C1,C2),
       (Value1=[]->Class=C1;Class=C2).

min_cand([H|T], Min) :-    min_cand(T, H, Min).

min_cand([], Min, Min).
min_cand([H|T], Min0, Min) :-     H=[V,_,_],Min0=[V0,_,_],
       ( V<V0 ->Min1=H;Min1=Min0),
       % Min1 is min(H, Min0),
       min_cand(T, Min1, Min).
```

```
cand_node([H|T],CurInstL,[[Val,H,SpliteL]|OtherAttr]):-
       info(H,CurInstL,Val,SpliteL),
       cand_node(T,CurInstL,OtherAttr).

cand_node([],_,[]):-!.
cand_node(_,[],[]).

concat3(A,B,C,R):- atom_concat(A,B,R1), atom_concat(R1,C,R).

info(A,CurInstL,R,Splite):- attribute(A,L),
       maplist(concat3(A,=),L,L1),%make a good form
       suminfo(L1,CurInstL,R,Splite). %L1=[size-small,size-large]

%suminfo(+[color=red,color=blue],+[1,2,3,4],-info,-SplitList).
%
suminfo([H|T],CurInstL,R,[Split|ST]) :-
       AllBag=CurInstL,
       hasClass(C1,C2),
       term_to_atom(H1,H),
       findall(X1,(instance(X1,_,L1),member(X1,CurInstL),
                     member(H1,L1)),BagGro),
       findall(X2,(instance(X2,class=C1,L2),member(X2,CurInstL),
                     member(H1,L2)),BagPos),
      findall(X3,(instance(X3,class=C2,L3),member(X3,CurInstL),
                     member(H1,L3)),BagNeg),
       (H11=H22) =H1,
       length(AllBag,Nall),length(BagGro,NGro),length(BagPos,NPos),
       length(BagNeg,NNeg),
       Split=H11-H22/BagPos-BagNeg,
       suminfo(T,CurInstL,R1,ST),
       (  NPos is 0 *->L1 = 0; L1 is (log(NPos/NGro)/log(2)) ),
       (  0 is NNeg *->L2 = 0; L2 is (log(NNeg/NGro)/log(2)) ),
       (  NGro is 0 -> R= 999;
            R is (NGro/Nall)*(-(NPos/NGro)*L1-(NNeg/NGro)*L2)+R1 ).
suminfo([],_,0,[]).
```

```
/* ============  Post-Data Mining : main program ========= */

/*-------- How to run  ------
1. call  id3menu.pl  to create decision-tree model
2. then call expertshell1.pl
3. interactive with the expert system shell
   with the following commands:
      > expertshell.
      > load.       and input '1.knb'.
      > solve.
      > why.
      > quit.
---------------------------*/

:-dynamic known/1, answer/2.

expertshell :-
            greeting,
            repeat,
            write('expert-shell> '),
            read(X),
            do(X),
            (X == quit;X == 99),
            writeln('>>>>Goodbye, see you later<<<<'), !.

greeting :-
            write('This is the Easy Expert System shell.'), nl,
            native_help.

do(help) :- native_help, !.
do(load) :- load_kb, !.
do(solve) :- solve, !.
do(why):- why,!.
do(quit). do(99).

do(X) :-    write(X),
            write(' is not a legal command.'), nl, fail.

native_help :-
            write('Type  help.   load.   solve.   why.    quit. or  99.'),nl,
            write('at the prompt.'), nl.

load_kb :-  write('Enter file name in single quotes (ex. ''1.knb''.): '),
            read(F),
            reconsult(F).

solve :-    retractall(known( _) ),retractall(answer(_,_)),
            top_goal(X,V),
            format('The answer is __~w__ with probability ~w',[X,V]),
            assert(answer(X,V)),nl.

solve :-    write('No answer found.'),nl.

menuask(Pred,Value,Menu):-
            menuask(Pred,Menu),
            atomic_list_concat([Pred,'(',Value,')'],X),
            term_to_atom(T,X),known(T),!.

menuask(Pred,_):-
      atomic_list_concat([Pred,'(','_',')'],X),  % check for recorded predicate
      term_to_atom(T,X),known(T),!.              % not ask again
```

```
menuask(Attribute,Menu):-
            nl,write('What is the value for '),
            write(Attribute),write('?'),nl,
            addchoice(Menu,MenuRes),writeln(MenuRes),
            write('Enter the  choice> '),
            read(C),
            member(C-V,MenuRes),
            (C=99 -> abort ; true),
            atomic_list_concat([Attribute,'(',V,')'],X),
            term_to_atom(T,X),
            asserta(known(T)).

why:- answer(A,V),
      format('~nThe answer is ...~w... with probability = ~w.~n',[A,V]),
      findall( X , known(X),Result),
      writeln('The known storage are'),
      writeln(Result).

addchoice(X,Res):-
       length(X,Len),
       numlist(1,Len,NumL),
       map(NumL,X,Res).

map([],[],[99-exitShell]).
map([H|T],[X|TT],[H-X|T1]):- map(T,TT,T1).

% ========= End of Expert System Shell Program ===========


%================================================
%  Rule induction program
%    ID3-based algorithm with probabilistic values
%================================================

addAllKnowledge:-
       findall([A],pathFromRootToLeaf(A,_),Res)       ,
       retractall(_>>_>>_),
       maplist(apply(assert),Res),
       write(addToKNB),nl.  % add to knowledge base

selectRule(V,Res):-
       findall(N>>X>>Class,(X>>Class>>N,N>=V),Res1),
       sort(Res1,Res2),
       reverse(Res2,Res).

path(A,[H|T],C):-
       edge(A,H,B),
       path(B,T,C).
path(C,[],C):-!.

pathFromRootToLeaf(V>>Class>>Num,C):-
       path(1,V,C),
       node(C,Value1-Value2),
       (Value1=[] ; Value2=[]),
       (Value1=[]->length(Value2,Numb) ; length(Value1,Numb)),
       total+Total,
       Num is Numb/Total,
       hasClass(C1,C2),
       (Value1=[]-> Class=C2 ; Class=C1).

min_cand([H|T], Min) :-
       min_cand(T, H, Min).
min_cand([], Min, Min).
```

```
min_cand([H|T], Min0, Min) :-
        H=[V,_,_],Min0=[V0,_,_],
        ( V<V0 ->Min1=H;Min1=Min0),
        min_cand(T, Min1, Min).

cand_node([H|T],CurInstL,[[Val,H,SpliteL]|OtherAttr]):-
        info(H,CurInstL,Val,SpliteL),
        cand_node(T,CurInstL,OtherAttr).

cand_node([],_,[]):-!.
cand_node(_,[],[]).

concat3(A,B,C,R):-
        atom_concat(A,B,R1),
        atom_concat(R1,C,R).

info(A,CurInstL,R,Splite):-
        attribute(A,L),
        maplist(concat3(A,=),L,L1),%make a good form
        suminfo(L1,CurInstL,R,Splite).

suminfo([H|T],CurInstL,R,[Splite|ST]) :-
        AllBag=CurInstL,hasClass(C1,C2),
        term_to_atom(H1,H),
        findall(X1,(instance(X1,_,L1),member(X1,CurInstL),
                    member(H1,L1)),BagGro),
        findall(X2,(instance(X2,class=C1,L2),member(X2,CurInstL),
                    member(H1,L2)),BagPos),
        findall(X3,(instance(X3,class=C2,L3),member(X3,CurInstL),
                    member(H1,L3)),BagNeg),
        (H11=H22) =H1,
        length(AllBag,Nall),
        length(BagGro,NGro),
        length(BagPos,NPos),
        length(BagNeg,NNeg),
        Splite=H11-H22/BagPos-BagNeg,
        suminfo(T,CurInstL,R1,ST),
        (   NPos is 0 *->L1 = 0; L1 is (log(NPos/NGro)/log(2)) ),
        (   0 is NNeg *->L2 = 0; L2 is (log(NNeg/NGro)/log(2)) ),
        (   NGro is 0 -> R= 999;
              R is (NGro/Nall)*(-(NPos/NGro)*L1-(NNeg/NGro)*L2)+R1 ) .
suminfo([],_,0,[]).

%-------------- ID3 ------------------

:- dynamic current_node/1,node/2,edge/3,hasClass/2,type/2.

init(AllAttr,[root-nil/PB-NB]):-
        retractall(hasClass(_,_)),
        attribute( class,[ Y1, Y2]),
        assert(hasClass(Y1,Y2)),
        retractall(node(_,_)),
        retractall(current_node(_)),
        retractall(type(_,_)),
        retractall(edge(_,_,_)),
        assert(current_node(0)) ,
        hasClass(C1,C2),
        findall(X,attribute(X,_),AllAttr1),delete(AllAttr1,class,AllAttr),
        findall(X2,instance(X2,class=C1,_),PB),
        findall(X3,instance(X3,class=C2,_),NB),
        length(PB,N1),length(NB,N2),N is N1+N2,
        retractall(total+_),
        apply(assert,[total+N]).
```

```
getnode(X):-
       current_node(X),
       X1 is X+1,
       retractall(current_node(_)),
       assert(current_node(X1)),
       X1 <4000.       % limit tree size at 4000 nodes

create_edge_onelevel(_,_,[]):-!.
create_edge_onelevel(_,[],_):-!.
create_edge_onelevel(N,AllAttr,EdgeList):- create_nodes(N,AllAttr,EdgeList).

create_nodes(N,AllAttr,[H1-H2/PB-NB|T]):-
       getnode(N1),
       assert(edge(N,H1=H2,N1)),
       assert(node(N1,PB-NB)),
       append(PB,NB,AllInst),
       ( (PB\==[], NB\==[])->
             (cand_node(AllAttr,AllInst,AllSplite),
              min_cand(AllSplite,[V,MinAttr,Splite]),
              delete(AllAttr,MinAttr,Attr2),
              create_edge_onelevel( N1,Attr2,Splite)) ; true ),
       create_nodes(N,AllAttr,T).

create_nodes(_,_,[]):-!.
create_nodes(_,[],_):-!.

mainId3(Min):-
       init(AllAttr,EdgeList),
       getnode(N),
       create_edge_onelevel(N,AllAttr,EdgeList),
       addAllKnowledge,
       selectRule(Min,Res),
       writeln(Res),
       tell('1.knb'),
       writeHeadF,
       maplist(createRule1,Res),
       nl,writeTailF,
       told, writeln(endProcess).

%------------------------
%  Generate rules in KB
%------------------------
writeHeadF :-
  format('% 1.knb ~n% for expert shell. --- written by Postprocess'),
  format('~n% top_goal where the inference starts.~n'),
  format('~ntop_goal(X,V) :- type(X,V).~n').

writeTailF:-
  findall(_,(attribute(S,L),
  format('~n~w(X):-menuask(~w,X,~w). %generated menu',[S,S,L])),_),
  format('~n~n%end of automatic post process').

%--------------  MENU ----------------
id3menu:-
  new(Dialog,dialog('Create Rules from Id3')),
  send_list(Dialog, append,
  [ new(D1, text_item(datafile,'post-operative.pl')),
    new(Per,text_item(minProb,'0.016')),
    button(cancel, message(Dialog, destroy)),
    button(enter, and(message(@prolog,callId3,D1?selection,Per?selection ),
    message(Dialog, destroy)))  % enter&destroy
  ]),
  send(Dialog, open).
```

```
%
callId3(Dfile,Per):-
       term_to_atom(Per1,Per),
       consult(Dfile),
       mainId3(Per1).

:-id3menu.

%---------------

transform1([X=V],[Res]):-
       atomic_list_concat([X,'(',V,')'],Res1),
       term_to_atom(Res,Res1),!.

transform1([X=V|T],[Res|T1]):-
       atomic_list_concat([X,'(',V,')'],Res1),
       term_to_atom(Res,Res1),
       transform1(T,T1).

createRule1(I):-
       I=Z>>X>>Y,
       transform1(X,BodyL),
       format('~ntype(~w,~w):-',[Y,Z]),
       myformat(BodyL) ,
       write('  % generated rule'),!.

myformat([X]):-write(X),write('.'),!.
myformat([H|T]):-write(H),write(','),myformat(T).


% ==================== End of Rule Induction Program =======
```

```
% ------------------------------------
%      Data File:  Post-operative.pl
% ------------------------------------
%   class home = after operation patient prepared to go home
%   class ward = patient not in good condition: sent to general floor

attribute( internalTemp,       [mid, high, low] ).
attribute( surfaceTemp,        [mid,high, low] ).
attribute( oxygenSaturation,   [excellent, good, fair, poor] ).
attribute( bloodPressure,      [high, mid, low] ).
attribute( tempStability,      [stable, mod_stable, unstable] ).
attribute( coreTempStability,  [stable,mod_stable, unstable] ).
attribute( bpStability,        [stable, mod_stable, unstable] ).
attribute( comfort,            [5, 7, 10, 15] ).
attribute( class,              [ home, ward]).

instance(1, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=15] ).
instance(2, class=home, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(3, class=ward, [internalTemp=high, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(4, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=stable, coreTempStability=unstable,
                         bpStability=mod_stable, comfort=15] ).
instance(5, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(6, class=home, [internalTemp=high, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=unstable, comfort=15] ).
instance(7, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=5] ).
instance(8, class=home, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=unstable,
                         coreTempStability=unstable, bpStability=stable, comfort=10]).
instance(9, class=home, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
instance(10, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                         bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(11, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                         bpStability=stable, comfort=15]).
instance(12, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=stable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(13, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=unstable, comfort=15] ).
instance(14, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=10] ).
instance(15, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable,
                         coreTempStability=unstable, bpStability=stable, comfort=15]).
instance(16, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=unstable, comfort=10] ).
instance(17, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=mod_stable, comfort=15]).
instance(18, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                         bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                         bpStability=stable, comfort=10] ).
```

```
instance(19, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=15] ).
instance(20, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(21, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15] ).
instance(22, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10] ).
instance(23, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=15] ).
instance(24, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(25, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(26, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10] ).
instance(27, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=low, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(28, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10] ).
instance(29, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=15]).
instance(30, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(31, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(32, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(33, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(34, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(35, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(36, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(37, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(38, class=home, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(39, class=home, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=7]).
instance(40, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(41, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
instance(42, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(43, class=home, [internalTemp=high, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(44, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(45, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(46, class=home, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(47, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(48, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=low, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(49, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=mod_stable, bpStability=mod_stable, comfort=10]).
instance(50, class=ward, [internalTemp=low, surfaceTemp=high, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(51, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(52, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(53, class=ward, [internalTemp=high, surfaceTemp=high, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(54, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(55, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=15]).
instance(56, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=high, tempStability=unstable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(57, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(58, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(59, class=ward, [internalTemp=mid, surfaceTemp=high, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(60, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(61, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(62, class=home, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(63, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=unstable, bpStability=unstable, comfort=10]).
instance(64, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
```

```
instance(65, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(66, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(67, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(68, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=low, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(69, class=ward, [internalTemp=low, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(70, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
%
/* *** Test Data
%     =========
instance(71, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(72, class=ward, [internalTemp=mid, surfaceTemp=low, oxygenSaturation=excellent,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(73, class=ward, [internalTemp=low, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(74, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=mod_stable, comfort=10]).
instance(75, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(76, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable,
                          coreTempStability=unstable, bpStability=stable, comfort=10]).
instance(77, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=high, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(78, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(79, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(80, class=ward, [internalTemp=high, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=unstable, comfort=5]).
instance(81, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=stable, coreTempStability=stable,
                          bpStability=unstable, comfort=10]).
instance(82, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(83, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(84, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
instance(85, class=ward, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=excellent,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=10]).
instance(86, class=home, [internalTemp=mid, surfaceTemp=mid, oxygenSaturation=good,
                          bloodPressure=mid, tempStability=unstable, coreTempStability=stable,
                          bpStability=stable, comfort=15]).
*/
```

ภาคผนวก ค

ผลงานวิจัยของชุดโครงการ SUT Miner ที่ได้รับการตีพิมพ์เผยแพร่

- K. Kerdprasop and N. Kerdprasop (2010). A technique of density biased sampling for clustering data of different sizes and densities. *Proceedings of 4th Ubon Ratchathani University Research Conference (UBRC 4th)*, Ubon Ratchathani University, Thailand, 9-10 September 2010. (To appear)
- K. Kerdprasop and N. Kerdprasop (2010). Post-data mining processing. *Proceedings of the 8th PSU-Engineering Conference (PEC8)*, Prince of Songkla University, Songkla, Thailand, 22-23 April 2010. (Best application paper award)
- N. Kerdprasop and K. Kerdprasop (2010). Probabilistic knowledge discovery from medical databases. *Proceedings of 14th International Annual Symposium on Computational Science and Engineering (ANSCSE 14),* Mae Fah Luang University, Chiang Rai, Thailand, March 23-26, pp. 543-548.
- K. Kerdprasop and N. Kerdprasop (2009). Knowledge mining with a higher-order logic approach. In K. Nakamatsu, G. Phillips-Wrens, L.C. Jain, R.J. Howlett (Eds.), *New Advances in Intelligent Decision Technologies*, pp. 151-159, Springer. (ISBN: 978-3-642-00908-2, DOI: 10.1007/978-3-642-00909-9).
- N. Kerdprasop and K. Kerdprasop (2009). Knowledge induction from medical databases with higher-order programming. *WSEAS Transactions on Information Science and Applications*, Issue 10, Volume 6, October 2009, pp. 1719-1728.
- K. Kerdprasop and N. Kerdprasop (2009). SUT-Miner: A knowledge mining and managing system for medical databases. *Proceedings of 20th International Workshop on Database and Expert Systems Applications (DEXA)*, Linz, Austria, August 31 – September 4, pp.318-322.
- N. Kerdprasop and K. Kerdprasop (2008). A declarative programming paradigm and the development of knowledge mining agents. *Proceedings of IADIS Multi Conference on Computer Science and Information Systems*, Amsterdam, Netherlands, July 22-27, pp. 45-52.
- K. Kerdprasop and N. Kerdprasop (2007). Functional programming paradigm and the development of SUT Miner system. *Proceedings of 33rd Congress on Science and Technology of Thailand*, Walailak University, Nakhon Srithammarat, Thailand, October 18-20.

เทคนิคการสุ่มข้อมูลตามความหนาแน่นเพื่อจัดกลุ่มข้อมูลที่มีขนาดและความหนาแน่นแตกต่างกัน

กิตติศักดิ์ เกิดประสพ และ นิตยา เกิดประสพ
หน่วยวิจัยด้านวิศวกรรมข้อมูลและการค้นหาความรู้ สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
111 ถนนมหาวิทยาลัย ต.สุรนารี อ.เมือง จ.นครราชสีมา 30000
kerdpras@sut.ac.th

A Technique of Density Biased Sampling for Clustering Data of Different Sizes and Densities

Kittisak Kerdprasop and Nittaya Kerdprasop
Data Engineering and Knowledge Discovery Research Unit, School of Computer Engineering, Suranaree University of Technology
111 University Avenue, Muang District, Nakhon Ratchasima 30000
kerdpras@suta.c.th

บทคัดย่อ

การจัดกลุ่มข้อมูลเป็นงานรวมกลุ่มข้อมูลด้วยการพิจารณาระยะใกล้เคียงหรือความคล้ายคลึงกัน อัลกอริทึมเคมีนส์เป็นอัลกอริทึมมาตรฐานที่ทำการรวมกลุ่มข้อมูล โดยเริ่มต้นกระบวนการด้วยการคำนวณระยะห่างเพื่อกำหนดข้อมูลแต่ละตัวเข้ากลุ่มที่อยู่ใกล้ที่สุด จากนั้นคำนวณค่าจุดกึ่งกลางของแต่ละกลุ่มโดยใช้ค่าเฉลี่ยของข้อมูลทั้งหมดในกลุ่ม อัลกอริทึมจะทำสองขั้นตอนนี้ซ้ำจนกระทั่งค่าจุดกึ่งกลางไม่มีการเปลี่ยนแปลงและข้อมูลไม่มีการเปลี่ยนกลุ่ม ประสิทธิภาพของอัลกอริทึมจะขึ้นกับจำนวนข้อมูล จำนวนกลุ่ม และจำนวนรอบในการวนซ้ำ ผู้วิจัยได้พัฒนาวิธีการสุ่มตามความหนาแน่นเพื่อลดจำนวนข้อมูลที่จะใช้ในการจัดกลุ่ม ซึ่งจะช่วยให้อัลกอริทึมจัดกลุ่มข้อมูลทำงานได้เร็วขึ้น การสุ่มข้อมูลจะใช้วิธีการคัดเลือกข้อมูลตัวแทนจากบริเวณที่มีความหนาแน่นสูง อัลกอริทึมที่พัฒนาขึ้นนี้ได้รับการแปลงเป็นโปรแกรมที่เขียนด้วยภาษาเออแลง ซึ่งเป็นภาษาเชิงฟังก์ชันที่ช่วยให้การพัฒนาโปรแกรมต้นแบบทำได้อย่างรวดเร็ว และจากผลการทดสอบโปรแกรมพบว่าโปรแกรมสุ่มข้อมูลและจัดกลุ่มข้อมูลตามความหนาแน่นสามารถทำงานได้ดีกับข้อมูลที่มีการกระจายแบบซิฟ ผลการจัดกลุ่มด้วยข้อมูลสุ่มพบว่าจุดกึ่งกลางกลุ่มเบี่ยงเบนไปจากจุดกึ่งกลางที่แท้จริงเพียงเล็กน้อย เมื่อวัดประสิทธิภาพการใช้เนื้อที่หน่วยความจำของโปรแกรมที่พัฒนาขึ้นพบว่าใช้เนื้อที่หน่วยความจำน้อยกว่าโปรแกรมจัดกลุ่มข้อมูลตามปกติถึง60% ดังนั้นเทคนิคและโปรแกรมที่พัฒนาขึ้นจึงมีศักยภาพที่จะพัฒนาให้ทำงานกับข้อมูลสตรีมได้ นอกจากนี้ภาษาเออแลงยังมีขีดความสามารถในการโปรแกรมแบบพร้อมกันกับหลายหน่วยประมวลผลได้ ซึ่งจะช่วยให้โปรแกรมทำงานได้เร็วขึ้นและรองรับข้อมูลปริมาณมากขึ้นได้

คำสำคัญ : การสุ่มข้อมูลตามความหนาแน่น ภาษาเออแลง โปรแกรมจัดกลุ่มข้อมูลอัตโนมัติ

Abstract

Clustering is a task of grouping data based on closeness or similarity. A standard k-means algorithm groups data by firstly assigning all data points to the closest clusters, then determining the new cluster mean of each cluster based on the average value of its members. The algorithm repeats these two steps until it converges; that is until there is no change in cluster means and cluster assignment among data points. Performance of the algorithm depends on the number of data points, number of data clusters, and number of iterations. To speed up the clustering process, we develop the density-biased reservoir sampling algorithm as an efficient data reduction technique. Instead of simply randomly selecting data for clustering, we evaluate data density and draw samples from the dense area. The proposed algorithm has been implemented with a functional programming paradigm using the Erlang language.

The declarative style of Erlang facilitates a rapid prototyping. Our experimental results reveal the effectiveness of drawing samples of high density from the Zipf distributed data. The shift of cluster means is minimal, whereas the decrease in memory usage is significant. The proposed density-biased reservoir sampling technique thus shows a great potential on dealing with large and streaming data. Moreover, the Erlang language itself has an important feature of concurrent programming on a multi-core architecture that can help speed up the computation of large and continuously generated data.

Keywords: Density biased sampling, ERLANG language, Automatic data clustering program

บทนำ

การจัดกลุ่มข้อมูล (data clustering) เป็นการวิเคราะห์ข้อมูลเพื่อรวมข้อมูลที่มีลักษณะหรือคุณสมบัติที่คล้ายกันให้อยู่กลุ่มเดียวกัน เพื่อประโยชน์ในการสรุปข้อมูลและพิจารณาลักษณะของแต่ละกลุ่มข้อมูล การจัดกลุ่มข้อมูลเป็นงานที่มีความสำคัญและนำไปใช้ประโยชน์ในหลายด้าน เช่น การจัดเอกสารอิเล็กทรอนิกส์ที่เนื้อหาเกี่ยวข้องกันให้อยู่ในหมวดเดียวกัน การจัดสารประกอบโปรตีนที่มีโครงสร้างโมเลกุลคล้ายกันให้อยู่ประเภทเดียวกัน การจัดกลุ่มลูกค้าที่มีพฤติกรรมการซื้อสินค้าคล้ายคลึงกัน เป็นต้น

เทคนิคการจัดกลุ่มข้อมูล เป็นการรวมกลุ่มข้อมูลโดยพิจารณาจากค่าที่ระบุลักษณะของข้อมูล (attribute values) กระบวนการทำงานของการจัดกลุ่มข้อมูลจึงเป็นการค้นหากลุ่มต่างๆ ของข้อมูลโดยข้อมูลภายในกลุ่มเดียวกันจะต้องมีลักษณะที่เหมือนกันมากที่สุด ในขณะที่ข้อมูลที่อยู่ต่างกลุ่มกันจะต้องมีลักษณะที่ต่างกันมากที่สุด

การวัดความเหมือนหรือความต่างนี้จะใช้เกณฑ์ที่แตกต่างกันไปในแต่ละอัลกอริทึม เช่น ใช้วัด distance ในอัลกอริทึม k-means (Jain and Dubes, 1988) หรืออาจจะใช้การวัดค่า cosine และค่าสัมประสิทธิ์ Jaccard (Stehl et al, 2000) การเลือกเกณฑ์วัดค่าความเหมือน (similarity) จะมีผลต่อรูปร่างของกลุ่มข้อมูลที่ค้นพบ เช่น ถ้าใช้เกณฑ์ระยะห่างแบบ Euclidean จะได้กลุ่มข้อมูลที่มีรูปร่างเป็นทรงกลม ในระยะเริ่มแรกของการพัฒนาเทคนิคการจัดกลุ่มข้อมูล กลุ่มข้อมูลที่ค้นพบมักจะมีลักษณะทรงกลม แต่ในระยะหลังได้มีการพัฒนาอัลกอริทึมที่สามารถค้นหากลุ่มข้อมูลที่มีรูปทรงแตกต่างไปจากทรงกลม เช่น อัลกอริทึม DBSCAN (Sander et al, 1998), CURE (Guha et al, 1998), Chameleon (Karypis et al, 1999)

ถึงแม้เทคนิคการค้นหากลุ่มข้อมูลจะพัฒนาขึ้นตามลำดับ แต่ปัญหาหลักของการจัดกลุ่มข้อมูลยังคงเป็นเรื่องของประสิทธิภาพการประมวลผล ทั้งนี้เนื่องจากการจัดกลุ่มข้อมูลเป็นงานเรียนรู้แบบไม่มีการชี้แนะ (unsupervised learning) ขั้นตอนการค้นหาคำตอบจึงใช้เวลามาก เทคนิคการลดเวลาที่ได้ผลดีที่สุด คือการลดขนาดข้อมูล โดยการเลือกใช้เฉพาะข้อมูลตัวแทน และวิธีการสุ่มเลือกข้อมูลตัวแทนที่ให้ผลดี คือ วิธี biased sampling

การสุ่มแบบ biased (Kollios et al, 2003) เป็นการสุ่มโดยกำหนดค่าความน่าจะเป็นที่จะถูกเลือกให้กับข้อมูลแต่ละตัวไม่เท่ากันขึ้นอยู่กับความหนาแน่นในกลุ่มข้อมูล รูปร่างของกลุ่มข้อมูล และรูปแบบการกระจายของข้อมูล วิธีการสุ่มแบบ uniform random sampling อาจพิจารณาได้ว่าเป็น biased sampling ที่กำหนดความน่าจะเป็นที่จะถูกเลือกให้กับข้อมูลทุกตัวเท่ากัน Palmer และ Faloutsos (2000) ได้พัฒนาวิธีการสุ่มแบบ biased เพื่อใช้ในงานจัดกลุ่มข้อมูล โดยใช้สมมติฐานว่าข้อมูลมีการกระจายแบบ Zipf และเทคนิคการสุ่มใช้ตารางแฮชช่วยในการทำงาน ดังนั้นปัญหาหลักของเทคนิคนี้คือ การชนกันของข้อมูล (collision) ซึ่งจะทำให้ประสิทธิภาพของการสุ่มลดลง

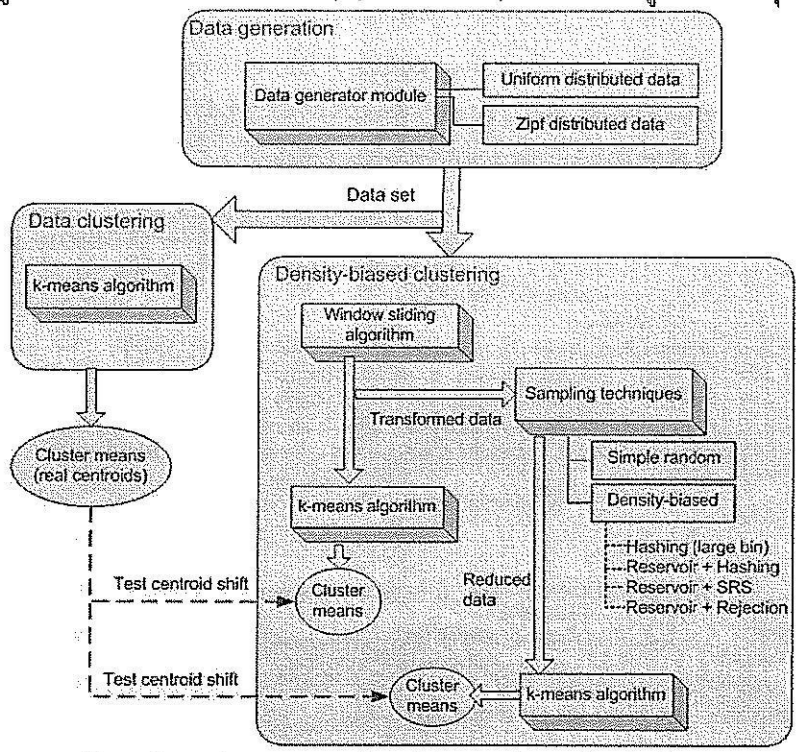
งานวิจัยนี้จึงมีวัตถุประสงค์หลักที่จะพัฒนาเทคนิค biased sampling เพื่องานจัดกลุ่มข้อมูลเช่นเดียวกับแนวคิดของ Palmer และ Faloutsos (2000) แต่เพื่อหลีกเลี่ยงปัญหาการชนกันของข้อมูล แทนที่จะใช้เทคนิคแฮชชิง โครงการวิจัยนี้จะใช้วิธีการสุ่มแบบลำดับ โดยมีโครงสร้างข้อมูลที่เรียกว่า reservoir (Vitter, 1985) เป็นโครงสร้างพื้นฐานแทนตารางแฮช reservoir เป็นเนื้อที่หน่วยความจำชั่วคราวที่ใช้เก็บข้อมูลตัวแทนระหว่างกระบวนการสุ่ม แนวคิดหลักของงานวิจัยนี้จึงเป็นการปรับปรุง random reservoir-sampling ให้เป็น biased reservoir-sampling การทดสอบผลด้านความถูกต้องของการสุ่มข้อมูลจะใช้วิธีให้โปรแกรมจัดกลุ่มข้อมูลประมวลผลหากลุ่มข้อมูลและจุดกึ่งกลางกลุ่มจากข้อมูลเริ่มต้น เปรียบเทียบกับค่าจุดกึ่งกลาง

กลุ่มที่ได้จากการจัดกลุ่มข้อมูลที่ผ่านกระบวนการสุ่มเพื่อดูความคลาดเคลื่อนของกลุ่ม ในด้านการทดสอบประสิทธิภาพของโปรแกรมจะใช้การวัดเวลาการประมวลผลของโปรแกรม ซึ่งจะรวมทั้งเวลาที่ใช้ในการสุ่มและเวลาที่ใช้จัดกลุ่มข้อมูล นอกจากนี้ยังมีการวัดประสิทธิภาพด้านการใช้เนื้อที่หน่วยความจำที่ใช้เก็บข้อมูลในช่วงของการสุ่มและการจัดกลุ่มข้อมูล

การพัฒนาโปรแกรมใช้ภาษาเออแลง (Erlang) ที่เป็นภาษาเชิงฟังก์ชันและมีวิธีการเขียนเชิงประกาศ (declarative) ซึ่งจะช่วยให้การพัฒนาโปรแกรมต้นแบบทำได้รวดเร็วและรูปแบบคำสั่งจะคล้ายกับซูโดโค้ด ทำให้ง่ายต่อการอ่านโปรแกรม นอกจากนี้ภาษาเชิงประกาศยังมีแนวคิดและรูปแบบที่เหมาะสมสำหรับงานที่จะพัฒนาเป็นฟังก์ชันอันดับสูง และปรับปรุงเป็นการประมวลผลฟังก์ชันแบบคู่ขนาน (concurrent) และแบบหลายหน่วยประมวลผล (multi-core) ต่อไปได้โดยง่าย การประมวลผลโปรแกรมภาษาเออแลงในงานวิจัยนี้ใช้คอมไพเลอร์รีลีส R13B04 (ซอฟต์แวร์นี้เป็นโอเพนซอร์สดาวน์โหลดได้จาก www.erlang.org)

กรอบแนวคิดและวิธีการ

กรอบแนวคิดและโมดูลต่างๆในแต่ละขั้นตอนของการดำเนินงาน สรุปเป็นแผนภาพได้ดังรูปที่ 1 ขั้นตอนเริ่มต้นตามแผนภาพเป็นโมดูล Data generation ทำหน้าที่สร้างข้อมูลสังเคราะห์ โดยสามารถกำหนดจำนวนกลุ่มข้อมูลและขนาดของข้อมูลในแต่ละกลุ่ม ถ้าข้อมูลมีลักษณะการกระจายอย่างสม่ำเสมอ (uniform distribution) ขนาดของข้อมูลในแต่ละกลุ่มจะใกล้เคียงกัน แต่ถ้าข้อมูลมีลักษณะการกระจายแบบซิฟ (Zipf distribution) ขนาดของข้อมูลในแต่ละกลุ่มจะเล็กใหญ่ต่างกันมาก



รูปที่ 1. กรอบแนวคิดและโมดูลต่างๆของการพัฒนาโปรแกรมสุ่มและจัดกลุ่มข้อมูลตามความหนาแน่น

ข้อมูลสังเคราะห์ที่สร้างจากโมดูล Data generation จะถูกนำไปใช้ในโมดูล Data clustering เพื่อให้โปรแกรม k-means (MacQueen, 1967) ทำหน้าที่จัดกลุ่มข้อมูลและรายงานผลลัพธ์ของการจัดกลุ่มเป็นค่าจุดกึ่งกลาง (cluster means, or centroids) ของแต่ละกลุ่ม ค่าจุดกึ่งกลางนี้จะใช้เป็นค่าจุดกึ่งกลางที่แท้จริง (real centroids) ของกลุ่มข้อมูล เพื่อเป็นเกณฑ์ในการเปรียบเทียบผลการจัดกลุ่มกับข้อมูลที่ได้จากการสุ่มแบบต่างๆ

ข้อมูลสังเคราะห์นอกจากใช้ในการคำนวณหา real centroids แล้ว ยังถูกนำไปใช้ในกระบวนการสุ่มข้อมูลตามความหนาแน่น (แผนภาพส่วน Density-biased clustering ของรูปที่ 1) โดยข้อมูลสังเคราะห์จะถูกส่งไปยังโมดูล Window sliding