**In The Name of Allah**

******

***Distributed AI (Summer 2012)***

*Implement of adversary strategy*

Presented by:

Farinaz Falahpour 908879

Nilofar Rastin 908876

هدف از انجام پروژه پیاده سازی Adversary Strategy بر روی دیتا ست spambase.data بوده است.

اسکریپت Main.m دیتاست را load می کند و به ازی هر instance روی دیتاست، تابع Adversay را صدا میزند. با توجه به این که زمان اجرای برنامه زیاد بود ما به نتایج خود را روی یک instance بدست آوردیم.

|  |
| --- |
| %% Main.M (By: Nilofar Rastin,Farinaz Falahpour)  clear all;  clc  %%  %\*\*\*\*\*\*\*\*\*\*\*\*calculating the size of features and instances\*\*\*\*\*\*\*\*\*\*\*\*\*  %%  Spambase=load('spambase.data');  Inst=Spambase;  [numInst numFeat]=size(Inst);  numFeat=numFeat-1; %%reducing the label column  %%  %\*\*\*\*\*\*\*\*\*\*because of very long run time for see the result we can run this  %part only for one instance; for example:j=9\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  %%  for j=1:numInst  Instance=(Inst(j,1:end-1));  Adversary(Inst,Instance,numFeat,numInst,j);  End |

تابع Advarsary با ورودی های مشخص شده، Gap را محاسبه میکند و بعد از محاسبه Gap تابع FindMCC را صدا میزند. با توجه به MinCost که تابع FindMcc به ما برمی گرداند، و همچنین UΔ تصمیم میگیرم که آیا عوض کردن Feature های یک instance به صرفه هست یا نه؟ برای بدست آوردن Lable های Estimate شده توسط Naïve Bayes از Static toolbox ،Matlab استفاده کرده ایم که محاسبه ی آن در فایل NBLabel.m موجود است

|  |
| --- |
| %% Adversary.m (By: Nilofar Rastin,Farinaz Falahpour)  %Inst is total Database  %Instance is one Instance from Data base  %numFeat is the number of features in database  %numInst is the number of instances in database  %%  function []=Adversary(Inst,Instance,numFeat,numInst,j)  %%  [myGap]=Gap(j,Inst,Instance,numFeat,numInst)  w=myGap  %%  %\*\*\*\*\*\*\*\*\*\*because of very long run time for see the result we can run this  %part for numFeat=3,w=5\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  %%  [MinCost,MinList]=FindMCC(numFeat,w,j,Inst)    save ('MinCost','MinCost')  save('MinList','MinList')  %%    deltaU=20;%UA(-,+)-UA(+,+)    load('NB.mat');    if (NB(j)==1 && MinCost< deltaU)  newx=Inst(j,:);  l=1;    for k=1 :(size(MinList,2)/2)  h=MinList(l);  newx(h)=MinList(k\*2);  l=l+2;  end    newx  end    end |

برای بدست آوردن Domain هر کدام از Feature ها از اسکریپت Domain.m استفاده کرده ایم و حاصل را در فایل Domain.mat ذخیره کرده ایم.

|  |
| --- |
| %% FindDomain.m (By: Nilofar Rastin,Farinaz Falahpour)  clear all;  clc;    Spambase=load('spambase.data');  Inst=Spambase;  [numInst numFeat]=size(Inst);  numFeat=numFeat-1;  %%  %Finding the Domain of all features and saving them in Domain Cell  %%  Domain=cell(numFeat,1);    for f=1:numFeat  Domain{f}=unique(Inst(:,f));  end    save('Domain','Domain'); |

تابع Gap با ورودی مشخص شده به ازای هر instance میزان Loc(x)-Lt(Uc) را بدست میاورد. برای محاسبه Loc(xi) از اسکریپت جداگانه ای(CalculatedLocxi.m ,Locxi.m) استفاده کرده ایم و آن را در فایل Out.mat ذخیره کرده ایم.

|  |
| --- |
| %% Gap.m (By: Nilofar Rastin,Farinaz Falahpour)  %% Gap=Loc(x)-Lt(Uc)  %Inst is total Database  %Instance is one Instance from Data base  %numFeat is the number of features in database  %numInst is the number of instances in database  %j is the index of instance(j=1 or 2 or ... 4096)  %%  function [myGap]=Gap(j,Inst,Instance,numFeat,numInst)    load('Domain.mat');  load('Out.mat');    loc=0;  %%  %calculating the sum of loc(xi) for one instance  %%  for i=1:numFeat  index=find(Instance(i)==Domain{i});  loc=loc+Out{i}(index);  end    SpamLable=1813;  EmailLable=2788;    %%  %Calculate email and spam probability P(+) and P(-)  %%  ProbEmail=EmailLable/numInst;%p(+)  ProbSpam=SpamLable/numInst;%p(-)  %%  %Calculate Log(P(+)/P(-))  %%  Prob=log(ProbEmail/ProbSpam)  totalLOCx=Prob+loc;  %%  %calculatin LT(uc)  %%  Uc00=1;%uc(-,-)=1  Uc11=1;%uc(+,+)=1  Uc01=-10;%uc(+,-)=-10 or -100 or -1000  Uc10=-1;%uc(-,+)=1  Ltc=log((Uc00-Uc10)/(Uc11-Uc01));  %%  %discritizing the gap  %%  myGap=round(totalLOCx-Ltc);      end |

تابع FindMcc یک تابع بازگشتی است که MinList و MinCost را به ما برمی گرداند.

|  |
| --- |
| %% FindMCC.m (By: Nilofar Rastin,Farinaz Falahpour)  %i is the total number of features(i=57)  %w is the discritized value of gap  %j is the index of instance(j=1 or 2 or ... 4096)  %Inst is the totlal database  %%  function [MinCost,MinList]=FindMCC(i,w,j,Inst)    if(w<=0)  MinCost=0;  MinList=[];  return  end    if(i==0)  MinCost=Inf;  MinList=NaN;  return  end      MinCost=Inf;  MinList=NaN;    load('Domain');  load('Out.mat');    for f=1:size(Domain{i},1)  index=find(Inst(j,i)==Domain{i});%find the cuurrent value of i'th feature  Deltaloc{i}(f)=Out{i}(index)-Out{i}(f);%out has the log(p(Xi=xi|+)/(p(Xi=xi|-)) for i'th feature and it's f'th value    if (Deltaloc{i}(f)>0)  [CurCost,CurList]=FindMCC((i-1),(w-Deltaloc{i}(f)),j,Inst)  CurCost=CurCost+0.2  x=Domain{i}(f);  CurList=[CurList i x ];      if (CurCost<MinCost)  MinCost=CurCost;  MinList=CurList;  end    end      end    MinCost=MinCost;  MinList=MinList ;    return  end |