Assignment 1: LogLinear graphical model

Ignacio Almodóvar Cárdenas

2022-04-27

Download and prepare data

This Dataset has been taken from Kaggle, which is an open source platform where data Scientist and other developers are engaged into running machine learning contests, write and share code, and to host datasets.

Specifically, this dataset is called Bank Marketing Dataset and its aim is to develop effective marketing campaigns for banks. Therefore, this dataset includes relevant information about a financial institution and it is made to analyze and find ways to look for future strategies.

```
bank=read.csv("bank.csv")
names(bank)
```

```
##  [1] "age"       "job"       "marital"   "education" "default"   "balance"
##  [7] "housing"   "loan"      "contact"   "day"       "month"     "duration"
## [13] "campaign"  "pdays"     "previous"  "poutcome"  "deposit"
```

As we can see this dataset contains 17 variables. It includes binary variables, categorical with several factors and continuous ones. However, as we want to apply a log linear model, we are only going to keep the binary variables.

```
reduced_bank=bank[,c(5,7,8,17)]
head(reduced_bank,3)
```

```
##   default housing loan deposit
## 1      no     yes   no     yes
## 2      no      no   no     yes
## 3      no     yes   no     yes
```

After this variable reduction, we will be working with 4 binary variables with labels “yes” and “no”.

- Default: Indicates if it has missed payments
- Housing: Not very sure about this one but I will consider it as having a mortgage with the bank.
- Loan: If it has a loan
- Deposit: Indicates if it has a deposit or not.

Once we have our dataset clear, we can start building the contingency and frequency tables.

```
data_bank=table(reduced_bank)%>% as.data.frame()
back_bank=xtabs(Freq~.,data=data_bank)
head(data_bank)
```

```
##   default housing loan deposit Freq
## 1      no      no   no      no 2089
## 2     yes      no   no      no   32
## 3      no     yes   no      no 2734
## 4     yes     yes   no      no   42
## 5      no      no  yes      no  380
## 6     yes      no  yes      no   26
```

Graphical Log-linear models

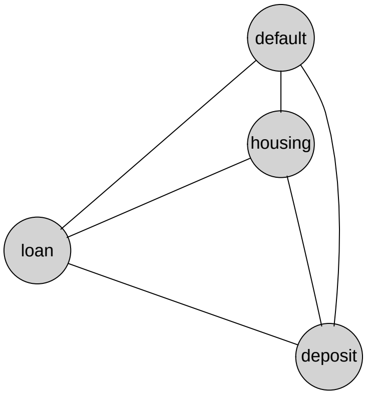
We are now going to try fitting a few different graphical log linear models to see the importance of the different dependencies between our categorical variables.

First of all we are going to check the satured model.

```
mt=dmod(~default*housing*loan*deposit,data=back_bank)
mt
```

```
## Model: A dModel with 4 variables
##  -2logL    :       40527.68 mdim :   15 aic :     40557.68
##  ideviance :         760.15 idf  :   11 bic :     40667.49
##  deviance  :           0.00 df   :    0
```

```
plot(mt)
```



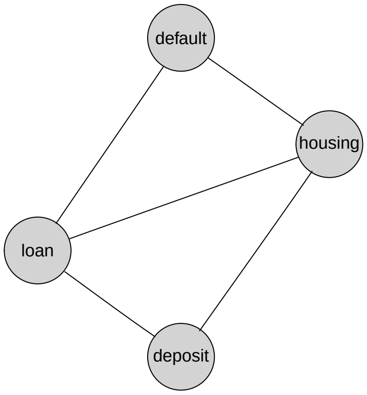
Within this plot we can see that all 4 nodes are connected between them, therefore there is a complete dependence between all the categorical variables. This makes total sense as all this nodes are like the steps you have to follow for getting a loan. When asking for a loan you will probably need to first have a deposit and same for applying for a mortgage. In some of the cases it could happen that due to different reasons you cannot do some of the payments that you agreed for the loan. Also, if you have a default it is probably that you cannot pay the loan or the mortgage but not so sure with the deposit, so maybe in future models we will see an independence between deposit and default.

Therefore we are going to fit now a model deleting that dependency and see if that dependency was important or not.

```
m1=dmod(~default*housing+default*loan+housing*loan+housing*deposit+deposit*loan,data=back_bank)
m1
```

```
## Model: A dModel with 4 variables
##  -2logL    :       40597.18 mdim :    9 aic :     40615.18
##  ideviance :         690.66 idf  :    5 bic :     40681.06
##  deviance  :          69.49 df   :    6
```

```
plot(m1)
```

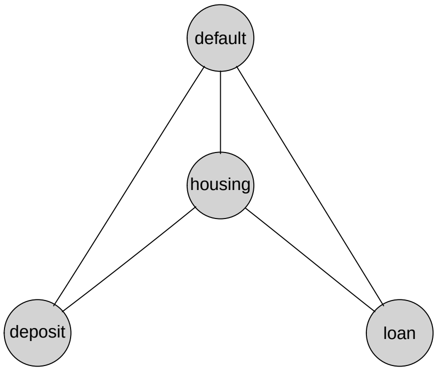


As it can be seen we no longer have that dependency between deposit and default. However, neither the AIC or BIC have improved against the first model fitted. Therefore, we could say that this relation is indeed important.

We are going to keep fitting different models and see it dependencies. Later on we will take a table summarizing both AIC and BIC measures to determine which model fits better our data.

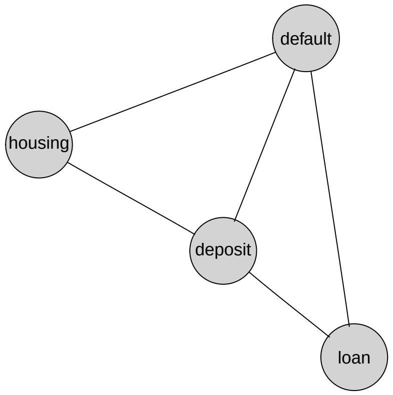
We now build a model without the loan and deposit connection. Which probably won't make sense as it might be mandatory to have a deposit in the bank before applying for a loan.

```
m2=dmod(~default*housing*deposit+default*housing*loan,data=back_bank)
plot(m2)
```



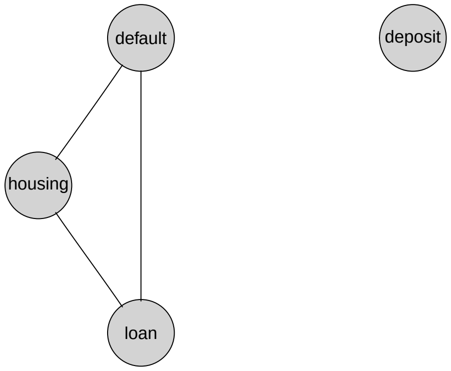
Now let's try another one but deleting the connection with housing and loan, which could make sense as it might be very risky for a bank to have both a loan and a mortgage with the same individual.

```
m3=dmod(~default*housing*deposit+default*loan*deposit,data=back_bank)
plot(m3)
```



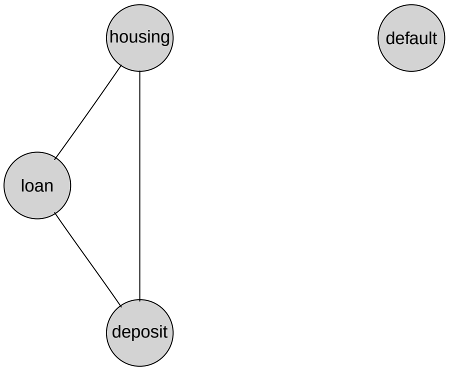
Now let's try isolating some variables. Let's see how it works with the deposit one.

```
m4=dmod(~default*housing*loan+deposit,data=back_bank)
plot(m4)
```



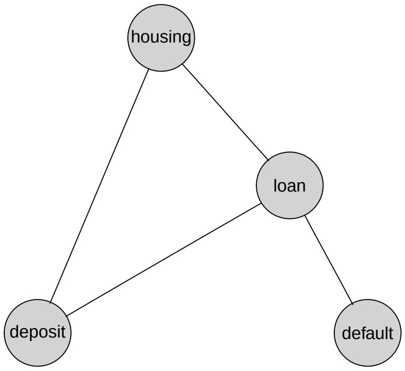
And then the default one, as might not be the best case as you probably need to have at least have something to pay in order to have a default.

```
m5=dmod(~housing*loan*deposit+default,data=back_bank)
plot(m5)
```



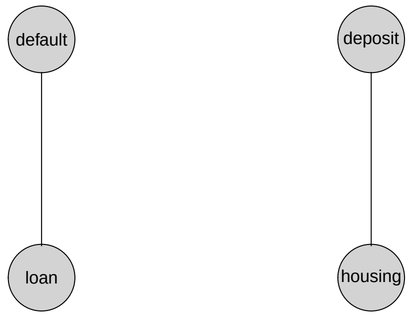
Now, let's try another model to see if the defaults are related with the loans.

```
m6=dmod(~housing*loan*deposit+default*loan,data=back_bank)
plot(m6)
```



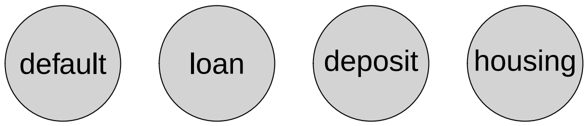
Now we are going to fit much easier models with just pair of connection to see how those work with our data. Lets start first by fitting one with a dependency between loan and default in one side and deposit with hosusing in the other.

```
m7=dmod(~default*loan+deposit*housing,data=back_bank)
plot(m7)
```



And finally another one with total independence between all the variables.

```
m8=dmod(~default+loan+deposit+housing,data=back_bank)
plot(m8)
```



We are now going to build a table to summarize all the models that we have fitted and look for the best one of them.

```
Aic=c(mt$fitinfo$aic,m1$fitinfo$aic,m2$fitinfo$aic,m3$fitinfo$aic,
      m4$fitinfo$aic,m5$fitinfo$aic,m6$fitinfo$aic,m7$fitinfo$aic,m8$fitinfo$aic)

Bic=c(mt$fitinfo$bic,m1$fitinfo$bic,m2$fitinfo$bic,m3$fitinfo$bic,
      m4$fitinfo$bic,m5$fitinfo$bic,m6$fitinfo$bic,m7$fitinfo$bic,m8$fitinfo$bic)

logL=c(mt$fitinfo$logL,m1$fitinfo$logL,m2$fitinfo$logL,m3$fitinfo$logL,
       m4$fitinfo$logL,m5$fitinfo$logL,m6$fitinfo$logL,m7$fitinfo$logL,m8$fitinfo$logL)

resume=data.frame(logL,Aic,Bic)
resume
```

```
##        logL      Aic      Bic
## 1 -20263.84 40557.68 40667.49
## 2 -20298.59 40615.18 40681.06
## 3 -20332.27 40686.55 40767.07
## 4 -20304.18 40630.36 40710.88
## 5 -20580.12 41176.24 41234.80
## 6 -20303.46 40622.92 40681.48
## 7 -20279.08 40576.15 40642.04
## 8 -20385.67 40783.35 40827.27
## 9 -20643.92 41295.84 41325.12
```

First of all we are going to focus on the AIC measure.

```
which.min(resume$Aic)
```

```
## [1] 1
```

According to the AIC, the best model is the one with all the interactions between them, which makes sense as we explained before. All of these variables are very related between them in a financial institution. Also, for the best graphical model, this one is the best one.

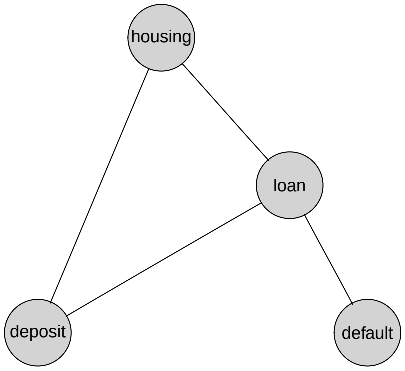
Then, focusing on the BIC:

```
which.min(resume$Bic)
```

```
## [1] 7
```

For this criteria we are getting that the best model is the sevent one. Recall that the plot for this model was:

```
plot(m6)
```



As it can be seen, it explains the dependency of default with just the variable loan. Meaning that, most of the defaults are for loan payments, which makes sense at all. Also it includes that for both loans and housing, it is common to have a deposit.

We have fitted several models (9) and we have obtained different solutions. Even though both of them makes sense we still have 55 models left that we did not calculate. It could happen that between those, the solutions obtained are better than the ones that we have already obtained. Therefore, we are going to compute now stepwise selection, both forward and backward to see if we can obtain better results or not.

Let's first start with AIC in both backward selection and forward selection

```
msat <- dmod(~.^.,data=back_bank)
mbaic <- backward(msat)
mmain <- dmod(~.^1,data=back_bank)
mfaic <- forward(mmain)
```

And now let's do the same with BIC:

```
msat2 <- dmod(~.^.,data=back_bank)
mbbic <- backward(msat,k=log(sum(back_bank)))
mmain2 <- dmod(~.^1,data=back_bank)
mfbic <- forward(mmain,k=log(sum(back_bank)))
```

We can also search without the restriction to decomposable models. The fitted decomposable and unrestricted models do not always have the same graphs

```
par(mfrow=c(2,4))
mbaic_unrestricted <- backward(msat,type="unrestricted")
plot(mbaic)
plot(mbaic_unrestricted)
mfaic_unrestricted <- forward(mmain,type="unrestricted")
```

```
##   change.AIC  465.7276 Edge added: housing,deposit
##   change.AIC  137.3905 Edge added: deposit,loan
##   change.AIC   69.8005 Edge added: housing,loan
##   change.AIC   46.7656 Edge added: loan,default
##   change.AIC    9.3146 Edge added: deposit,default
##   change.AIC    9.1556 Edge added: housing,default
## No edges can be added
```

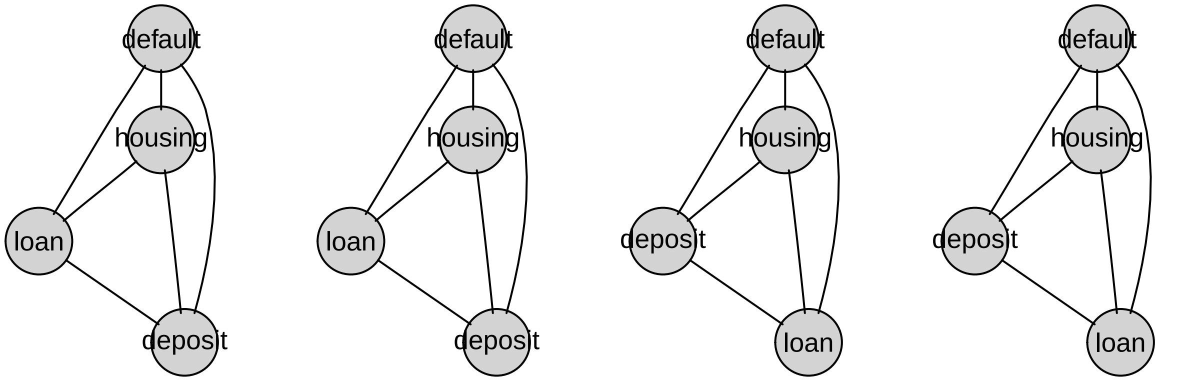
```
plot(mfaic)
plot(mfaic_unrestricted)
mbbic_unrestricted <- backward(msat,type="unrestricted",k=log(sum(back_bank)))
```

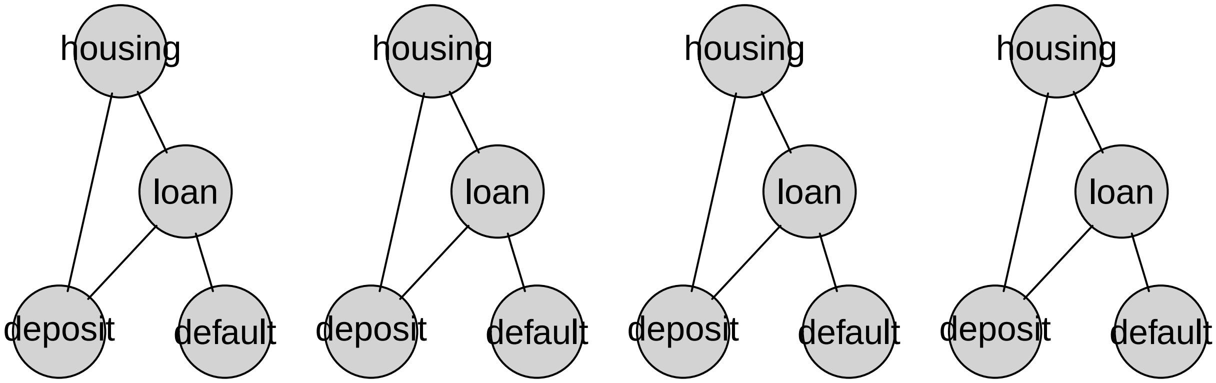
```
##   change.AIC  -20.1255 Edge deleted: default,housing
##   change.AIC   -5.3259 Edge deleted: deposit,default
```

```
plot(mbbic)
plot(mbbic_unrestricted)
mfbic_unrestricted <- forward(mmain,type="unrestricted",k=log(sum(back_bank)))
```

```
##   change.AIC  458.4074 Edge added: housing,deposit
##   change.AIC  130.0702 Edge added: deposit,loan
##   change.AIC   55.1600 Edge added: housing,loan
##   change.AIC   39.4453 Edge added: loan,default
```

```
plot(mfbic)
plot(mfbic_unrestricted)
```





We can see that the plots obtained for are the sames as the one obtained before doing the step wise selection, both for the AIC and BIC respectively.

We have explained already explained the relation with the variables in the graphical representation before, so we are now going to carry out a goodness fit test to see whether the models selected fits the data or not.

Goodness of fit test

We are now going to check if the models fits the data or not. We are going to include all the models that we have considered as good models since we started. That is those obtained with the stepwise selection for AIC and BIC measures including the unrestricted cases and the first two models that we fitted at the beginning.

```
pchisq(mt$fitinfo$dev,mt$fitinfo$dimension[4])
```

```
## [1] 0
```

```
pchisq(m6$fitinfo$dev,m6$fitinfo$dimension[4])
```

```
## [1] 0.9896656
```

```
pchisq(mbaic$fitinfo$dev,mbaic$fitinfo$dimension[4])
```

```
## [1] 0
```

```
pchisq(mfaic$fitinfo$dev,mfaic$fitinfo$dimension[4])
```

```
## [1] 0
```

```
pchisq(mbbic$fitinfo$dev,mbbic$fitinfo$dimension[4])
```

```
## [1] 0.9896656
```

```
pchisq(mfbic$fitinfo$dev,mfbic$fitinfo$dimension[4])
```

```
## [1] 0.9896656
```

```
pchisq(mbaic_unrestricted$fitinfo$dev,mbaic$fitinfo$dimension[4])
```

```
## [1] 0
```

```
pchisq(mfaic_unrestricted$fitinfo$dev,mfaic$fitinfo$dimension[4])
```

```
## [1] 0
```

```
pchisq(mbbic_unrestricted$fitinfo$dev,mbbic$fitinfo$dimension[4])
```

```
## [1] 0.9896656
```

```
pchisq(mfbic_unrestricted$fitinfo$dev,mfbic$fitinfo$dimension[4])
```

```
## [1] 0.9896656
```

As we can see we are obtaining the same results for each model selected with AIC and each one selected with BIC, therefore we can say that the models obtained were the same. However, we can also see that for AIC we are obtaining a p-value of 0, which is less than 0.05. Therefore we can see that for those models selected with the AIC measure we can reject the independence between variables.

In the other hand, for models selected with BIC, the p-value is above 0.05, therefore we can say that there exist an independence between some variables.

Conclusions

Through this whole analysis we have learn how to apply log-linear graphical model to see how different binary variables are related between them. This has definitely been a great exercise to practice and to better understand how this model works.

It has also been very interesting to figure out whether the dependencies obtained were reasonable or not, and indeed they were (at least for my very poor knowledge about banking procedures).

It would have also been very cool to see a dataset with variables that were not that directly related so the different graphs obtained could really change a lot from the satured model.

Finally, learning these tools have been a really good experience as it is very useful to carry out analysis between several binary variables which is something that we haven't learned before.