Bayesian networks

Aaron Smith

5/30/2020

Graphical models treat all variables as predictors and as target variables.

The bnlearn package has structure-learning algorithms. This is the strongest aspect of the package.

bnlearn supports data sets with all factor variables very well. bnlearn has features to support models for data sets with all numeric variables. bnlearn has the fewest features for mixed data types.

# The hailfinder data set

options(digits = 1)  
require(bnlearn)

## Loading required package: bnlearn

##   
## Attaching package: 'bnlearn'

## The following object is masked from 'package:stats':  
##   
## sigma

data(hailfinder,package = "bnlearn") ## all factors  
summary(hailfinder)

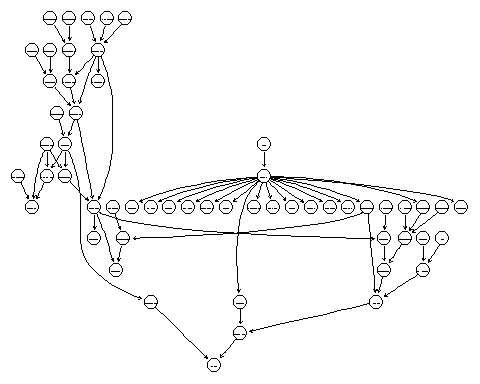
## N07muVerMo SubjVertMo QGVertMotion CombVerMo   
## Down :4937 Down : 4032 Down : 3968 Down :3433   
## Neutral :5044 Neutral :10013 Neutral :10111 Neutral :8782   
## StrongUp:4984 StrongUp: 2975 StrongUp: 2968 StrongUp:2251   
## WeakUp :5035 WeakUp : 2980 WeakUp : 2953 WeakUp :5534   
##   
##   
##   
## AreaMesoALS SatContMoist RaoContMoist CombMoisture AreaMoDryAir   
## Down :3433 Dry :5030 Dry :5005 Dry :3496 Dry :3329   
## Neutral :8782 Neutral:8044 Neutral:7931 Neutral:7278 Neutral:6777   
## StrongUp:2251 VeryWet:3053 VeryWet:2999 VeryWet:3713 VeryWet:4220   
## WeakUp :5534 Wet :3873 Wet :4065 Wet :5513 Wet :5674   
##   
##   
##   
## VISCloudCov IRCloudCover CombClouds CldShadeOth AMInstabMt   
## Clear : 7952 Clear :8142 Clear :8860 Clear :8254 None :6619   
## Cloudy: 1964 Cloudy:2864 Cloudy:3152 Cloudy:4085 Strong:6808   
## PC :10084 PC :8994 PC :7988 PC :7661 Weak :6573   
##   
##   
##   
##   
## InsInMt WndHodograph OutflowFrMt MorningBound Boundaries   
## None : 4111 DCVZFavor :5994 None :7845 None :10090 None :4542   
## Strong:11737 Other :3974 Strong:7012 Strong: 2921 Strong:6042   
## Weak : 4152 StrongWest:5030 Weak :5143 Weak : 6989 Weak :9416   
## Westerly :5002   
##   
##   
##   
## CldShadeConv CompPlFcst CapChange LoLevMoistAd   
## Marked:6626 DecCapIncIns:4562 Decreasing :4562 Negative :6043   
## None :7335 IncCapDecIns:8241 Increasing :8241 Neutral :5850   
## Some :6039 LittleChange:7197 LittleChange:7197 StrongPos:2415   
## WeakPos :5692   
##   
##   
##   
## InsChange MountainFcst Date Scenario   
## Decreasing :6202 SIG :7980 Aug11\_Aug20:1430 B :2725   
## Increasing :7199 SVR :3553 Aug20\_Sep15:4394 E :2682   
## LittleChange:6599 XNIL:8467 Jul16\_Aug10:4289 K :2500   
## Jul2\_Jul15 :2120 J :2056   
## Jun15\_Jul1 :2615 G :1949   
## May15\_Jun14:5152 C :1876   
## (Other):6212   
## ScenRelAMCIN MorningCIN AMCINInScen CapInScen   
## AB : 3905 None : 3012 Average :8675 Average :5761   
## CThruK:16095 PartInhibit :11480 LessThanAve:6710 LessThanAve:6725   
## Stifling : 3973 MoreThanAve:4615 MoreThanAve:7514   
## TotalInhibit: 1535   
##   
##   
##   
## ScenRelAMIns LIfr12ZDENSd AMDewptCalPl AMInsWliScen   
## ABI :5403 LIGt0 : 1956 Instability:6048 Average :6702   
## CDEJ:8228 LILt\_8 : 1578 Neutral :5053 LessUnstable:6585   
## F : 698 N1GtLIGt\_4:10316 Stability :8899 MoreUnstable:6713   
## G :1949 N5GtLIGt\_8: 6150   
## H :1222   
## K :2500   
##   
## InsSclInScen ScenRel34 LatestCIN LLIW   
## Average :5416 ACEFK:8936 None :7925 Moderate :7610   
## LessUnstable:6964 B :2725 PartInhibit :8016 Strong :3650   
## MoreUnstable:7620 D :1614 Stifling :2994 Unfavorable:2355   
## GJ :4005 TotalInhibit:1065 Weak :6385   
## HI :2720   
##   
##   
## CurPropConv ScnRelPlFcst PlainsFcst N34StarFcst R5Fcst   
## Moderate:4151 B :2725 SIG : 4812 SIG : 5021 SIG :8864   
## None :6724 E :2682 SVR : 3017 SVR : 3178 SVR :6141   
## Slight :4949 K :2500 XNIL:12171 XNIL:11801 XNIL:4995   
## Strong :4176 J :2056   
## G :1949   
## C :1876   
## (Other):6212   
## Dewpoints LowLLapse MeanRH MidLLapse   
## HighEvrywher:2724 CloseToDryAd:5098 Average :8697 CloseToDryAd:5918   
## LowAtStation:3345 ModerateOrLe:4867 Dry :7370 ModerateOrLe:6985   
## LowEvrywhere:2999 Stable :3350 VeryMoist:3933 Steep :7097   
## LowMtsHighPl:3588 Steep :6685   
## LowNHighS :1894   
## LowSHighN :3330   
## Other :2120   
## MvmtFeatures RHRatio SfcWndShfDis SynForcng   
## MarkedUpper:4403 DryMMoistL:6934 DenvCyclone :3223 LittleChange:6599   
## NoMajor :9497 MoistMDryL:6104 DryLine :3924 NegToPos :3371   
## OtherRapid :3314 Other :6962 E\_W\_N :2716 PosToNeg :3705   
## StrongFront:2786 E\_W\_S :2227 SigNegative :3354   
## MovingFtorOt:2567 SigPositive :2971   
## None :3876   
## Other :1467   
## TempDis WindAloft WindFieldMt WindFieldPln   
## Moving :2915 AllElse:5324 LVorOther: 9523 DenvCyclone:3692   
## None :8687 LV :4335 Westerly :10477 E\_NE :2453   
## Other :5349 NWQuad :4378 LongAnticyc:3337   
## QStationary:3049 SWQuad :5963 LV :4535   
## SEQuad :2862   
## WidespdDnsl:3121   
##

# A quick example of a graphical model

Let’s look a an example of a graphical model.

model\_hailfinder <- hc(hailfinder)  
graphviz.plot(model\_hailfinder)

## Loading required namespace: Rgraphviz



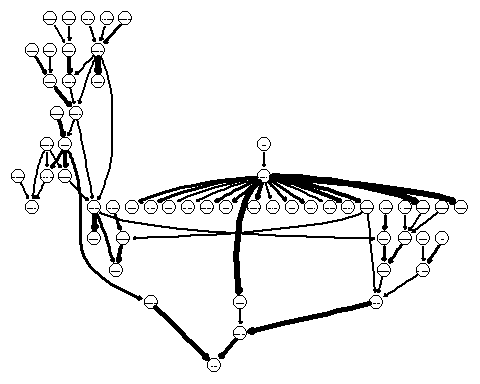
# Strength plots

Strength plots visually shows the strength between variables.

There are three types of errors.

* Includeing an arc/edge when there should not be one
* Missing an arc/edge when there should be one
* Arc pointed in the wrong direction

arc.strength\_hailfinder <- arc.strength(  
 x = model\_hailfinder,  
 data = hailfinder  
)  
strength.plot(  
 x = model\_hailfinder,  
 strength = arc.strength\_hailfinder  
)



# Other plots

Unfortunately, the bnlearn package has multiple object classes that fill similar purposes. To be able to use all of the features of the package, we need to be able to convert object classes.

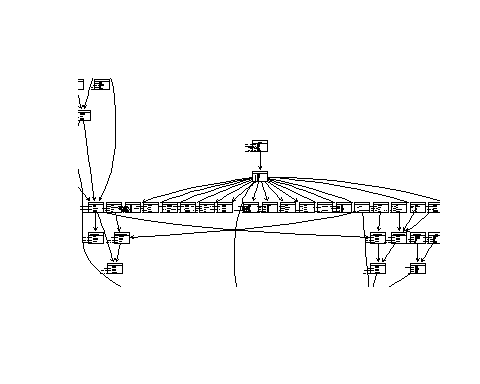
These plots only work for discrete data sets.

bn\_hailfinder <- bn.fit(  
 x = model\_hailfinder,  
 data = hailfinder  
)  
graphviz.chart(  
 x = bn\_hailfinder,  
 type = "barchart"  
)

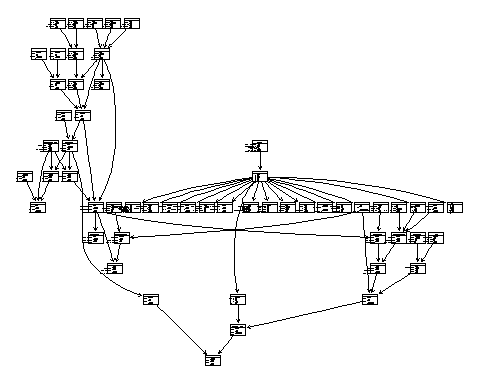
## Loading required namespace: gRain

##   
## Attaching package: 'gRbase'

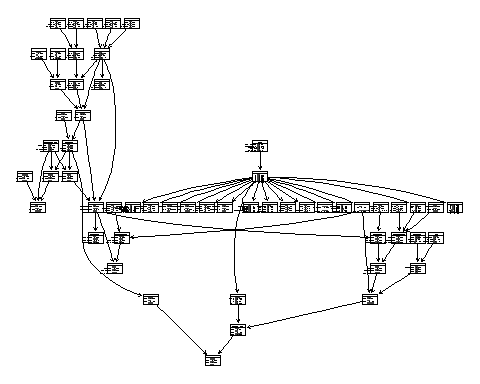
## The following objects are masked from 'package:bnlearn':  
##   
## ancestors, children, parents



graphviz.chart(  
 x = bn\_hailfinder,  
 type = "dotplot"  
)



graphviz.chart(  
 x = bn\_hailfinder,  
 type = "barprob"  
)



# Model fitting

In machine learning, we fit multiple models. Measure the quality of each model and select the best one.

With graphical models, all variables are predictors and target variables.

bnlearn supports using cross-validation well. In this example, we will use repeated two-fold cross-validation.

v\_models <- c(  
 "pc.stable","gs","iamb","fast.iamb","inter.iamb","iamb.fdr",  
 "hc", "tabu",  
 "mmhc","rsmax2","h2pc",  
 "mmpc","si.hiton.pc","hpc",   
 "chow.liu","aracne"  
)  
  
list\_cv <- list()  
for(j in v\_models) try({   
 list\_cv[[j]] <- bn.cv(  
 data = hailfinder,  
 bn = j,  
 k = 2,  
 runs = 2  
 )  
},silent = TRUE)  
list\_cv

## $gs  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Grow-Shrink   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 63   
## standard deviation of the loss: 0.2   
##   
##   
## $iamb  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: IAMB   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.02   
##   
##   
## $fast.iamb  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Fast-IAMB   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.02   
##   
##   
## $inter.iamb  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Inter-IAMB   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.02   
##   
##   
## $iamb.fdr  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: IAMB-FDR   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.004   
##   
##   
## $hc  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Hill-Climbing   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 49   
## standard deviation of the loss: 0.009   
##   
##   
## $tabu  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Tabu Search   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 49   
## standard deviation of the loss: 0.006   
##   
##   
## $mmhc  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Max-Min Hill-Climbing   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.02   
##   
##   
## $rsmax2  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Two-Phase Restricted Maximization   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 57   
## standard deviation of the loss: 0.01   
##   
##   
## $h2pc  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Hybrid^2 Parent Children   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 58   
## standard deviation of the loss: 0.01   
##   
##   
## $chow.liu  
##   
## k-fold cross-validation for Bayesian networks  
##   
## target learning algorithm: Chow-Liu   
## number of folds: 2   
## loss function: Log-Likelihood Loss (disc.)   
## number of runs: 2   
## average loss over the runs: 51   
## standard deviation of the loss: 3e-04

list\_mean <- list()  
for(j in names(list\_cv)){  
 for(k in 1:length(list\_cv[[j]])){  
 list\_mean[[j]][[k]] <- rep(NA,length(list\_cv[[j]][[k]]))  
 for(l in 1:length(list\_cv[[j]][[k]])){  
 list\_mean[[j]][[k]][l] <- list\_cv[[j]][[k]][[l]]$loss  
 }  
 }  
 list\_mean[[j]] <- unlist(list\_mean[[j]])  
}  
sort(base::sapply(X = list\_mean,FUN = mean))

## tabu hc chow.liu inter.iamb iamb.fdr fast.iamb mmhc   
## 49 49 51 57 57 57 57   
## rsmax2 iamb h2pc gs   
## 57 57 58 63