CaMML MB prior test

Kelvin

23 June 2017

```
source('~/PhDProjects/RStudioProjects/local2global/scripts/load_libraries.R')
```

```
## dirichletMultinomial.R :
## findCMB.R :
## findMB.R :
## getCumSumMmlMD.R :
## getPCMB.R :
## logFactorial.R :
## mb2stage.R :
## mbAccuracy.R :
## mmlCPT_fast.R :
## mmlCPT.R :
## mmlCPTRevised.R :
## mmlMultinomialDirichlet.R :
## mmlPC.R :
## multiParentsComputation.R :
## nodeToIndex.R :
## numeric2Nominal.R :
## proftable.R :
## singleParentComputation.R :
## symmetryCorrection.R :
## generateCPTsWithAsymmetricDir.R :
## randAdjmtx.R :
## randCPTs.R :
## randDag.R :
## randSeed.R :
## checkArc.R :
## computeRatio.R :
## cpts2dag.R :
## dagIsom.R :
## editDistDags.R :
## estProb.R :
## findCollider.R :
## isDag.R :
## learnBN.R :
## makePaString.R :
## matrix2dag.R :
## netica2bnlearn.R :
## parentsList2BN.R :
## powerset.R :
## threeWayInteraction.R :
## fisherMatrix.R :
## getIndicator.R :
## getInteractData.R :
## getInteractionIndices.R :
## intersectIndices.R :
## logDeterminant.R :
```

```
## makeFormula.R :
## mmlLogit.R :
## mmlLogit2ndOrder_addOneTermEachTime.R :
## mmlLogit2ndOrder_noParameterPrior.R :
## mmlLogit2ndOrder_revised1.R :
## mmlLogit2ndOrder_revised2.R :
## mmlLogit2ndOrder.R :
## negLogLike.R :
## getDataInfo.R :
## mbBackwardElemination.R :
## mbForwardSelection_fast.R :
## mbForwardSelection.R :
## mbForwardSelectionForMML2ndOrderLogit.R :
## mbForwardSelectionUsingMMLMultinomialDirichlet.R :
## mbGreedySearchNonDeterministic.R :
## mbGreedySearchWithLookAhead.R :
## mbSimulatedAnnealing.R :
## addArcs_greedy.R :
## arcCount.R :
## arcPrior.R :
## cammlPrior.R :
## computeCI.R :
## computeMMLMatrix.R :
## count2Prior.R :
## cpdag2dag.R :
## dfs_cycle.R :
## enumerateMBPTs.R :
## enumWithNoSp.R :
## extractArcs.R :
## featureUncertainty.R :
## find_bidirected_arcs.R :
## format_camml_prior.R :
## get_arcs.R :
## isDag.R :
## learnMBPT.R :
## mbEditDist.R :
## mBlkt.R :
## mbLocalStr.R :
## mergeDags.R :
## mergeMBPTs.R :
## mmlDag_fast.R :
## mmlDag.R :
## nMBPTs.R :
## polytree_exhaustive.R :
## randPolytree.R :
## refineMergedMBPT.R :
## sa.R :
## strAccuracy.R :
## string2numeric.R :
## substituteVar.R :
## txt2csv.R :
## varLevels.R :
```

Generate models and data

```
for (i in 1:10) {
  seed = randSeed()
  set.seed(seed)
  nvars = 15
  maxPar = 3
  maxArity = 3
  beta = 1
  n = 1000
  dag = randDag(nvars, maxPar)
  cpts = randCPTs(dag, maxArity, beta)
  data = rbn(cpts, n)
  name = paste0(c(nvars, maxPar, maxArity, beta, n, seed), collapse = "_")
  saveRDS(dag, paste0("~/Experiments/CaMML/MB prior test/Dag/", name, ".rds"))
  saveRDS(cpts, paste0("~/Experiments/CaMML/MB prior test/CPT/", name, ".rds"))
  write.csv(data, paste0("~/Experiments/CaMML/MB prior test/Data/", name, ".csv"),
            row.names = FALSE)
}
list.files("~/Experiments/CaMML/MB prior test/Data/")
```

Get the Markov blanket of each node. Firstly, test camml by giving the connections in each mb. That is by telling camml the true arcs in dag.

```
files = list.files("~/Experiments/CaMML/MB prior test/Dag/")
for (i in 1:length(files)) {
  dag = readRDS(paste0("~/Experiments/CaMML/MB prior test/Dag/", files[i]))
  arcList = bnlearn::arcs(dag)
  priors = rep(1, nrow(arcList))
  text = "arcs {"
  for (j in 1:nrow(arcList)) {
    text = paste(text, "\n", arcList[j, 1], "->", arcList[j, 2], priors[j], ";")
  }
  text = paste(text, "\n }")
  name = strsplit(files[i], ".rds")[[1]]
  write_file(text, paste0("~/Experiments/CaMML/MB prior test/Prior/", name, ".txt"))
}
list.files("~/Experiments/CaMML/MB prior test/Prior/")
```

Evaluate CaMML outputs with and without priors.

```
#ed = editDistDags
ed = bnlearn::shd
files = list.files("~/Experiments/CaMML/MB prior test/CaMML without prior/")
dags = list.files("~/Experiments/CaMML/MB prior test/Dag/")
ed_matrix = matrix(0, nrow = length(files) + 1, ncol = 2)
colnames(ed_matrix) = c("noPiror", "prior")
rownames(ed_matrix)[nrow(ed_matrix)] = "mean"
for (i in 1:length(files)) {
  x = netica2bnlearn(paste0("~/Experiments/CaMML/MB prior test/CaMML without prior/",
                            files[i]))
  y = netica2bnlearn(paste0("~/Experiments/CaMML/MB prior test/CaMML with prior/",
                            files[i]))
  dag = readRDS(paste0("~/Experiments/CaMML/MB prior test/Dag/", dags[i]))
  cammlWOPrior = parentsList2BN(x)
```

```
  cammlWPrior = parentsList2BN(y)
  ed_matrix[i, ] = c(ed(cammlWOPrior, dag), ed(cammlWPrior, dag))
}
ed_matrix[nrow(ed_matrix), ] = colMeans(ed_matrix[1:length(files), ])
ed_matrix
```

```
##      noPiror prior
## <NA>       0   0.0
## <NA>       2   1.0
## <NA>       9   0.0
## <NA>       0   0.0
## <NA>       0   0.0
## <NA>       5   0.0
## <NA>      14   0.0
## <NA>      10   0.0
## <NA>       8   0.0
## <NA>       2   0.0
## mean       5   0.1
```