Towards consistency in general dependency networks

José A. Gámez and Juan L. Mateo and José M. Puerta

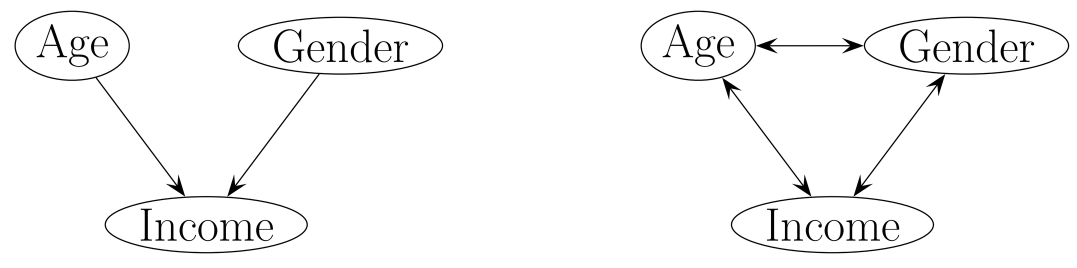
Computing Systems Department, University of Castilla-La Mancha, Albacete, Spain

Introduction

Dependency networks were proposed in [HCM00] as an alternative to Bayesian networks with some advantages like *visualization* or easier automatic learning if we deal with *general* DNs. The main difference is that they can encode cyclic relationship between variables. However this class of DNs has a drawback: inconsistency. In this work we propose an heuristic method to reduce this disadvantage.

Dependency networks

A DN has a similar definition to a BN: a directed graph (potentially cyclic) and a set of local probabilities distributions (LPD).



BN DN

In a DN the parents for each variable are those variables which make it independent of all the others.

- **Consistent DN**:
  A DN is *Consistent* if we can recover the joint probability distribution (JPD) of the domain through the set of LPD:

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|\mathbf{Pa}_i)$$

  Is very difficult to assure that condition when learning the model automatically with machine learning techniques. That is why authors proposed an alternative definition relaxing that requirement.

- **General DN**:
  In a general DN is not needed that the set of LPD to be fully consistent with the JPD.

$$P(\mathbf{X}) \approx \prod_{i=1}^{n} P(X_i|\mathbf{Pa}_i)$$

  In this way the model can be learned independently.

- **Inference**:
  Due to the existence of cycles we cannot use most of the inference algorithms for BNs. In [HCM00] is proposed Gibbs sampling and also is developed a framework with which we can avoid some sampling steps: *Modified ordered Gibbs sampler.* If we ask for a single variable and in the conditioning set are all its parents we do not need to perform sampling. We avoid Gibbs sampling, for instance, in classification or for computing likelihood for a model.

Analysis of Inconsistencies

Consider an example with two dependent variables X, Y, in a BN

$$P(X,Y) = P(X) \cdot P(Y|X) = P(Y) \cdot P(X|Y)$$

but in a DN

$$P(X,Y) \approx \hat{P}_{DN}(X,Y) = P(X|Y) \cdot P(Y|X) = P(X,Y) \cdot \left[\frac{P(X,Y)}{P(X) \cdot P(Y)}\right]$$

$\hat{P}$ is closer to P as the dependence between X and Y is lower!!!

In [HCM00] are proposed *probability decision trees* (PDT) to encode local distributions. This representation helps to reduce inconsistencies because reduces dependence between variables in it.

Proposal

Even in a case so simple the independent learning of the LPD lead to an inconsistent model.

The idea is to get a factorization as close as possible to a BN in the same domain by eliminating bidirectional relationship but only in the parametrical side. Besides we try to reduce the size of conditioning set which lead to better estimations from data.

foreach *variable* X_i **do**
  foreach Y_j *parent of* X_i **do**
    if X_i *is also a parent of* Y_j **then**
      if *the conditioning set of* X_i *is grater that* Y_j*'s* **then**
        Y_j is removed as parent of X_i
      else
        X_i is removed as parent of Y_j

The order in which variables are checked in first loop is important. We proposed to use the order imposed by the size of the LPD of each variable.

Experiments

network	Num. vars	States range	Aver. states	MB range	Aver. MB
`alarm`	37	2-4	2.84	1-12	3.89
`asia`	8	2-2	2.00	1-5	2.50
`car-starts`	18	2-3	2.06	1-9	3.44
`credit`	12	2-4	2.83	2-6	3.67
`headache`	12	1-4	2.92	1-4	2.67
`insurance`	27	2-5	3.30	1-16	6.22
`water`	16	3-4	3.63	1-12	6.00

$$score(d_1,\ldots,d_N|model) = -\frac{\sum_{i=1}^{N} \ln P(d_i|model)}{nN}$$

We test the real model (BN), empty model, and DN model with PDT or probability tables (PT). 'f' modifier means that we force to the real structure, and '*' means that we apply our proposed method to reduce inconsistencies.

Conclusions

Our proposal with PT achieves almost total consistency and also gets computational saving.

We plan to perform a deeper experimentation with more networks and with other kind of probabilistic queries. Also we want to test some modifications over our heuristic like the ordering for the variables.

References

[HCM00] D. Heckerman, D. M. Chickering, and C. Meek. Dependency networks for inference, collaborative filtering and data visualization. *Machine Learning Research*, 1:49–75, 2000.

	Empty	PT-f	PT-f*	PDT	PDT*	PDT-f	PDT-f*
`alarm`	0.115	0.110	**0.015**	0.029	0.060	0.040	0.056
`asia`	0.055	0.062	0.002	0.062	**0.000**	0.062	0.002
`car-starts`	0.048	0.057	**0.000**	0.057	0.009	0.057	**0.000**
`credit`	0.080	0.114	**0.007**	0.071	0.009	0.071	0.021
`headache`	0.174	0.222	**0.000**	0.017	0.150	0.017	0.158
`insurance`	0.161	0.092	**0.029**	0.070	0.066	0.071	0.059
`water`	0.009	0.016	0.010	0.013	**0.008**	0.013	0.007
	0.092	0.096	**0.009**	0.046	0.043	0.047	0.043

Absolute score difference between BN-f and the other models.

	BN-f	Empty	PT-f	PT-f*	PDT	PDT*	PDT-f	PDT-f*
`asia`	1.00	1.00	3.60	1.00	3.42	1.00	3.42	1.00
`car-starts`	1.00	1.00	20.04	1.08	11.40	1.00	11.40	1.00
`credit`	1.00	1.00	6.41	1.00	4.26	1.00	4.26	1.00
`headache`	1.00	1.00	29.68	1.00	5.70	1.00	5.70	1.00

Total joint probability for tested models.

dataset	% gain
`alarm`	43
`asia`	13
`car-starts`	23
`credit`	18
`headache`	11
`insurance`	95
`water`	98

Percentage of run time our proposal can reduce the original algorithm.