Pre and Post Counting for Scalable Statistical-Relational Model Discovery

Richard Mar and Oliver Schulte

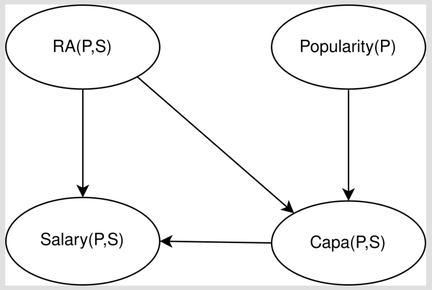
School of Computing Science

Simon Fraser University, Vancouver, Canada

Code Repository: https://github.com/sfu-cl-lab/FactorBase

First-Order Bayesian Networks

- Type of Bayesian network (BN) that can be learned from a relational dataset.
- Scoring metrics can be used to determine the best structure of the BN for a given dataset.



BDeu Scoring Metric

$$BDeu(B,D) = \log\left(P(B)\right) + \sum_{i=1}^{n}\sum_{j=1}^{q_i}\left(\log\left(\frac{\Gamma\left(\frac{N'}{q_i}\right)}{\Gamma\left(N_{ij}+\frac{N'}{q_i}\right)}\right) + \sum_{k=1}^{r_i}\log\left(\frac{N_{ijk}+\frac{N'}{r_i q_i}}{\Gamma\left(\frac{N'}{r_i q_i}\right)}\right)\right)$$

Contingency Tables

- Allow for efficient computation of the N_{ij} and N_{ijk} terms of the BDeu scoring metric.
- Applicable for other Bayesian network scoring metrics as well.
- Generating this data structure is a challenging problem!

Count	*Capa(P,S)*	*RA(P,S)*	*Salary(P,S)*
203	N/A	F	N/A
5	4	T	HIGH
4	5	T	HIGH
2	3	T	HIGH
1	3	T	LOW
2	2	T	LOW
2	1	T	LOW
2	2	T	MED
4	3	T	MED
3	1	T	MED

References

1. Zhensong Qian, Oliver Schulte, and Yan Sun. Computing Multi-Relational Sufficient Statistics for Large Databases. In Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, editors, Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pages 1249–1258. ACM, 2014

Precount Counts Caching Method

Algorithm 1 The PRECOUNT method: pre-compute ct-tables for each lattice point.

for each $latticePoint\ LP \in relationshipLattice$ **do**
  $ct_+(LP) \leftarrow \text{INNERJOIN}(\text{TABLES}(LP))$
  $ct(LP) \leftarrow \text{MÖBIUSJOIN}(ct_+(LP))$
end for
for each $family \in structureLearning$ **do**
  $ct(family) \leftarrow \text{PROJECT}(ct(LP), family)$
  $score \leftarrow \text{BDEU}(ct(family))$
end for

Ondemand Counts Caching Method

Algorithm 2 The ONDEMAND method: compute ct-tables for each family during structure search.

for each $family \in structureLearning$ **do**
  $ct_+(family) \leftarrow \text{INNERJOIN}(\text{TABLES}(family))$
  $ct(family) \leftarrow \text{MÖBIUSJOIN}(ct_+(family))$
  $score \leftarrow \text{BDEU}(ct(family))$
end for

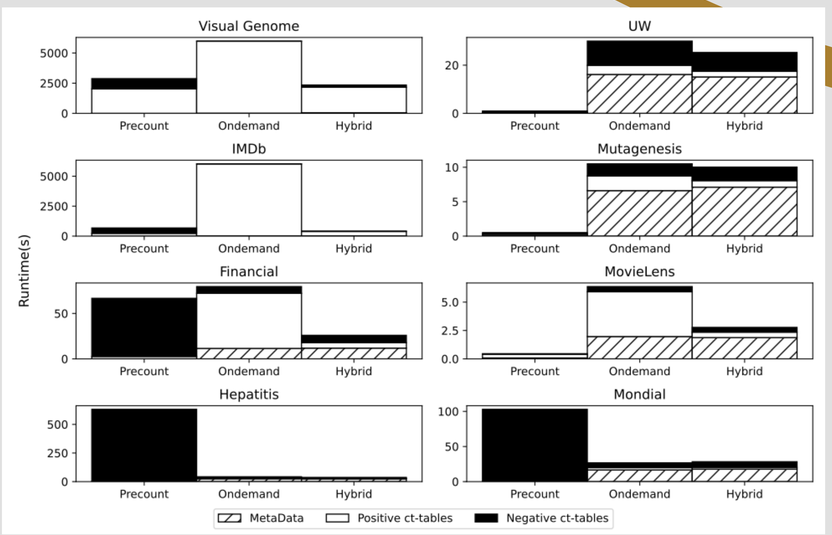
Hybrid Counts Caching Method

Algorithm 3 The HYBRID method: pre-compute positive ct-tables for each lattice point and compute ct-tables for each family during structure search.

for each $latticePoint\ LP \in relationshipLattice$ **do**
  $ct_+(LP) \leftarrow \text{INNERJOIN}(\text{TABLES}(LP))$
end for
for each $family \in structureLearning$ **do**
  $ct_+(family) \leftarrow \text{PROJECT}(ct_+(LP), family)$
  $ct(family) \leftarrow \text{MÖBIUSJOIN}(ct_+(family))$
  $score \leftarrow \text{BDEU}(ct(family))$
end for

All 3 methods use the Möbius Join algorithm (Qian et. Al 2014).

Results



DB	Estimated $ct(family)$ Count	Total Row	$ct(database)$ Total Row Count
MovieLens	816		239
Mutagensis	6075		1631
UW	15318		2828
Visual Genome	2923968		20447
Mondial	55800		1738867
Financial	930468		3013006
Hepatitis	176220		12374892
IMDb	33040		15537457

Conclusion

- Generating instantiation counts for relational data is a non-trivial task where pure pre and post count counting methods do not scale well.
- A hybrid approach where pre-counting is used to generate the counts for True relationships and post-counting is used to generate the counts for False relationships is best.
- Hybrid mitigates the numerous expensive operations of post-counting by using the pre-counting approach.