An Approach Based on Bayesian Networks for Query Selectivity Estimation

Max Halford[1,2] Philippe Saint-Pierre[1] Franck Morvan[2]

[1]Toulouse Institute of Mathematics (IMT)

[2]Toulouse Institute of Informatics Research (IRIT)

DASFAA 2019, Chiang Mai

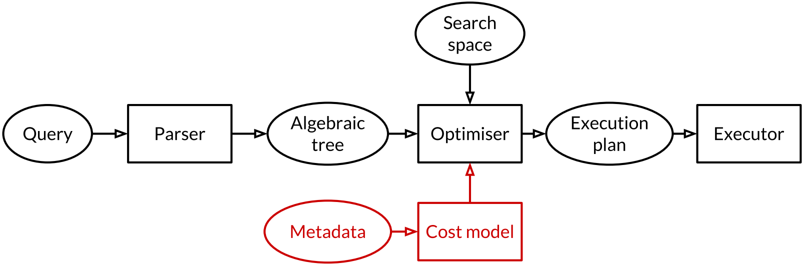
Introduction

Query optimisation

Assuming a typical relational database,

1. A user issues an SQL query
2. The query is compiled into a **execution plan** by a **query optimiser**
3. The plan is executed and the resulting rows are returned to the user
4. **Goal**: find the most efficient query execution plan

Cost-based query optimisation



Query optimisation time is part of the query response time!

Selectivity estimation

- An execution plan is a succession of operators (joins, aggregations, etc.)
- Cost of operator depends on number of tuples to process, which called the **selectivity**
- Selectivity is by far the most important parameter, but also the most difficult to estimate [WCZ+13]
- Errors propagate exponentially [IC91]

Example

```
SELECT *
FROM customers, shops, purchases
WHERE customers.id = purchases.customer_id
AND shops.id = purchases.shop_id
AND customers.nationality = 'Swedish'
AND customers.hair = 'Blond'
AND shops.city = 'Stockholm'
```

- Pushing down the selections is usually a good idea, so the best QEP should start by filtering the customers and the shops
- At some point the optimiser has to pick a join algorithm to join customers and shops
- How many Swedish blond customers are there? How about the number of shops in Stockholm?

Related work

- Statistics
  - Unidimensional [IC93, TCS13, HKM15] (textbook approach)
  - Multidimensional [GKTD05, Aug17] (exponential number of combinations)
  - Bayesian networks [GTK01, TDJ11] (complex compilation procedure and high inference cost)
- Sampling
  - Single relation [PSC84, LN90] (works well but has a high inference cost)
  - Multiple relations [Olk93, VMZC15, ZCL+18] (empty-join problem)
- Learning
  - Query feedback [SLMK01] (useless for unseen queries)
  - Supervised learning [LXY+15, KKR+18] (not appropriate in high speed environments)

Bayesian networks

A statistical point of view

- A relation is made of p attributes $X_1, \dots, X_p$
- Each attribute X_i follows an unknown distribution $P(X_i)$
- Think of $P(X_i)$ has a function/table which can tell us the probability of a predicate (e.g. `hair IS 'Blond'`)
- $P(X_i)$ can be estimated, for example with a histogram
- The distribution $P(X_i, X_j)$ captures interactions between X_i and X_j (e.g. `hair IS 'Blond' AND nationality IS 'Swedish'`)
- Memorising $P(X_1, \dots, X_p)$ takes $\prod_0^p |X_i|$ units of space

Independence

- Assume $X_1, \ldots, X_p$ are independent with each other
- We thus have $P(X_1, \ldots, X_p) = \prod_0^p P(X_i)$
- Memorising $P(X_1, \ldots, X_p)$ now takes $\sum_0^p |X_i|$ units of space
- We've compromised between accuracy and space
- In query optimisation this is called the **attribute value independence** (AVI) assumption

Conditional independence

- Bayes' theorem: $P(A, B) = P(B|A) \times P(A)$
- A are B are **conditionally independent** if C determines them
- In that case $P(A, B, C) = P(A|C) \times P(B|C) \times P(C)$
- $|P(A|C)| + |P(B|C)| + |P(C)| < |P(A, B, C)|$
- Conditional independence can save space without compromising on accuracy!

Example

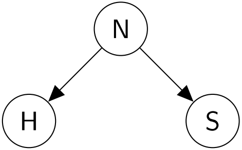
nationality	hair	salary
Swedish	Blond	42000
Swedish	Blond	38000
Swedish	Blond	43000
Swedish	Brown	37000
American	Brown	35000
American	Brown	32000

- Truth: $P(Swedish, Blond) = \frac{3}{6} = 0.5$
- With independence:
  - $P(Swedish) = \frac{4}{6}$
  - $P(Blond) = \frac{3}{6}$
  - $P(Swedish, Blond) \simeq P(Swedish) \times P(Blond) = \frac{2}{6} = 0.333$
- With conditional independence:
  - $P(Blond \mid Swedish) = \frac{3}{4}$
  - $P(Swedish, Blond) = P(Blond \mid Swedish) \times P(Swedish) = \frac{3 \times 4}{4 \times 6} = 0.5$

Bayesian networks

- Assuming full independence isn't accurate enough
- Memorising all the possible value interactions takes too much space
- Pragmatism: some variables are independent, some aren't
- Bayes' theorem + pragmatism = Bayesian networks
- Conditional independences are organised in a **graph**
- Each node is a variable and is dependent with it's parents

Example



American	Swedish
0.333	0.666

Table: $P(nationality)$

	Blond	Brown
American	0	1
Swedish	0.75	0.25

Table: $P(hair \mid nationality)$

	$<$ 40k	$>$ 40k
American	1	0
Swedish	0.5	0.5

Table: $P(salary \mid nationality)$

Structure learning

- In a tree, each node has 1 parent, benefits:
  - 2D conditional distributions (1D for the root)
  - Low memory footprint
  - Low inference time
- Use of Chow-Liu trees [CL68]
  1. Compute mutual information (MI) between each pair of attributes
  2. Let the MI values define fully connected graph G
  3. Find the maximum spanning tree (MST) of G
  4. Orient the MST (i.e. pick a root) to obtain a directed graph

Selectivity estimation

- Use of **variable elimination** [CDLS06]
- Works in $\mathcal{O}(n)$ time for trees [RS86]
- **Steiner tree** [HRW92] extraction to speed up the process

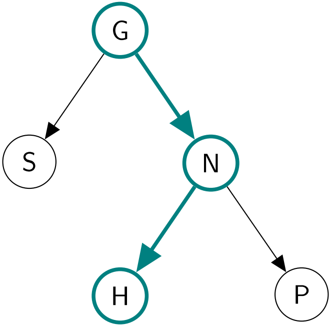


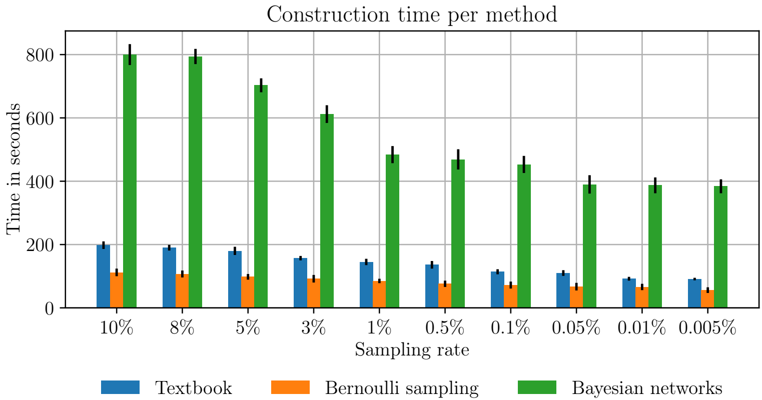
Figure: Highlighted Steiner tree containing nodes G, N, and H needed to compute H's marginal distribution

Experimental results

Setup

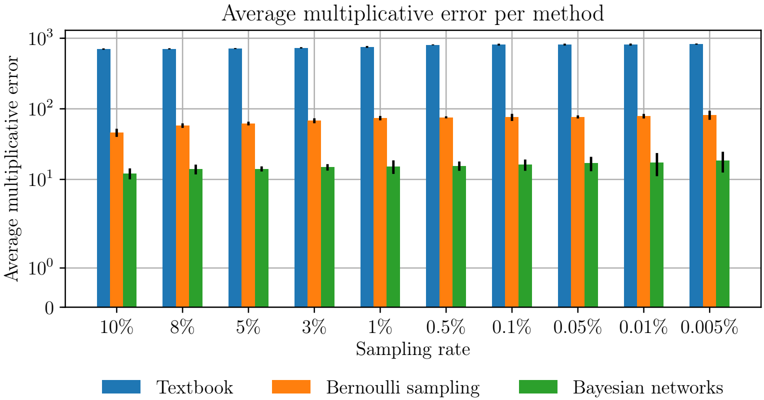
- We ran 8 queries from the TPC-DS benchmark with a scale of 20 over samples of the database 10000 times
- We compared
  - The "textbook approach" used by PostgreSQL
  - Bernoulli sampling
  - Bayesian networks (our method)
- All methods used the same samples
- We measured
  - Time needed to build the model
  - Accuracy of the cardinality estimates
  - Time needed to produce estimates
  - Values needed to store each model

Construction time



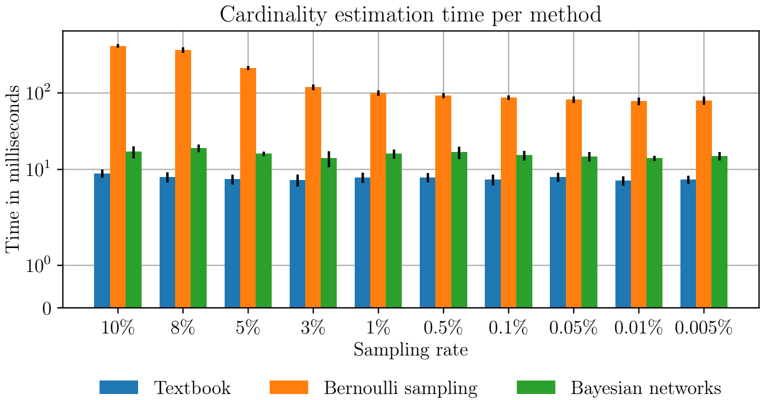
The Bayesian networks method, with a 10% sample size, requires on average a construction time of around 800 seconds.

Selectivity estimation accuracy



The Bayesian networks method, with a 10% sample size, produces estimates that are on average 10 times lower/higher than the truth.

Cardinality estimation time



The Bayesian networks method, with a 10% sample size, takes on average 35 milliseconds to produce an estimate.

Conclusion

- Sampling is the fastest to build
- The textbook approach is the quickest to produce estimates
- Bayesian networks are the most accurate

As expected, no free lunch! But a better compromise.

Thank you!

References I

Dariusz Rafal Augustyn.
Copula-based module for selectivity estimation of multidimensional range queries.
In *International Conference on Man–Machine Interactions*, pages 569–580. Springer, 2017.

Robert G Cowell, Philip Dawid, Steffen L Lauritzen, and David J Spiegelhalter.
Probabilistic networks and expert systems: Exact computational methods for Bayesian networks.
Springer Science & Business Media, 2006.

C Chow and Cong Liu.
Approximating discrete probability distributions with dependence trees.
IEEE transactions on Information Theory, 14(3):462–467, 1968.

References II

Dimitrios Gunopulos, George Kollios, J Tsotras, and Carlotta Domeniconi.
Selectivity estimators for multidimensional range queries over real attributes.
The VLDB Journal—The International Journal on Very Large Data Bases, 14(2):137–154, 2005.

Lise Getoor, Benjamin Taskar, and Daphne Koller.
Selectivity estimation using probabilistic models.
In *ACM SIGMOD Record*, volume 30, pages 461–472. ACM, 2001.

Max Heimel, Martin Kiefer, and Volker Markl.
Self-tuning, gpu-accelerated kernel density models for multidimensional selectivity estimation.
In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1477–1492. ACM, 2015.

References III

Frank K Hwang, Dana S Richards, and Pawel Winter.
The Steiner tree problem, volume 53.
Elsevier, 1992.

Yannis E Ioannidis and Stavros Christodoulakis.
On the propagation of errors in the size of join results, volume 20.
ACM, 1991.

Yannis E Ioannidis and Stavros Christodoulakis.
Optimal histograms for limiting worst-case error propagation in the size of join results.
ACM Transactions on Database Systems (TODS), 18(4):709–748, 1993.

References IV

Andreas Kipf, Thomas Kipf, Bernhard Radke, Viktor Leis, Peter Boncz, and Alfons Kemper.
Learned cardinalities: Estimating correlated joins with deep learning.
arXiv preprint arXiv:1809.00677, 2018.

Richard J Lipton and Jeffrey F Naughton.
Query size estimation by adaptive sampling.
In *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 40–46. ACM, 1990.

Henry Liu, Mingbin Xu, Ziting Yu, Vincent Corvinelli, and Calisto Zuzarte.
Cardinality estimation using neural networks.
In *Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering*, pages 53–59. IBM Corp., 2015.

References V

Frank Olken.
Random sampling from databases.
PhD thesis, University of California, Berkeley, 1993.

Gregory Piatetsky-Shapiro and Charles Connell.
Accurate estimation of the number of tuples satisfying a condition.
ACM Sigmod Record, 14(2):256–276, 1984.

Neil Robertson and Paul D. Seymour.
Graph minors. ii. algorithmic aspects of tree-width.
Journal of algorithms, 7(3):309–322, 1986.

Michael Stillger, Guy M Lohman, Volker Markl, and Mokhtar Kandil.
Leo-db2's learning optimizer.
In *VLDB*, volume 1, pages 19–28, 2001.

References VI

Hien To, Kuorong Chiang, and Cyrus Shahabi.
Entropy-based histograms for selectivity estimation.
In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1939–1948. ACM, 2013.

Kostas Tzoumas, Amol Deshpande, and Christian S Jensen.
Lightweight graphical models for selectivity estimation without independence assumptions.
Proceedings of the VLDB Endowment, 4(11):852–863, 2011.

David Vengerov, Andre Cavalheiro Menck, Mohamed Zait, and Sunil P Chakkappen.
Join size estimation subject to filter conditions.
Proceedings of the VLDB Endowment, 8(12):1530–1541, 2015.

References VII

Wentao Wu, Yun Chi, Shenghuo Zhu, Junichi Tatemura, Hakan Hacigümüs, and Jeffrey F Naughton.
Predicting query execution time: Are optimizer cost models really unusable?
In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 1081–1092. IEEE, 2013.

Zhuoyue Zhao, Robert Christensen, Feifei Li, Xiao Hu, and Ke Yi.
Random sampling over joins revisited.
2018.