Lightweight Graphical Models for Selectivity Estimation Without Independence Assumptions

Kostas Tzoumas, Amol Deshpande, Christian S. Jensen

Presented by Guozhang Wang

DB Lunch, Nov 23rd, 2011

Lightweight Graphical Models for Selectivity Estimation ~~Without~~ With Little Independence Assumptions

Kostas Tzoumas, Amol Deshpande, Christian S. Jensen

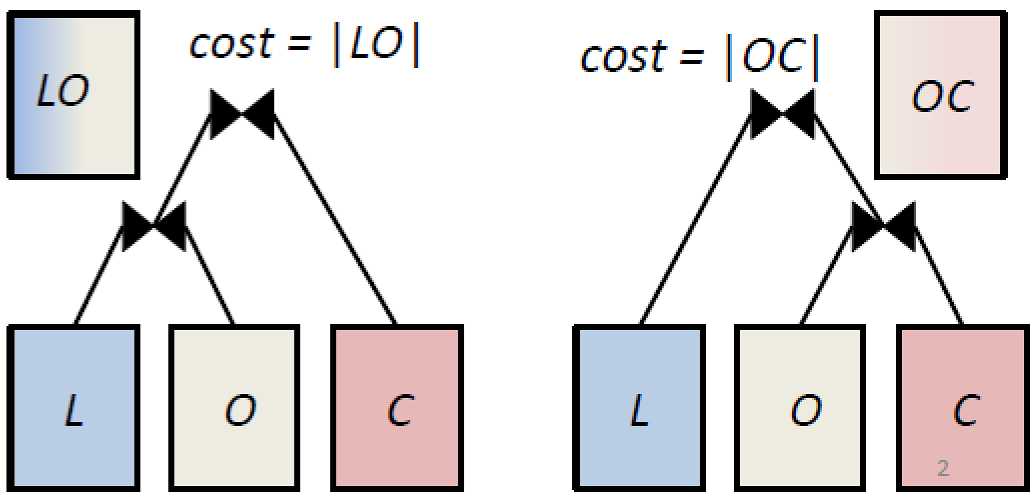
Presented by Guozhang Wang

DB Lunch, Nov 23rd, 2011

Motivation: Query Optimization

- The “best” join plan
  - Cost = # intermediate tuples along the path

- Errors in estimates lead to wrong plan



Selectivity Estimation

```
select c_name,c_address
from   lineitem,orders,customer
where  l_orderkey=o_orderkey and
       o_custkey=c_custkey and
       o_totalprice in [t1,t2] and
       l_extendedprice in [e1,e2] and
       c_acctbal in [b1,b2]
```

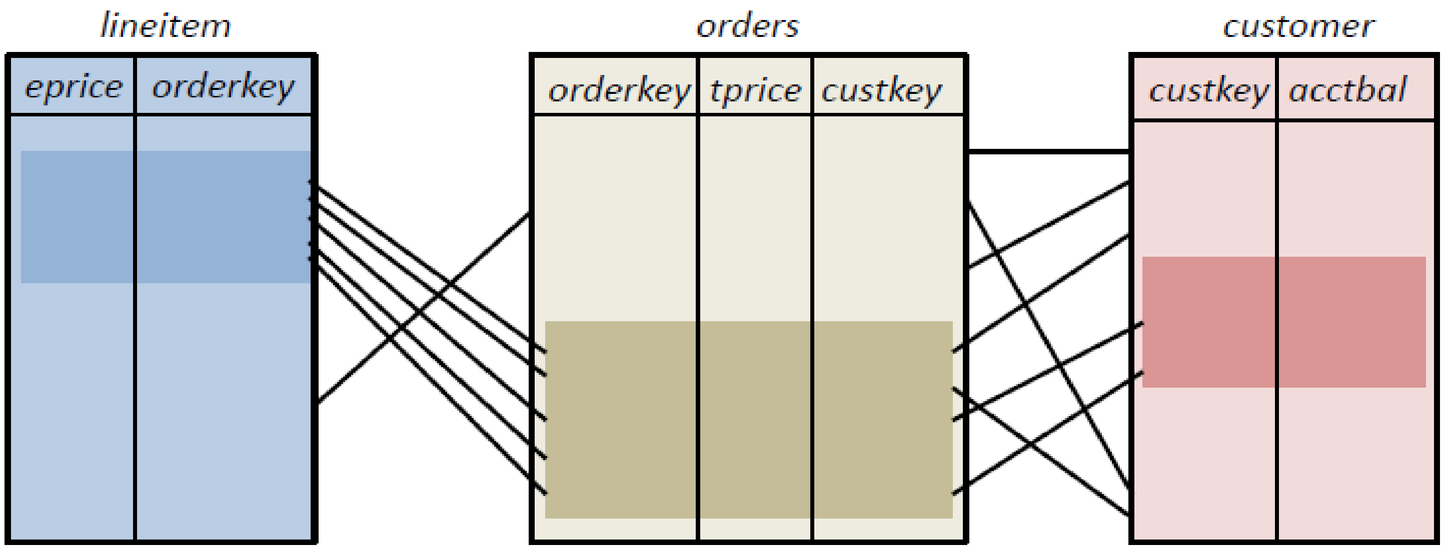
$$|LO| = |L| * |O| *$$

$$Pr(\, l.orderkey = o.orderkey \bigwedge o.totalprice \;\epsilon\; [t1, t2] \bigwedge l.extendedprice \;\epsilon\; [e1, e2])$$

- Equal to distribution estimation
  - Estimation based on histograms
  - How complicated histograms we need?

Correlations Matter in Estimation

- Need multi-dim. histograms to capture correlation between attributes




```
select c_name,c_address
from   lineitem,orders,customer
where  l_orderkey=o_orderkey and
       o_custkey=c_custkey and
       o_totalprice in [t1,t2] and
       l_extendedprice in [e1,e2] and
       c_acctbal in [b1,b2]
```

Idea #1: Full Independence

- Assume attributes are mutually indept.
  - Only need 1-D histograms, one for each attribute

- Estimates done by multiplication

$$Pr\left(l.ok = o.ok \bigwedge o.tp \in [t1, t2] \bigwedge l.ep \in [e1, e2] \right)$$

$$= Pr(l.ok = o.ok) * Pr(o.tp \in [t1, t2]) * Pr(l.ep \in [e1, e2])$$

Possible Big Error!

Idea #2: No Independence

- Any subsets of attributes could be correlated
  - Construct one n-Dim histogram, n = # total attributes in the database

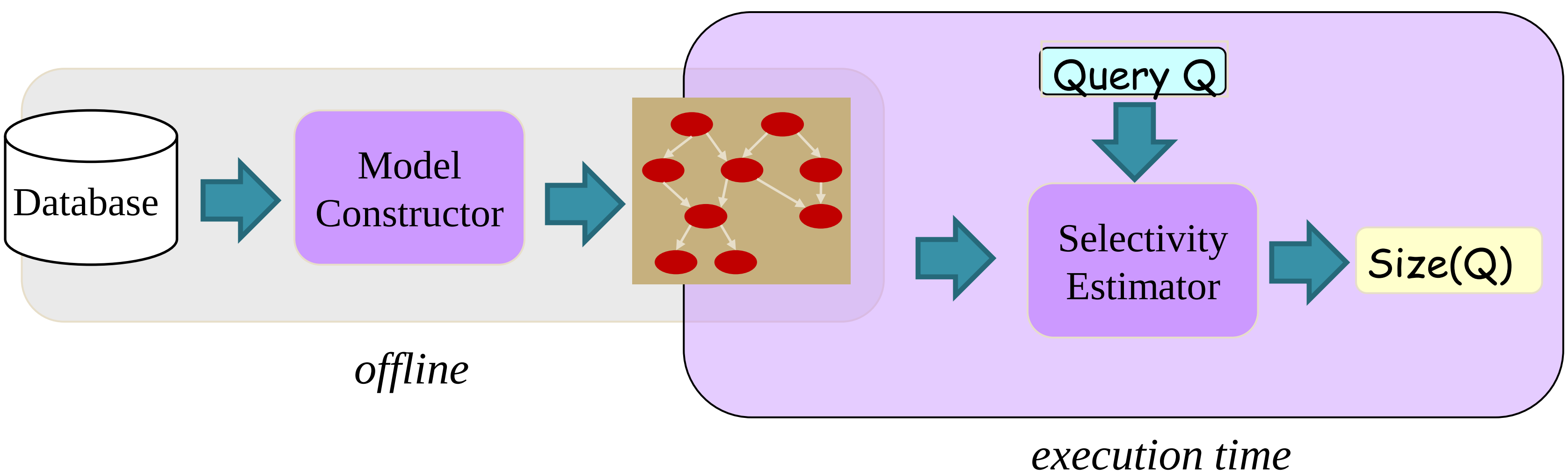
- Estimates done by marginalization

$$Pr\left(l.ok = o.ok \bigwedge o.tp \in [t1, t2] \bigwedge l.ep \in [e1, e2] \right) = \sum_{!l.ok\ldots} Pr(..)$$

Storage Blowup!

Idea #3: Cond. Independence [SIGMOD'01]

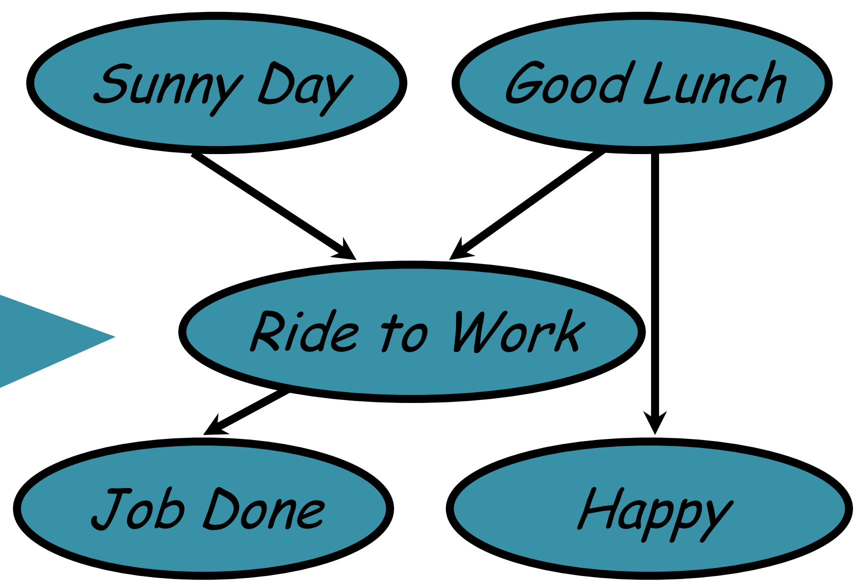
- Capture correlation in a Bayes network
  - BN model is constructed at start
  - Estimates done by computing joint dist'n



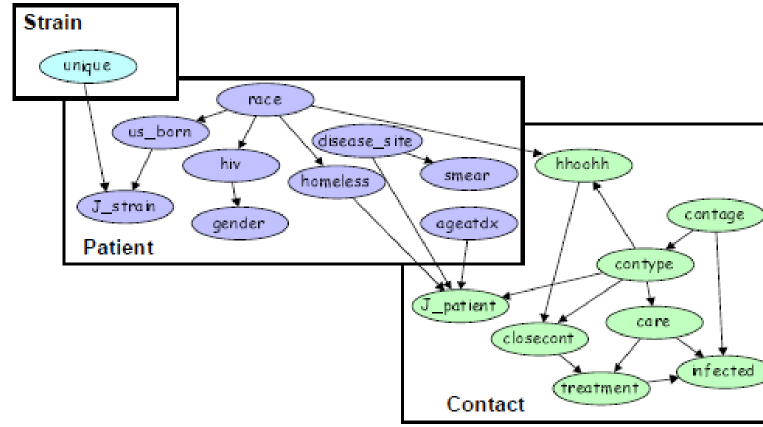
Bayesian Network

- Each node x has a conditional probability distribution $P(x \mid Pa(x))$
- Encodes independence in directed graph

S	*G*	*P(R|S, G)*	
t	*t*	0.8	0.2
t	*f*	0.6	0.4
f	*t*	0.2	0.8
f	*f*	0.01	0.99



BN Construct

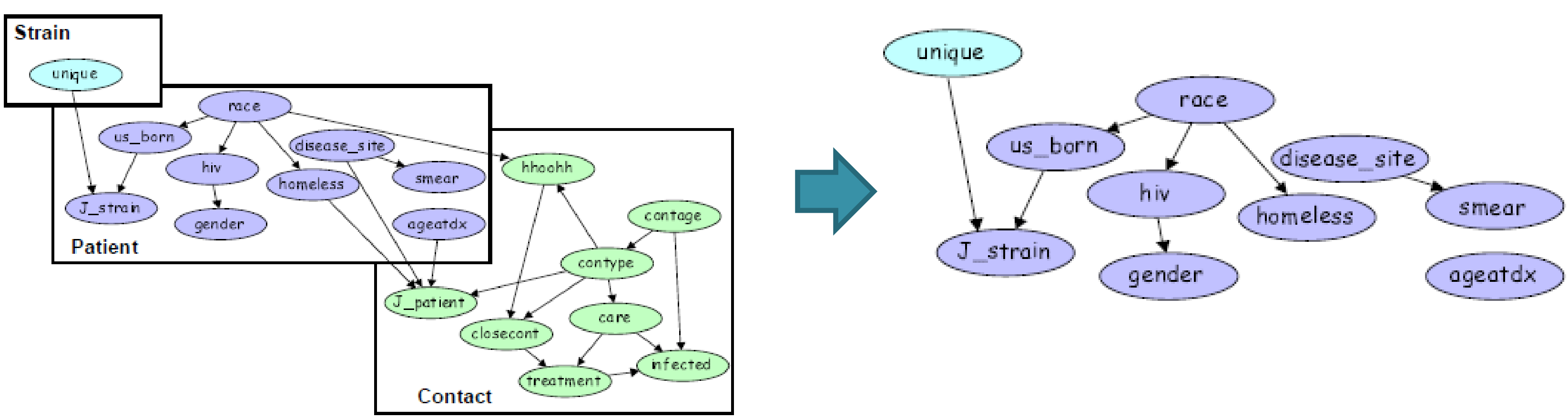


- Nodes
  - Each attribute: *a*
  - Each foreign key: join indicator J_f
- Edges
  - Find the model that maximize log-likelihood given data (greedy local structure search)
  - Parameters estimation after the structure is decided

Selectivity Estimation in BN

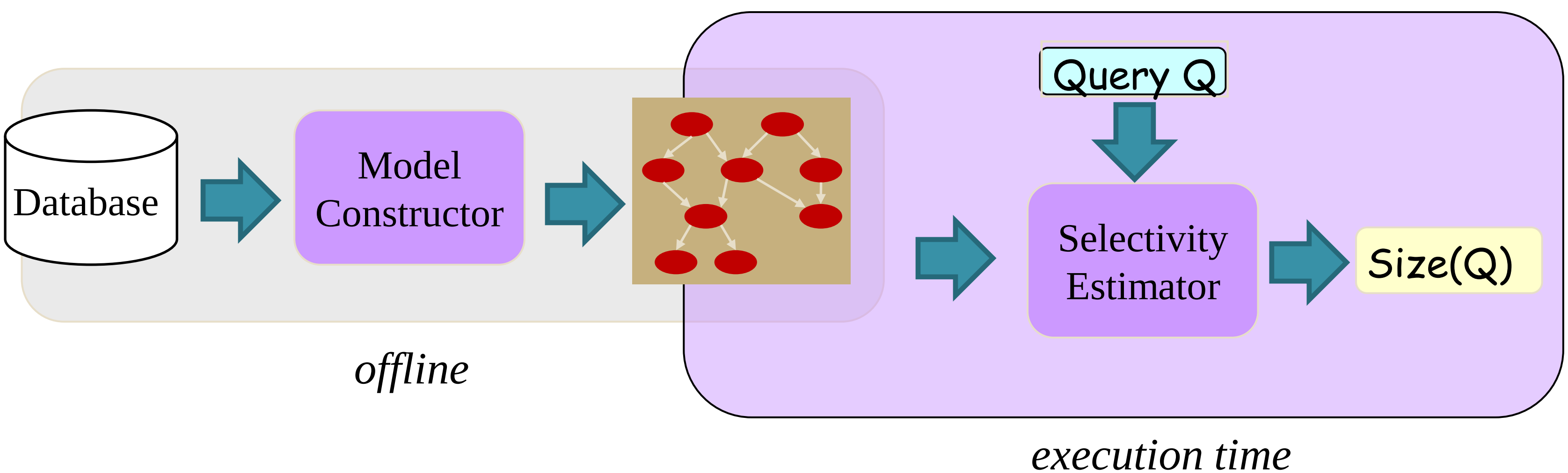
- Extend the query to include all the involved nodes' parents
- Multiplication along the graph to get the joint distribution

select *
from P, S
where P.Strain = Strain-ID ..

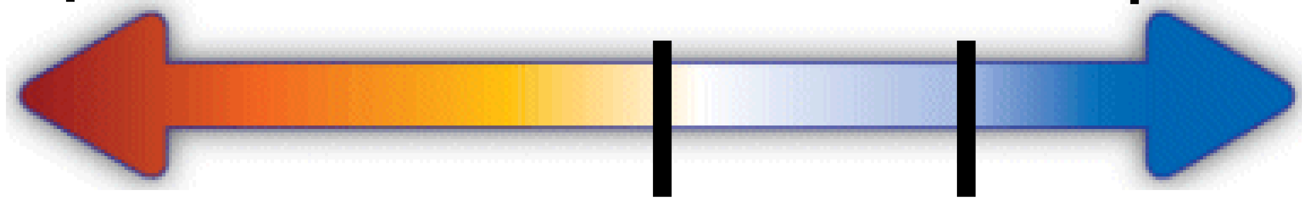


Cons. of BN

- Model is still too complicated…
  - Construction is expensive
  - Selectivity estimation is expensive



The Tradeoff Spectrum



- Red: better efficiency, worse accuracy
- Blue: better accuracy, worse efficiency

Idea #4: Constraint BN Dep.

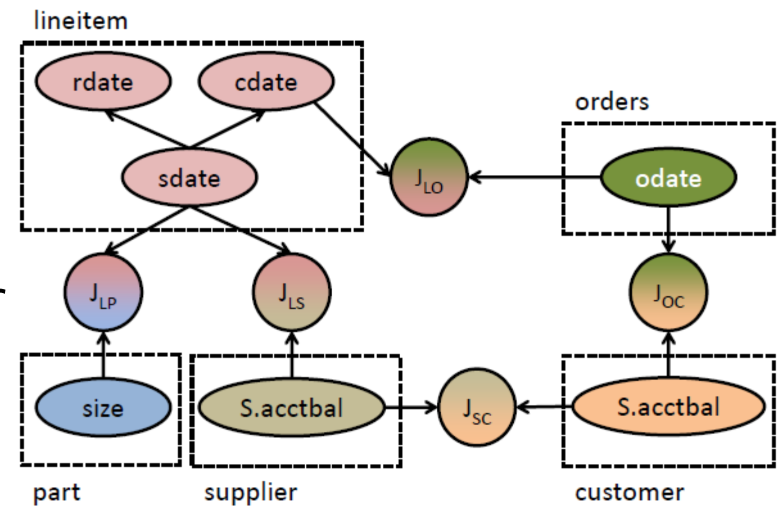
- Further restrict the structure of BN:
  - Acyclic [SIGMOD'01]
  - Fixed structure [this paper]
- Challenge: how to choose the fixed structure to get the good tradeoff in the spectrum?
  - Model simple enough for efficient algorithms
  - Model still capture important correlations

Fixed Structure BN

- Within a table
  - Attributes have at most one parent: tree structure
- Across a table
  - Joint indicators have at most two parents
  - No other cross-table edges
- 3D histograms only, scalable construction

Fixed Structure BN Construct

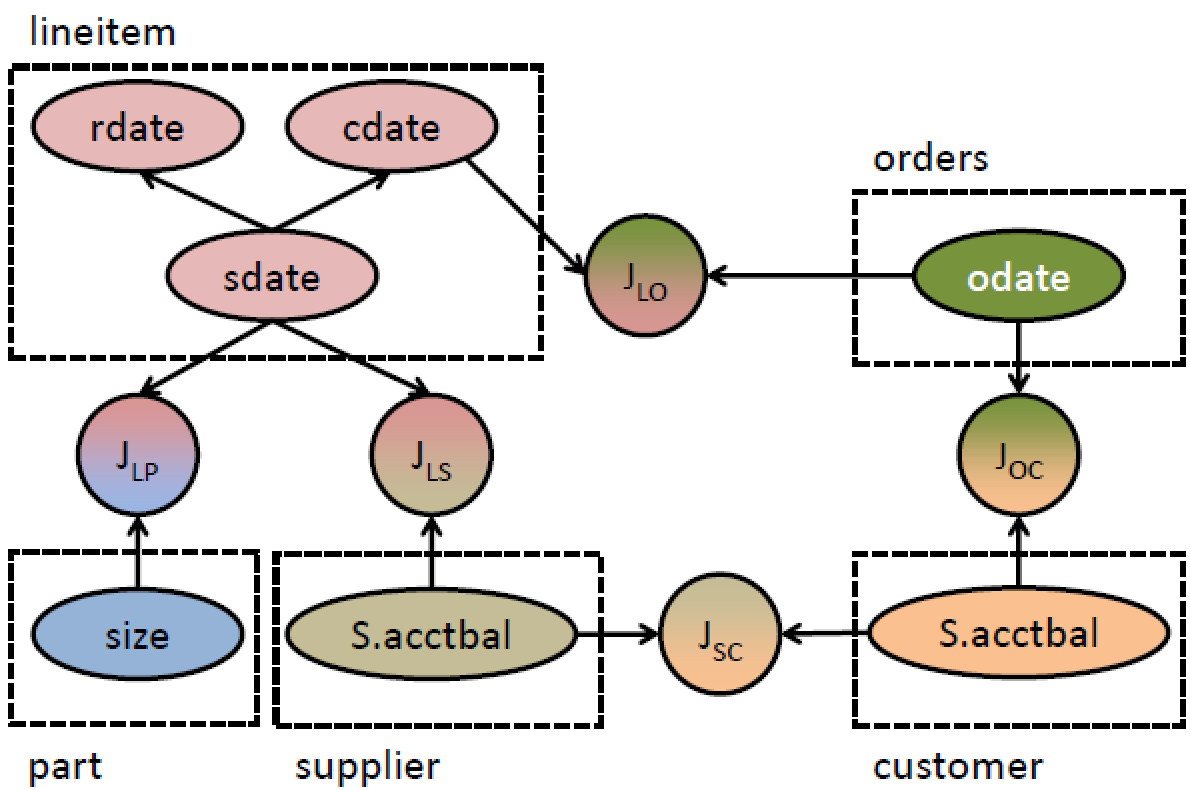
- Nodes
  - Each attribute *a*
  - Each join indicator J_f based on workload



- Edges
  - Within a table: maximum spanning tree
  - Across tables: best parent from each table
  - Weights based on mutual information

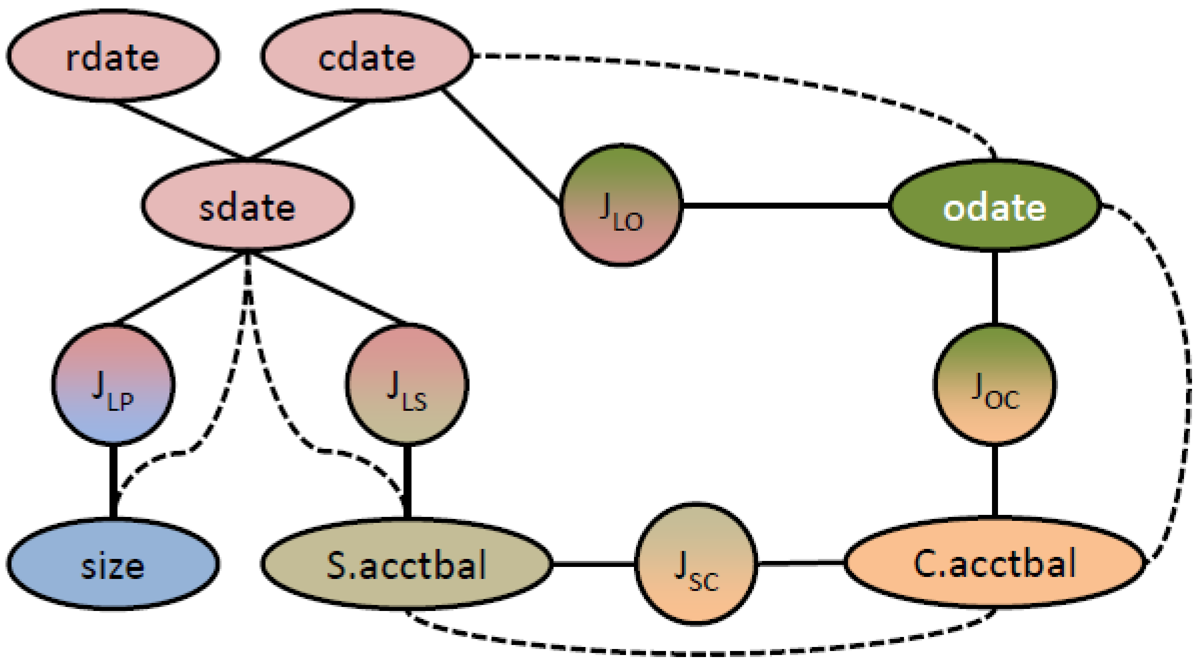
Selectivity Est. in Fixed BN

- Transform BN into a junction tree



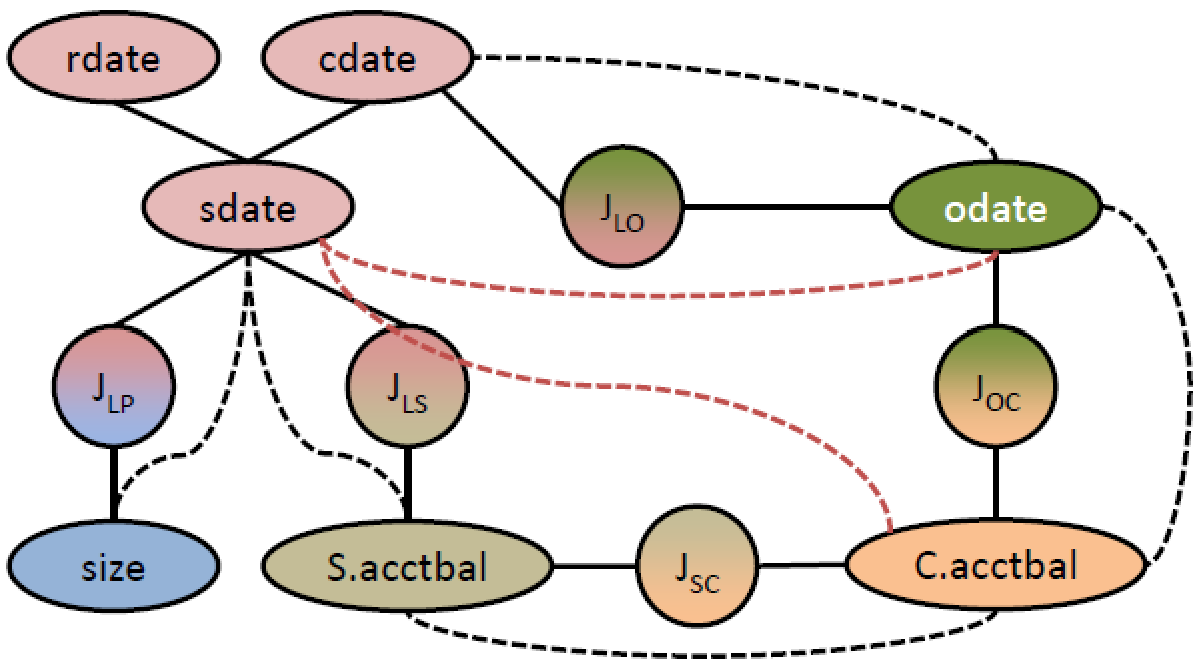
Selectivity Est. in Fixed BN

- Transform BN into a junction tree



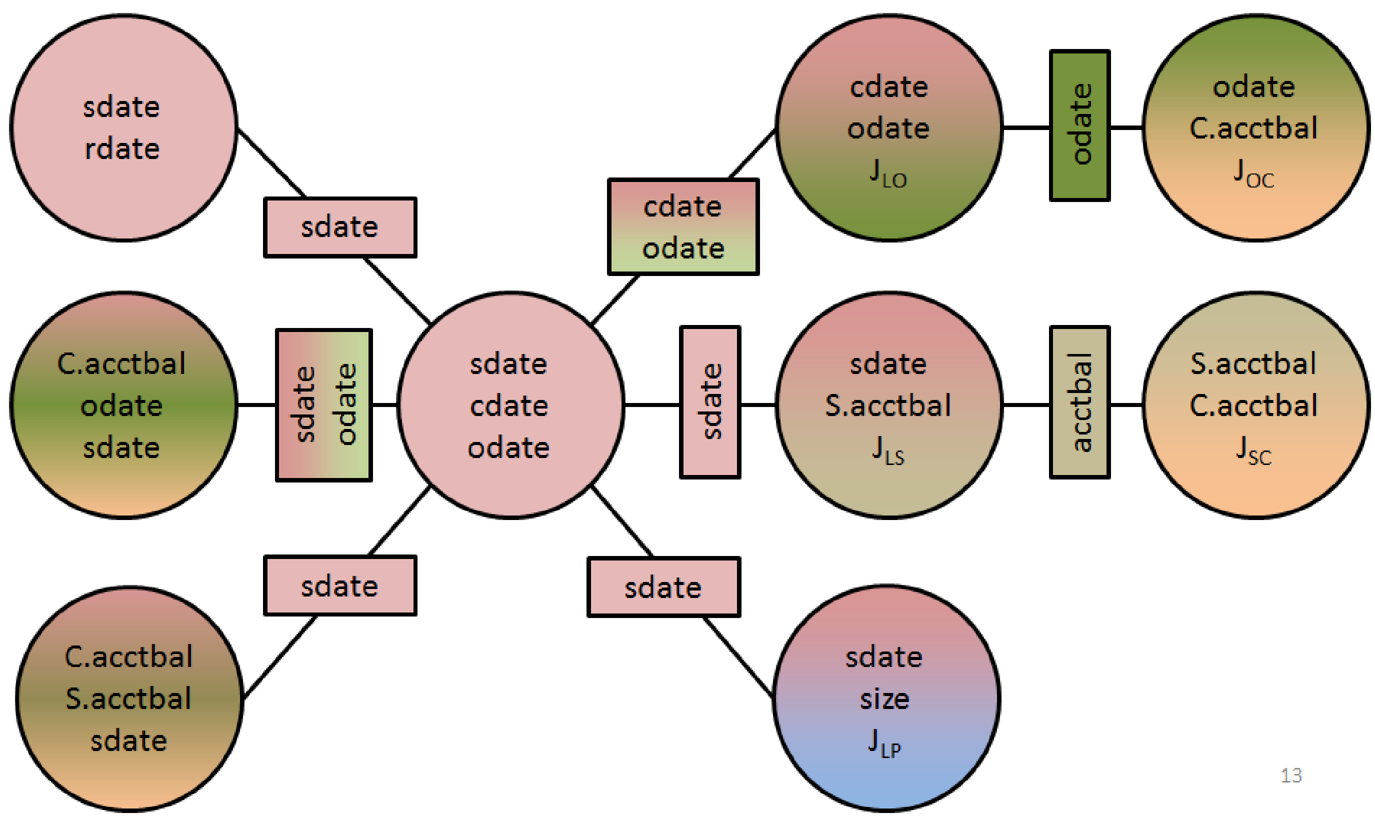
Selectivity Est. in Fixed BN

- Transform BN into a junction tree



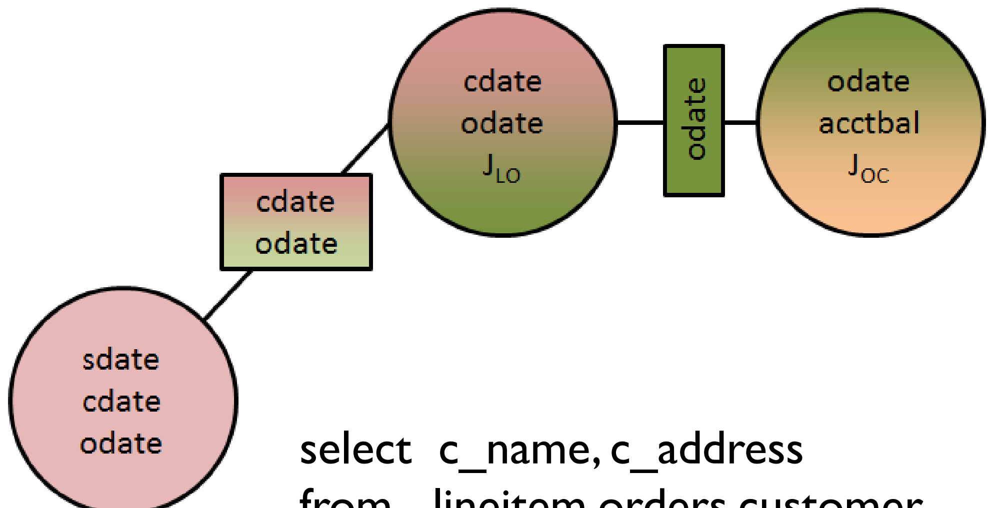
Selectivity Est. in Fixed BN

- Transform BN into a junction tree



Selectivity Est. in Fixed BN

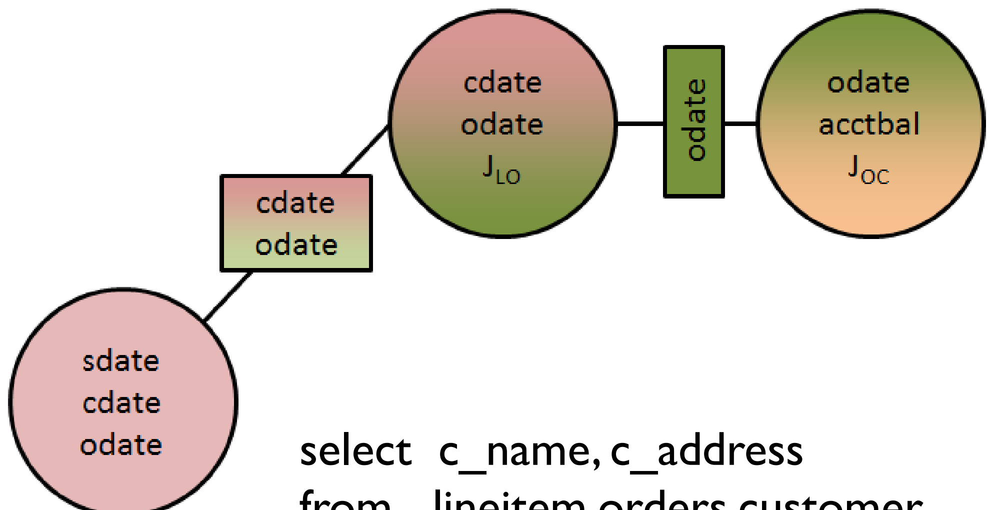
- Extract the subtree related to the query



select c_name, c_address
from lineitem,orders,customer
where l_orderkey=o_orderkey and
o_custkey=c_custkey and
l_sdate<="25/7/2011" and
c_acctbal<=200000

Selectivity Est. in Fixed BN

- Tree algorithms for joint distribution
  - Sum-product
  - DP



select c_name, c_address
from lineitem,orders,customer
where l_orderkey=o_orderkey and
o_custkey=c_custkey and
l_sdate<=“25/7/2011” and
c_acctbal<=200000

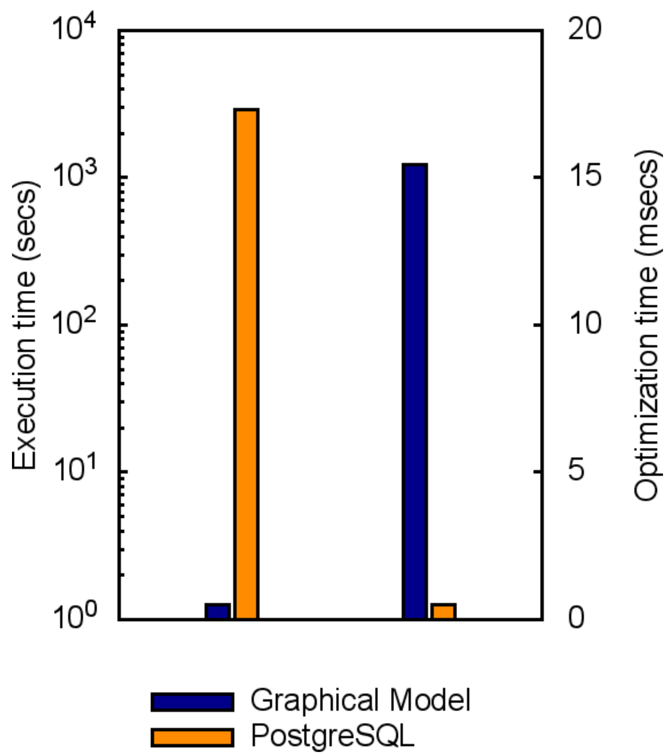
Experiments

- Implementation
  - Model construction outside DBMS, use queries to get distributions
  - Stores junction tree as tables in catalog
  - Replace selectivity estimation procedure

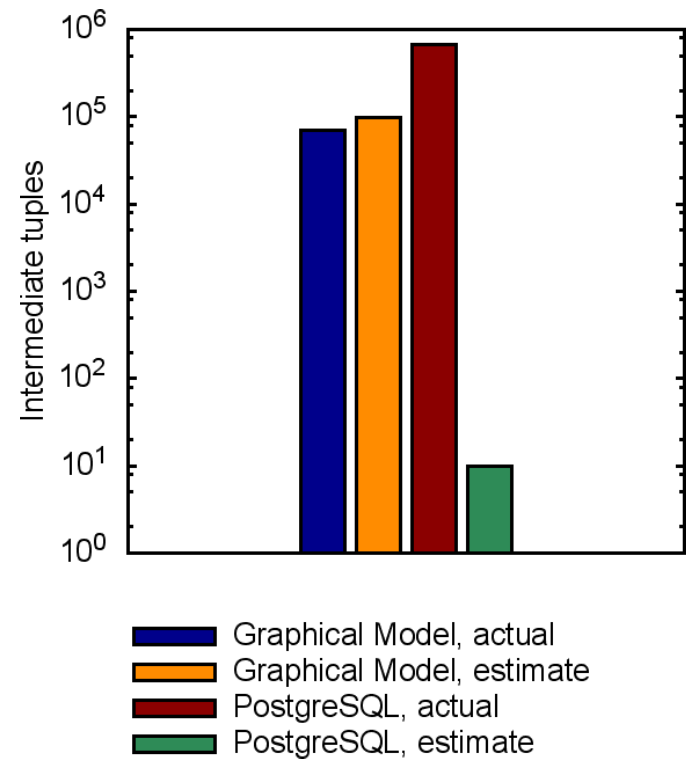
- Compare with PostgresSQL

Efficiency and Accuracy

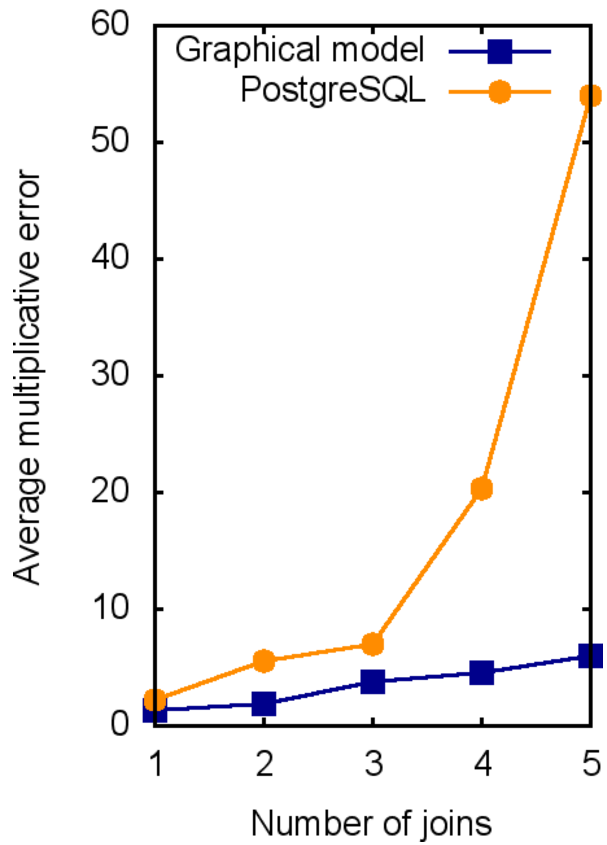
Execution & optimization times

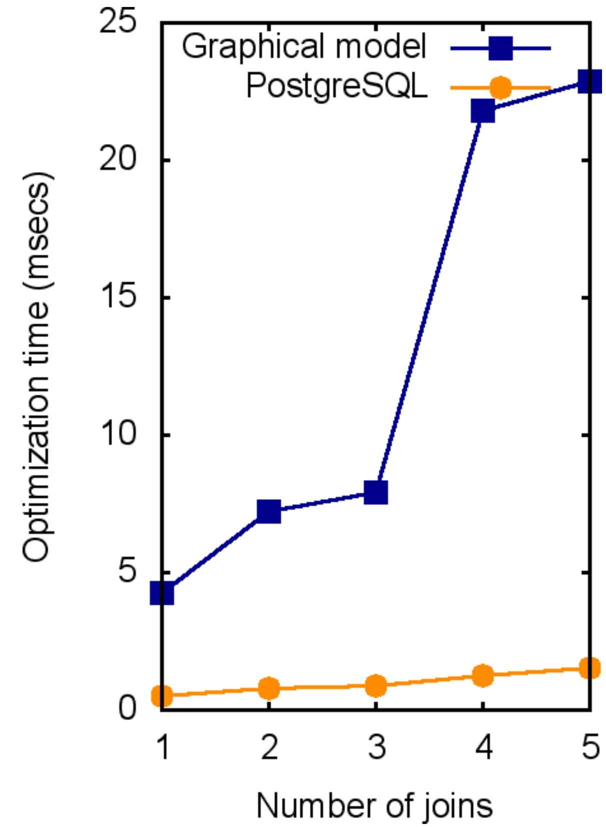


Cost of plans (intermediate tuples)



Scalability





Error: *max(real,estimate) / min(real,estimate)*

Conclusion

Thank You!

- Inaccurate selectivity estimation leads to bad plan
- Fixed structure BN is a good tradeoff between efficiency and accuracy (?)