Congressional samples

Juho Lamminmäki

Based on Congressional Samples for Approximate Answering of Group-By Queries (2000) by Swarup Acharyua et al.

Data Sampling

- Trying to obtain a maximally representative subset of the original data to reduce computation time or required storage.
- 100% accurate data is not always needed for analytics.
- The sample should work well with different kinds of queries.

Data Sampling

- The problem with plain uniform sampling
- Congressional samples
- Querying the sampled data
- Drawbacks of the approach
- Conclusion

Aggregation queries


```
SELECT sex, muncipality, party, AVG(age)
FROM poll
WHERE election_year = 2017
GROUP BY sex, muncipality, party
```

Uniform sampling

- Given a sample size *X* and the size of the original data *D*, pick *X* random rows with an equal probability.
- However, if some groups are very small, only a few rows are picked from those groups.
- Accuracy becomes an issue with very small samples.

The basic idea behind the solution

- A larger proportion of the original group has to be sampled if the group is small.
- Fewer rows can be sampled from the larger groups since the accuracy does not suffer as much.
- Uniform sampling is important because it works the best if the sample is later queried using predicates.

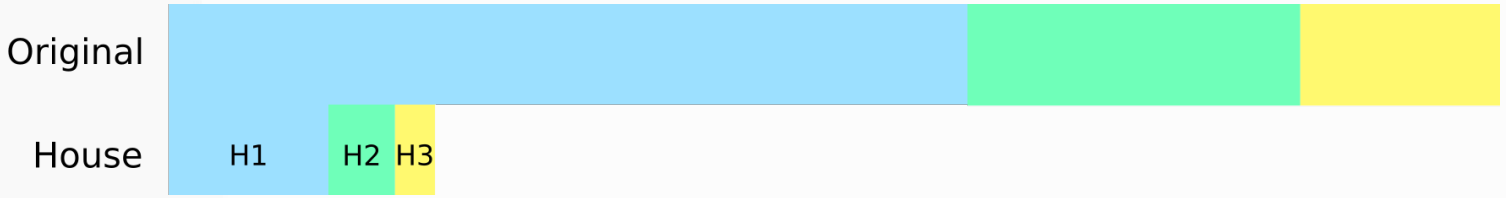
Congressional samples

- House
- Senate
- Basic Congress
- Congress

House

- Uniform sampling over the whole data.

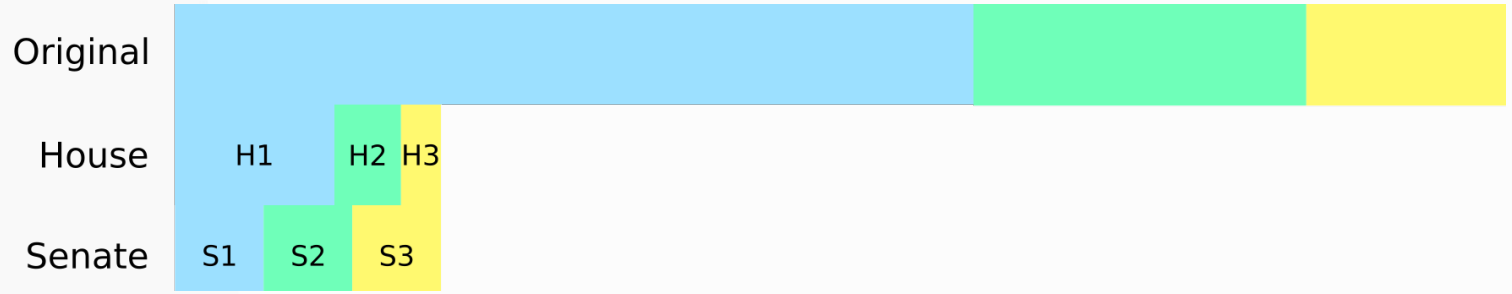
House



Senate

- Given *m* groups and a sample size *X*, take a sample of X/m rows from each group, i.e. the total sample size is divided equally between all groups.
- May use too few samples from the larger groups.

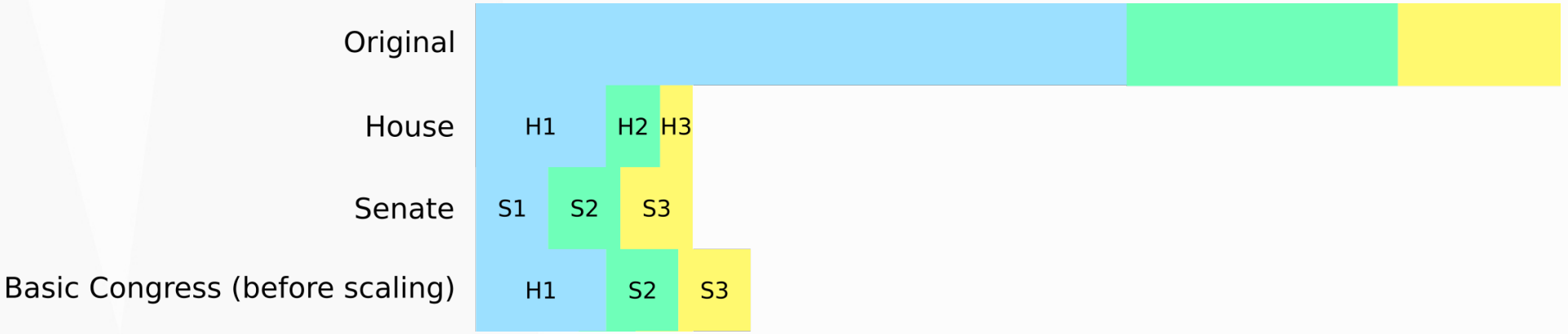
House and Senate



Basic Congress

- A combination of House and Senate
- For each group *g*, the sample size is max*(Hg, Sg)* where *Hg* and *Sg* are the expected sample sizes of group *g* in House and Senate sampling methods respectively.

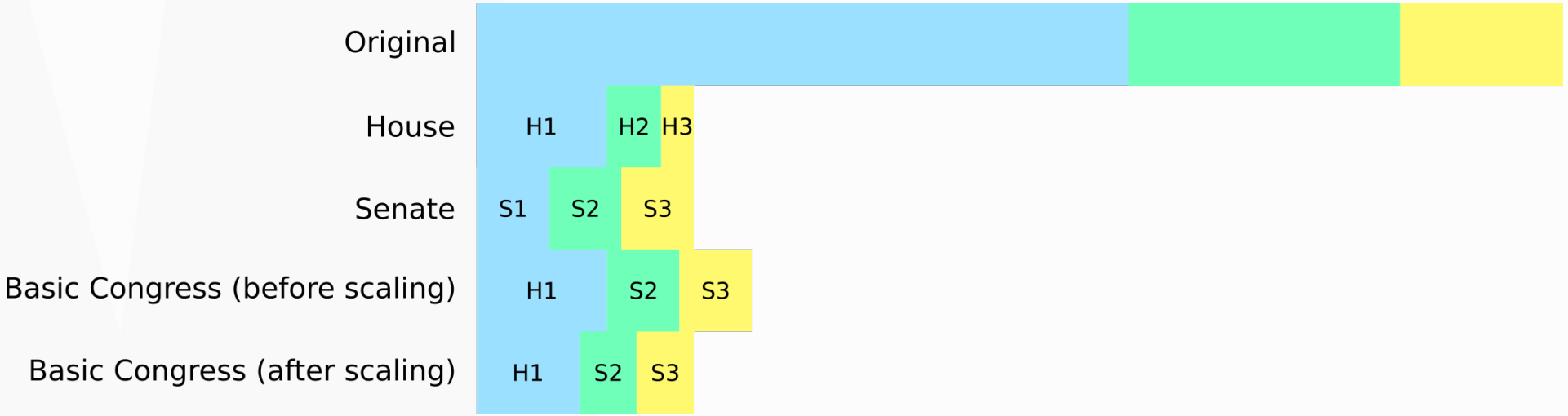
House, Senate and Basic Congress



Basic Congress

- Produces a total sample size ≥ X, so the sample sizes of each group have to be scaled with a constant so that the total sample size becomes X.

House, Senate and Basic Congress



Not perfect

Basic Congress

- Let *A* and *B* be some grouping attributes that group the data into four groups i.e. **GROUP-BY A, B**

A	*B*	avg(*C*)	
a1	b1	...	Group (a1, b1)
a1	b2	...	Group (a1, b2)
a1	b3	...	Group (a1, b3)
a2	b3	...	Group (a2, b3)

Basic Congress

Grouping attributes: A

(a1)	(a2)
75%	25%

Grouping attributes: A, B

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
30%	30%	15%	25%

Basic Congress

Grouping attributes: A

(a1)	(a2)
75%	25%

(a1)	(a2)
60%	40%

As a percentage of the total sample size

Grouping attributes: A, B

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
30%	30%	15%	25%

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
27%	27%	23%	23%

As a percentage of the total sample size

Basic Congress

Grouping attributes: A

(a1)	(a2)
75%	25%

→

(a1)	(a2)
60%	40%

Grouping attributes: A, B

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
30%	30%	15%	25%

→

(a1)	(a2)
77%	23%

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
27%	27%	23%	23%

Basic Congress

Grouping attributes: A

(a1)	(a2)
75%	25%

Optimal

(a1)	(a2)
60%	40%

Grouping attributes: A, B

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
30%	30%	15%	25%

Not optimal

(a1)	(a2)
77%	23%

(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
27%	27%	23%	23%

Optimal

Congress

- A solution to the problem i.e. it works better than Basic Congress with subsets of the original grouping attributes.
- An extension of the basic congress

Congress

- All subsets of the grouping attributes are ∅, {A}, {B} and {A, B}.
- First, calculate the amount of groups created by each subset.

Subset	Groups	Total #
∅	The whole data	1
{A}	(a1), (a2)	2
{B}	(b1), (b2), (b3)	3
{A, B}	(a1, b1), (a1, b2), (a1, b3), (a2, b3)	4

Congress

- Then, calculate the expected sample size for each group using senate sampling.
- If X is the total sample size, then each group has a sample size of X/(number of groups).

Subset	Groups	Total #	Sample size of a single group
∅	The whole data	1	X/1
{A}	(a1), (a2)	2	X/2
{B}	(b1), (b2), (b3)	3	X/3
{A, B}	(a1, b1), (a1, b2), (a1, b3), (a2, b3)	4	X/4

Congress

- So the expected sample size as a percentage of the total sample size X for each group (a1, b1), (a1, b2), (a1, b3), (a2, b3) becomes

	(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
∅	30%	30%	15%	25%
{A}	20%	20%	10%	50%
{B}	25%	25%	18.75%	31.25%
{A, B}	25%	25%	25%	25%

Congress

- The empty set does not group at all, so taking a senate sample with no grouping attributes is the same as taking a House (uniform) sample.

	(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
∅	30%	30%	15%	25%
{A}	20%	20%	10%	50%
{B}	25%	25%	18.75%	31.25%
{A, B}	25%	25%	25%	25%

Congress

- Taking the maximum sample size from either ∅ or {A, B} and scaling is the same as Basic Congress

	(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
∅	30%	30%	15%	25%
{A, B}	25%	25%	25%	25%
MAX	30%	30%	25%	25%

Congress

- Adding the other subsets makes the Basic Congress into Congress.

	(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
∅	30%	30%	15%	25%
{A}	20%	20%	10%	50%
{B}	25%	25%	18.75%	31.25%
{A, B}	25%	25%	25%	25%
MAX	30%	30%	25%	50%

Congress

- This ensures that the sample works reasonably well with any subset of the original grouping attributes.

	(a1, b1)	(a1, b2)	(a1, b3)	(a2, b3)
MAX	30%	30%	25%	50%
SCALED	22.22%	22.22%	18.52%	37.04%

(a1)	(a2)
62.96%	37.04%

(b1)	(b2)	(b3)
22.22%	22.22%	55.56%

Querying sampled data

- Averages, medians etc. work fine without modifications.
- Sums, counts etc. require modification.

Querying sampled data

- **SELECT sum(value) * original_size/sample_size**
  - Works only for uniform samples since original_size/sample_size is not the correct “scale factor” for all groups in non-uniform (biased) samples.
- Storing the scale factor for each row
  - Very high maintenance overhead.
- Storing the scale factor for each group
  - Most likely the best solution

Querying sampled data

```
SELECT v.A, v.B, v.C, sum(v.value) * s.scale_factor
FROM values v
JOIN scale_factors s USING(A, B, C)
GROUP BY v.A, v.B, v.C
```

- Can be optimized further, but this is the basic idea.
- The scale factors have to be constantly maintained, but the overhead is not very high.

Drawbacks

- For some data, uniform sampling over the whole data, which is much easier to implement and maintain, may be good enough.
- Such data might be something where not many grouping attributes are needed and/or there exists no small groups

Drawbacks

- Senate sampling (used in Congress and Basic Congress too) might try to sample more rows than there are in the original data.
- The original paper simply states that handling these scenarios is not straightforward and leaves it at that.

Drawbacks

- Aggregate attributes with a very high variance or outliers with extreme values do not behave well when uniformly sampled.
- e.g. avg(-3, 0, 3 ,1, 1, 100000) = 16667, but avg(-3, 0, 3 ,1) = 0.5

Drawbacks

- In these cases, implementing a solution that buckets the values into ranges [v1, vn] =[v1, v2] ∪ ... ∪ [v[n-1], vn] and takes a representative sample from each bucket will yield better results (Error-bounded Sampling for Analytics on Big Sparse Data, Yin Yan et al., 2014).
- This kind of a solution is more accurate in general, but it is less flexible with e.g. query predicates and the aggregate attributes must be known beforehand.

Conclusion

- Data sampling is useful when saving resources or time trumps accuracy.
- Small groups a problem with uniform sampling.
- Congress sampling fixes the problem with small groups, but does not handle situations where the aggregate attribute has some extreme values.
- Sampling makes querying more complex.

Phew, it’s finally over!

In case you missed it, my name is Juho Lamminmäki