Putting the “Learning” in Learning-Augmented Algorithms for Frequency Estimation

Elbert Du*, **Franklyn Wang***, and Michael Mitzenmacher

Harvard University

Learning-Augmented Algorithms

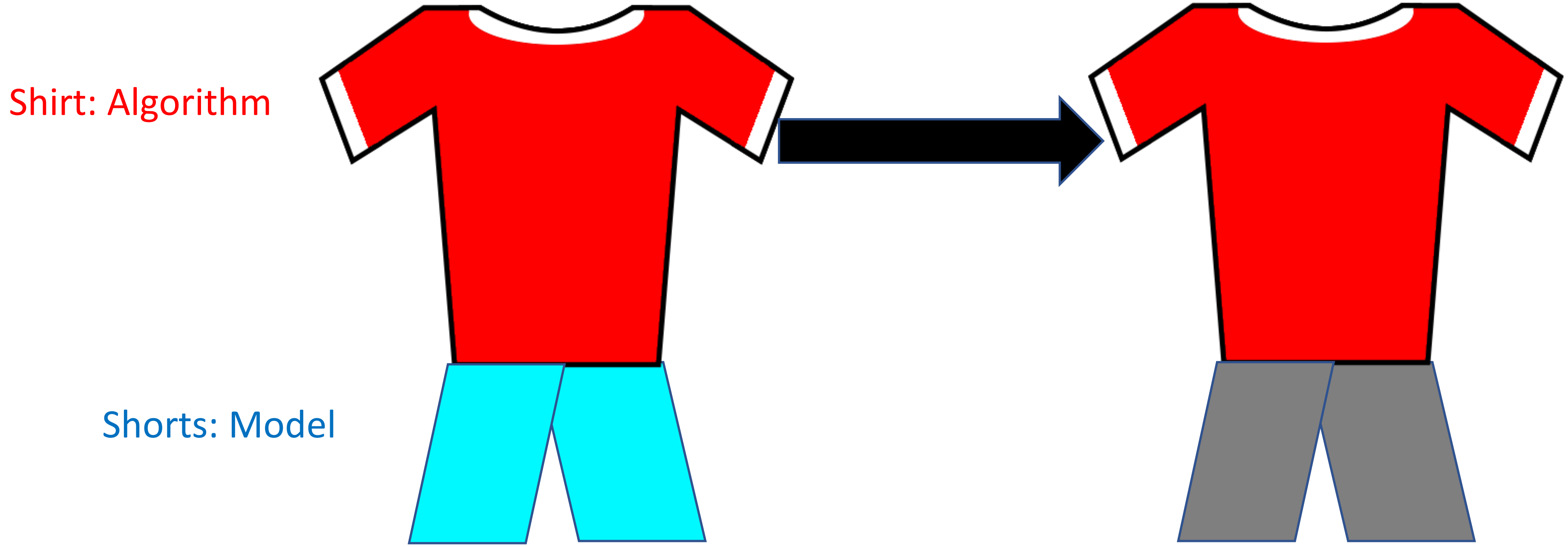
- Learning-Augmented Algorithms are fusions of machine learning models and classical algorithms.

- In the present paper, we address the Learned Count-Min sketch [Hsu et. al 2019]

Contribution, in a nutshell

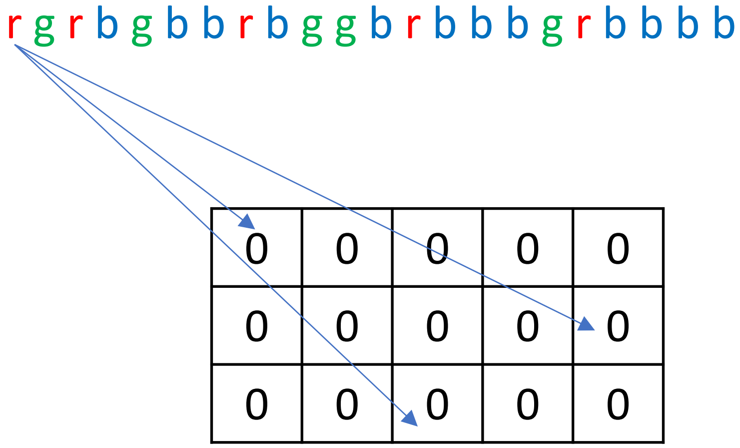
One can think of learning-augmented algorithms as an outfit.

Given the shirt, we choose the shorts.



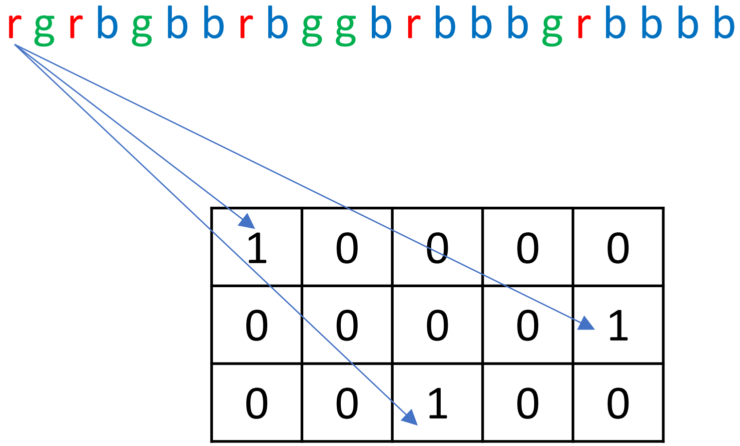
Learned Count—Min Sketch

The Count-Min sketch is an algorithm which estimates frequencies in data streams with small space. It first hashes each element several times and increments a table of counters.



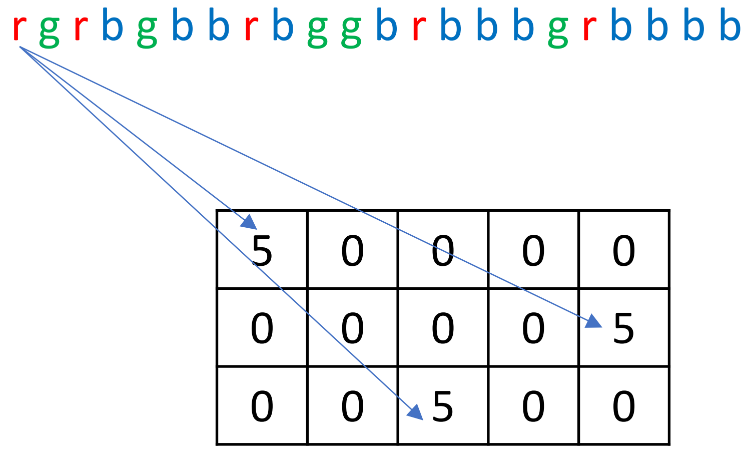
Learned Count—Min Sketch

The Count-Min sketch is an algorithm which estimates frequencies in data streams with small space. It first hashes each element several times and increments a table of counters.



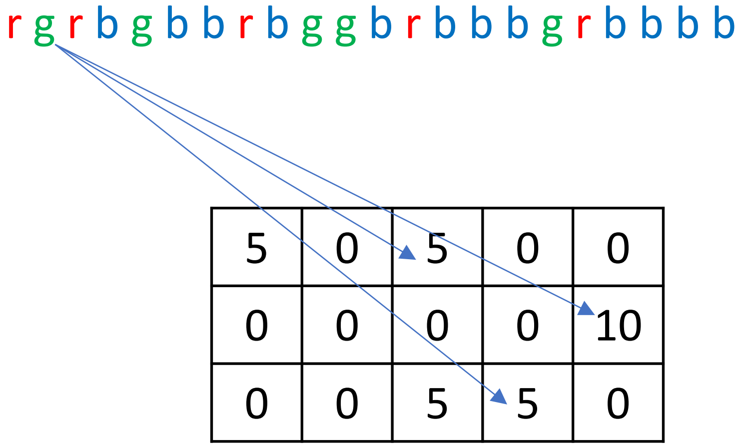
Learned Count—Min Sketch

The Count-Min sketch is an algorithm which estimates frequencies in data streams with small space. It first hashes each element several times and increments a table of counters.



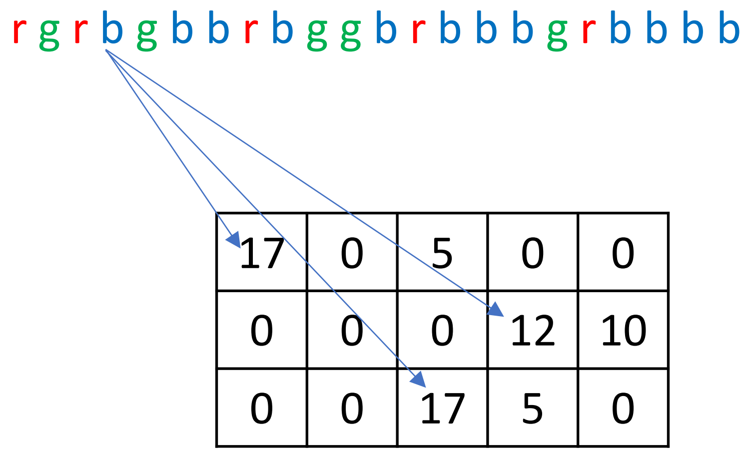
Learned Count—Min Sketch

The Count-Min sketch is an algorithm which estimates frequencies in data streams with small space. It first hashes each element several times and increments a table of counters.



Learned Count—Min Sketch

The Count-Min sketch is an algorithm which estimates frequencies in data streams with small space. It first hashes each element several times and increments a table of counters.



Learned Count—Min Sketch

Finally, we estimate the frequency of an element by taking the *minimum* of all counters it corresponds to.

r g r b g b b r b g g b r b b b g r b b b b

r: 10

17	0	5	0	0
0	0	0	12	**10**
0	0	**17**	5	0

Learned Count—Min Sketch

Finally, we estimate the frequency of an element by taking the *minimum* of all counters it corresponds to.

r g r b g b b r b g g b r b b b g r b b b b

17	0	**5**	0	0
0	0	0	12	**10**
0	0	17	**5**	0

r: 10
g: 5

Learned Count—Min Sketch

Finally, we estimate the frequency of an element by taking the *minimum* of all counters it corresponds to.

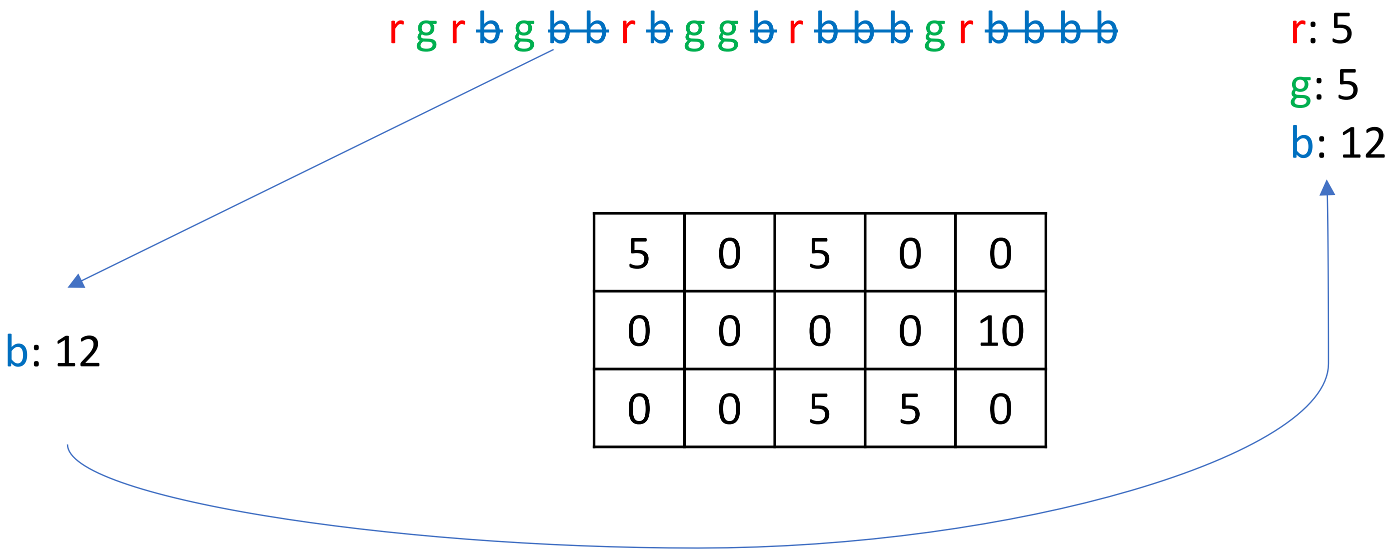
r g r b g b b r b g g b r b b b g r b b b b

17	0	5	0	0
0	0	0	**12**	10
0	0	**17**	5	0

r: 10
g: 5
b: 12

Learned Count—Min Sketch

The learned count-min sketch is the same, except we *screen* some of the keys which are predicted to be heavy and track their frequencies individually.



5	0	5	0	0
0	0	0	0	10
0	0	5	5	0

Optimizing the Sketch

Note that intuitively, screening more frequent keys is more valuable. We call the sum of all screened keys the *coverage*, which we seek to maximize.

r g r ~~b~~ g ~~b b~~ r ~~b~~ g g ~~b~~ r ~~b b b~~ g r ~~b b b b~~

r: 5
g: 5
b: 12

b: 12

5	0	5	0	0
0	0	0	0	10
0	0	5	5	0

How do we optimize the model?

- We seek to optimize the model for downstream performance, **without changing architecture!**
- We will maximize the coverage.
- **Theorem (Informal):** The estimation error is essentially increasing in the complementary coverage.

Optimizing for Coverage

- Hsu et. al uses the squared loss function:

$$\mathbb{E}_{i\sim F^0}\left[\left(\underbrace{g_\theta(i)}_{\textbf{predictor}} - \underbrace{\ln f_i}_{\textbf{log frequency}}\right)^2\right]$$

- Since we are optimizing for coverage, frequent elements are more important than infrequent elements.

$$\mathbb{E}_{i\sim F^1}\left[\left(g_\theta(i) - \ln f_i\right)^2\right]$$

Optimizing for Coverage (cont.)

- Note that the coverage is proportional to

$$\mathbb{E}_{i\sim F^0}[1_{i \text{ screened}} f_i]$$

- Another method: **BatchRank**. We split each batch into sub-batches of size-K and normalize, so we have

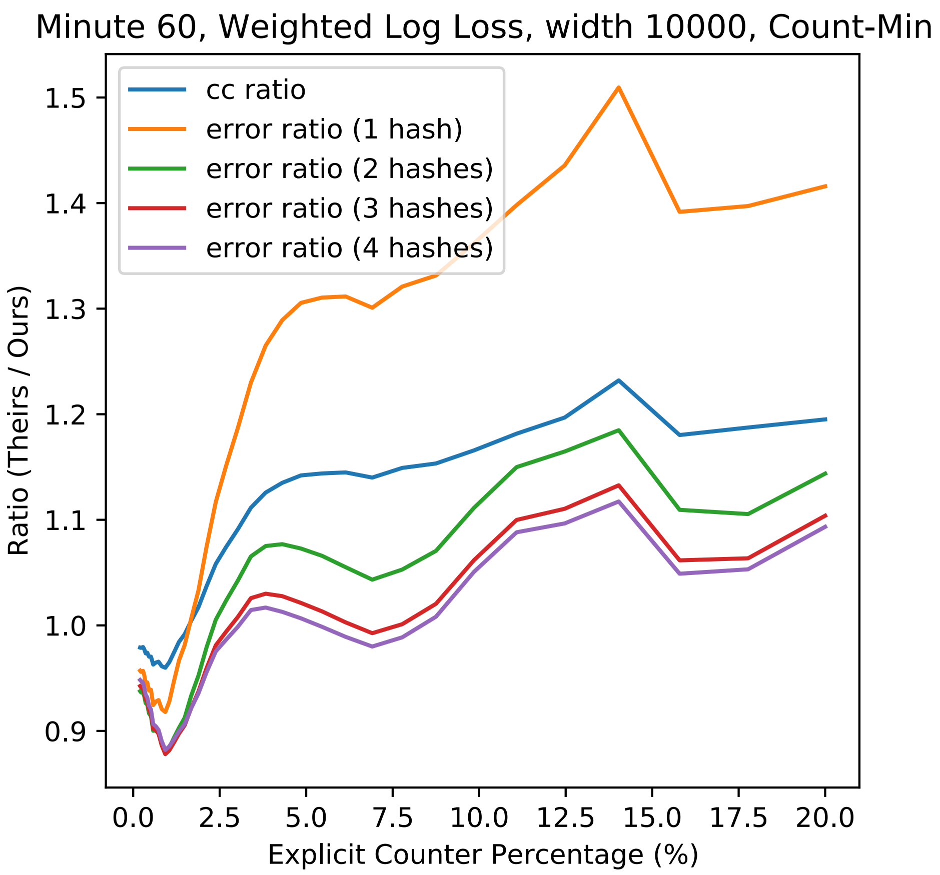
$$g'_\theta(i) = \frac{g_\theta(i) - \mu}{\sigma}.$$

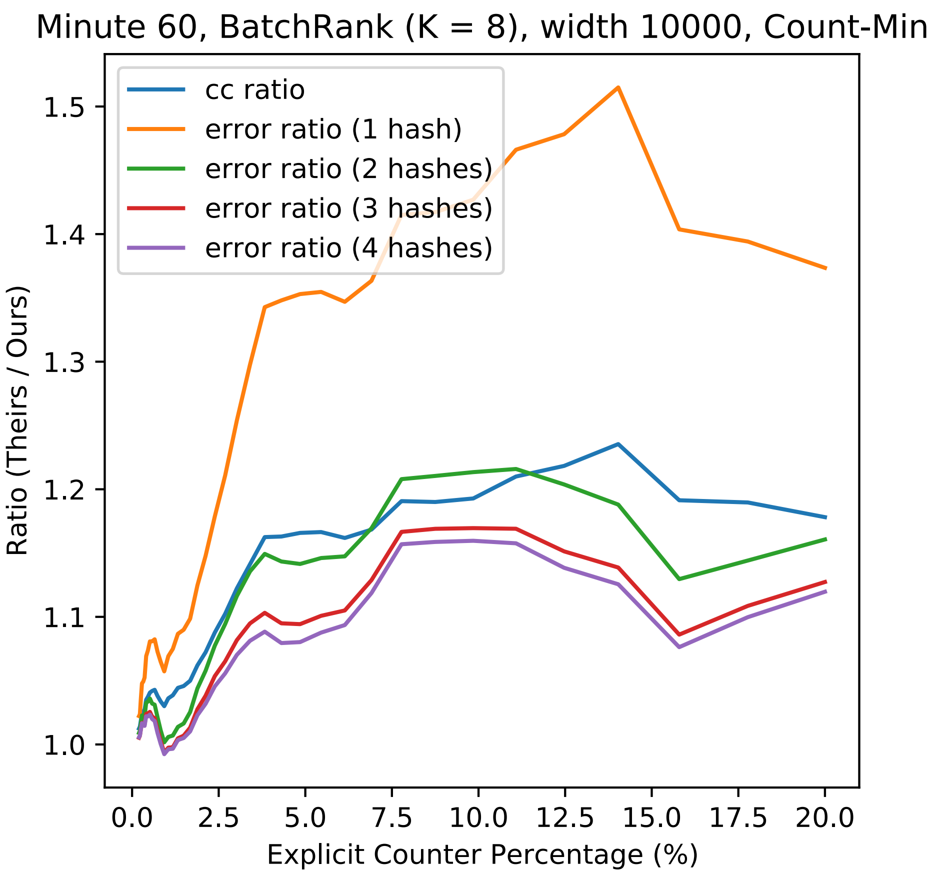
- Then, our final loss function is:

$$\mathbb{E}_{i\sim F^0}[g'_\theta(i) f_i]$$

Results

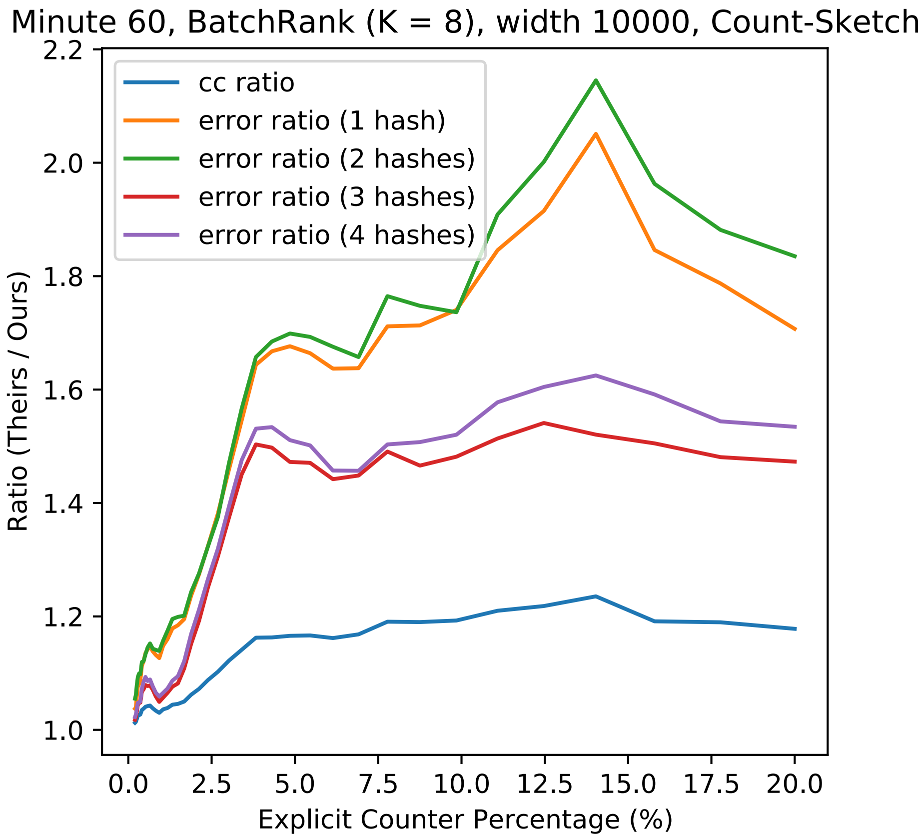
Ratios are theirs / ours



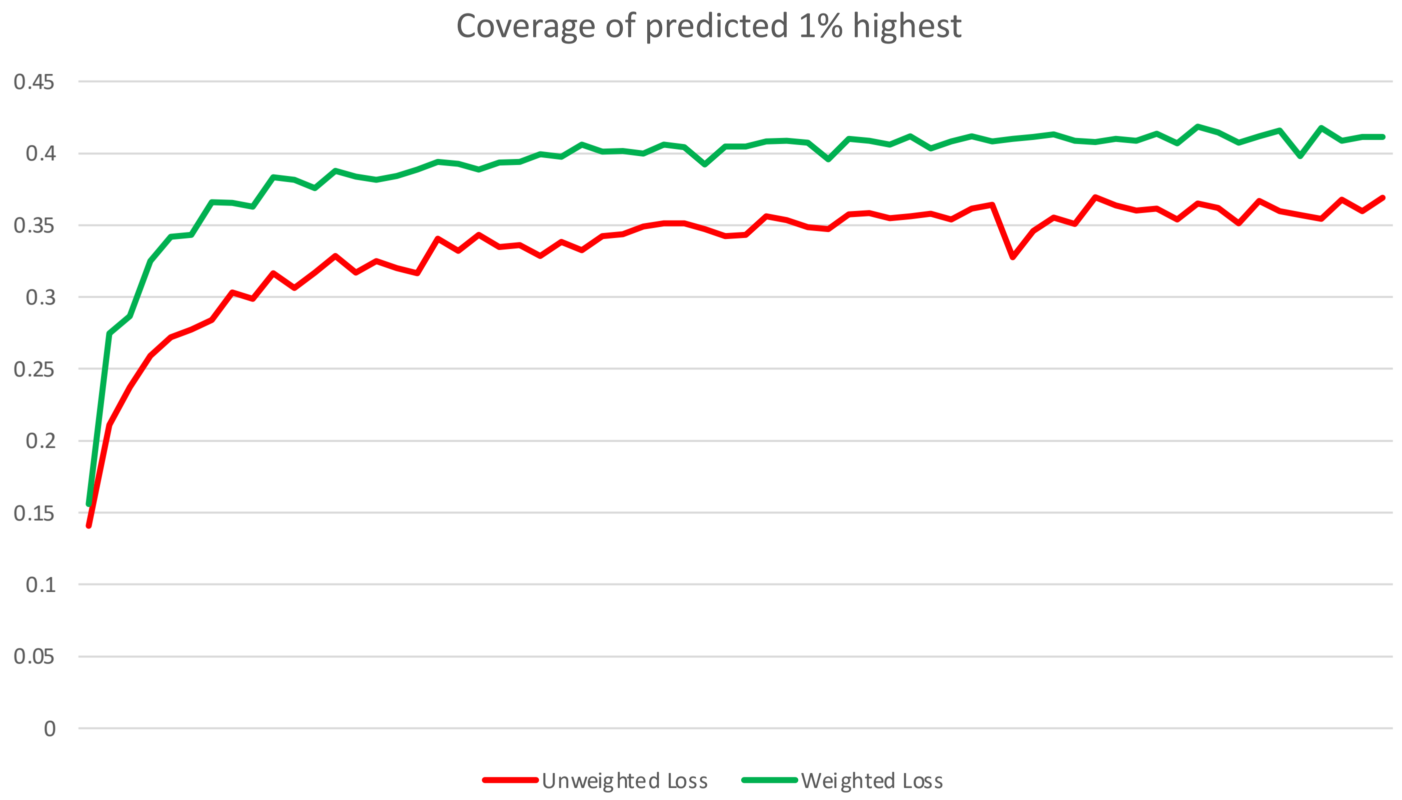


Results

Ratios are theirs / ours

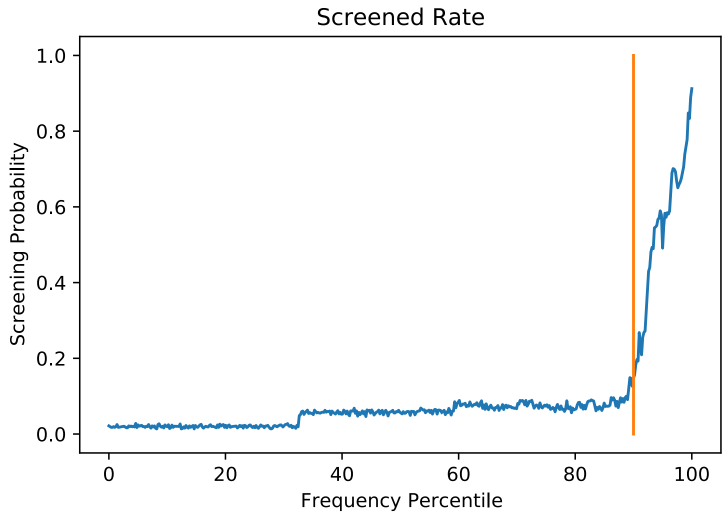


Results


Coverage of predicted 1% highest
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Unweighted Loss
Weighted Loss

A new model for classification errors

- We observe the rates by which keys are screened and note that it increases monotonically with frequency.



- This suggests a heterogenous error model, allowing for stronger guarantees than a uniform error model.

Conclusion

- We would like to invite the machine learning community to try out these tasks—our code can be found here.
- When one can find an accurate proxy for performance (e.g. coverage), training to optimize the proxy can lead to significant improvements.

References

- Hsu, Chen-Yu, et al. "Learning-Based Frequency Estimation Algorithms." International Conference on Learning Representations. 2019.