Author Response: Submission #8113

We thank all reviewers for the constructive feedback. We will incorporate the valuable suggestions from all reviewers (including a title change, as suggested by **R2**). In addition, we briefly address some of the comments below.

Experiments

All reviewers gave feedback or had questions about the experimental set up. We briefly recap the experiments and comment on the purpose and importance of each one, as well as address some reviewer specific comments.

Purpose: The experiments highlight avenues toward bridging the gaps between the following standard theory assumptions and practice: (i) **Asymptotic run-time analysis:** Our PTAS is fantastic in theory but not practical. In practice we show that QR, which is very simple, is also near-optimal (Fig. 1). (ii) **Constant-factor approximation:**[1] In practice, Greedy, QR, and KR all perform extremely well (in fact, they have comparable behavior despite the very different theoretical guarantees) and much better than the baseline of Expectation (Fig. 1). (iii) **Availability of explicit distribution:** Thanks **R3** for bring this up! (See also Lines 292-4.) In practice explicit distributions typically only arise if we fit a model (e.g. Gaussians) to data. A slightly more realistic assumption are historical samples from the same distribution. In our real data experiments, we go one step further: We consider the practical scenario where we observe only one value for each feature vector. Here, we have an *implicit distribution over our uncertainty*. We develop a novel approach that allows us to successfully apply analogs of KR and QR in this setting (Fig. 3). (iv) **Just maximize the objective:** In theory, improving the objective function is always a better outcome. In practice, in particular in the context of the **broader impact** of ML research, it is important to explore the bias introduced by different algorithms. In particular we hypothesized that some algorithms for our problems will be biased by data scarcity, a well-documented bias in practical ML. In our second synthetic experiment each distribution gets a random label: l (less data) or m (more data); based on this label we only see a less or more samples from this distribution. In this experiment we measure the percentage of each population (l vs m) selected by each algorithm, as well as performance (expected largest/second largest value). We see that while the performance is almost identical, the choice of method and quantile (both for KR and QR) has major effects on the percentage of small sample candidates selected.

R2 asks what the random variables are in the real data experiment. The random variables are our implicit uncertainty about the value corresponding to a feature vector. We estimate the quantiles (resp. expectation) of this implicit distribution using quantile (resp. squared loss) regression. For all methods we use neural nets (and hopefully this clears up a confusion of **R4** regarding linear regression being treated differently: it isn't), and as **R2** correctly comments, "linear regression" should instead be "neural net with squared loss"; we have corrected this. **R4** asks about neural net models; we will provide additional information, including the depth, loss function and platform used to train these neural nets, as well as our methodology for training and testing. **R2** asks about the number of simulations in Fig. 3. Each feature vector/tweet in the test data is used only once. An experiment samples 500 tweets and picks k of them (we give figures for multiple values of k), and Fig. 3 is averaged over 8000 experiments (the 4000 number is a typo).

As **R1 and R4** point out, the first part of the synthetic experiments and the Twitter data experiments don't give a clean separation between the algorithms. However, the surprise here is that despite the poor approximation guarantees of QR in theory, in practice it does just as well as the theoretically superior (better approximation guarantee without the MHR assumption, at least for expected maximum) KR algorithm. Furthermore, in terms of simplicity, the QR algorithm has the advantage as it has a *strict* subset of the steps of the KR algorithm. Finally, even though [KR18] introduced this score function (the score equals to the expectation over the top $1/k$ quantile), adapting it to the real data as two consecutive training steps (first quantile loss and then squared loss) is a contribution of this paper. We hope this addresses the comments of **R3 and R4** about the applicability, significance and advantages of the proposed method.

Additional Comments

R2 asks about the optimal $c/\sqrt{k}$ quantile. We do not know about the optimal one, but one can improve the approximation factor for the expected maximum objective by picking the $1/k$ quantile, using a similar analysis. Maximizing the expected maximum is indeed NP-hard; we will add this result.

R2 and R4 ask about the running of the PTAS. The running time is $O(n|V|polylog(k))$, where $|V|$ is the maximum size of the support of a random variable, so "near-linear". This does, of course, take into account all operations, including simplifications, calculating conditional expectations and so on. We will clarify this.

[1] We did not try to optimize the constant factors.