

# cs208 HW 3

Anthony Rentsch

4/2/2019

## Question 1

(a) To prove that this mechanism is  $\epsilon$ -DP, I will show that (i) the percentile trimming transformation is 1-Lipschitz, (ii) that the Laplace noise injection mechanism is  $\epsilon$ -DP, and (iii) that this implies that the entire mechanism  $M(x)$  is  $(1 * \epsilon)$ -DP.

- (i) A mapping  $T$  from dataset to dataset is  $c$ -Lipschitz iff  $\forall x, x' d(T(x), T(x')) \leq c * d(x, x')$ . Here let's consider that  $x$  and  $x'$  only differ on one element. It follows that  $d(x, x') = 1$ .

Now consider the percentile trimming transformation in this mechanism. It again follows that  $d(T(x), T(x')) = 1$  since the maximum number of rows that these two datasets will differ on is 1. Returning the inequality in the definition of a Lipschitz constant, we see that this transformation is 1-Lipschitz.

- (ii) First, we observe that

$$\frac{1}{.9n} \sum_{P_{.05} \leq x \leq P_{0.95}} x_i$$

is simply an estimator for the mean of  $x$  after trimming the bottom and top 5% of the data. For simplicity, replace  $.9n$  with  $n'$  and call this mechanism  $M'$ . Note that the global sensitivity of this query is  $GS_q = D/n'$ . Since the Laplace noise is scaled by  $\frac{GS_q}{\epsilon}$ ,  $M'$  is  $\epsilon$ -DP.

- (iii) In class, we discussed a lemma that states that if  $M$  is  $\epsilon$ -DP and  $T$  is  $c$ -Lipschitz, then  $M \circ T$  is  $(c * \epsilon)$ -DP. Following from (i) and (ii), we then have that  $M = M' \circ T$  is  $(1 * \epsilon)$ -DP.

Below is the implementation of this mechanism.

```
sgn <- function(x) {      # function borrowed from class
  return(ifelse(x < 0, -1, 1))
}

rlap = function(mu=0, b=1, size=1) {      # function borrowed from class
  p <- runif(size) - 0.5
  draws <- mu - b * sgn(p) * log(1 - 2 * abs(p))
  return(draws)
}

trimmedMean <- function(x, d, n, epsilon) {
  scale <- d/(epsilon*0.9*n)
  quants <- quantile(x, c(0.05,0.95))
  x_trimmed <- x[x>quants[1] && x<quants[2]]
  mean_trimmed <- (1/(0.9*epsilon*n))*sum(x_trimmed)
  mean_release <- mean_trimmed + rlap(mu=0,b=scale)
  return(mean_release)
}
```

- (b) Let's first consider the Lipschitz constant of the transformation  $[x]_{P_{0.5}}^{P_{0.95}}$ .

- this must be 2-Lipschitz

(c)

- describe mechanism
- implement with code

(d)

- code

(e)

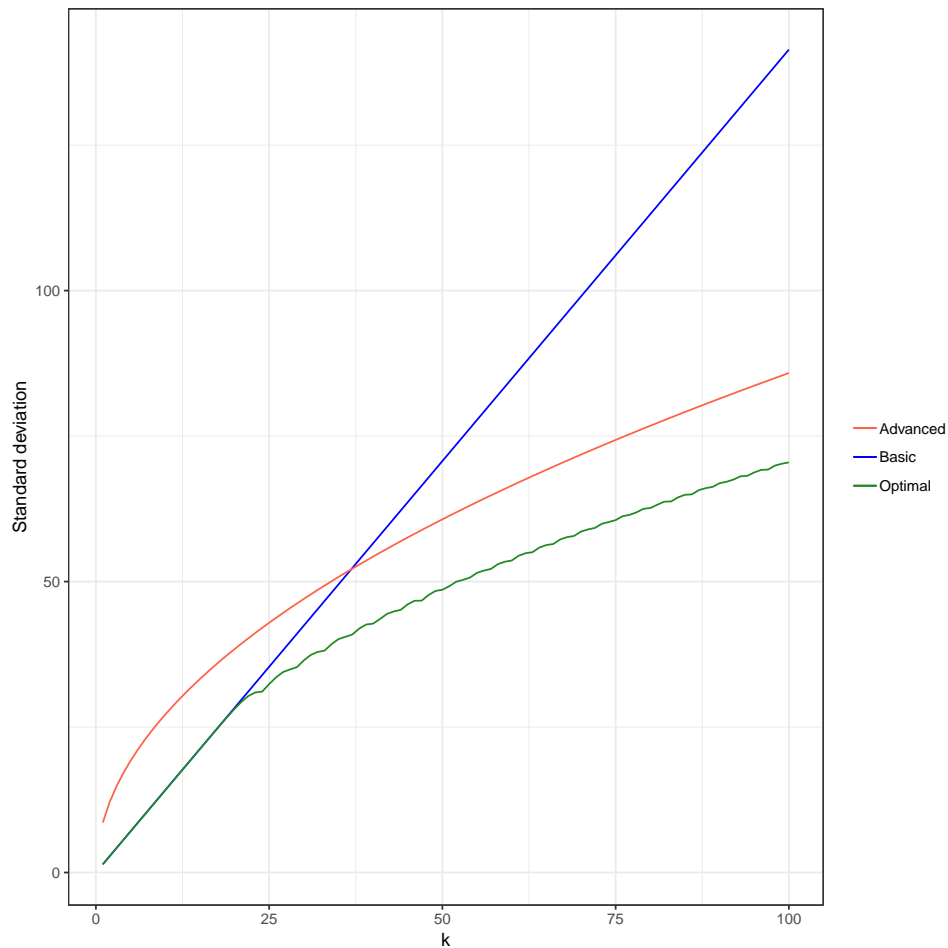
- informal sketch of proof

(f)

- have to re-implement laplace clamped mean from last hw
- code + plots
- description of results/ situation in which this would be a good approach

## Question 2

The “optimal” composition theorem strictly improves upon the standard deviation of the injected noise from the basic composition theorem when  $k \geq 14$  and advanced composition strictly improves upon basic composition when  $k \geq 37$ . Up to  $k = 14$ , basic and “optimal” composition correspond to roughly the same standard deviation. The standard deviation from advanced composition is strictly larger than the standard deviation from “optimal” composition, but the ratio between the two appears to remain constant as  $k \rightarrow \infty$ .



### Question 3

### BONUS

### Appendix

I put the code for all of my analyses here. You can also find it on [Github](#).

### Question 1

### Question 2

### Question 3