Proceedings of Machine Learning Research 58 (2016) 61-74

Optimal Number of Choices in Rating Contexts

Sam Ganzfried SGANZFRI@CIS.FIU.EDU
Florida International University
School of Computing and Information Sciences

Editor: Tatiana V. Guy, Miroslav Kárný, David Rios-Insua, David H. Wolpert

Abstract

In many settings people must give numerical scores to entities from a small discrete set. For instance, rating physical attractiveness from 1–5 on dating sites, or papers from 1–10 for conference reviewing. We study the problem of understanding when using a different number of options is optimal. For concreteness we assume the true underlying scores are integers from 1–100. We consider the case when scores are uniform random and Gaussian. While in theory for this setting it would be optimal to use all 100 options, in practice this is prohibitive, and it is preferable to utilize a smaller number of options due to humans' cognitive limitations. Our results suggest that using a smaller number of options than is typical could be optimal in certain situations. This would have many potential applications, as settings requiring entities to be ranked by humans are ubiquitous.

1. Introduction

Humans rate items or entities in many important settings. For example, users of dating websites and mobile applications rate other users' physical attractiveness, teachers rate scholarly work of students, and reviewers rate the quality of academic conference submissions. In these settings, the users assign a numerical (integral) score to each item from a small discrete set. However, the number of options in this set can vary significantly between applications, and even within different instantiations of the same application. For instance, for rating attractiveness, three popular sites all use a different number of options. On "Hot or Not," users rate the attractiveness of photographs submitted voluntarily by other users on a scale of 1–10 (Figure 1[1]). These scores are aggregated and the average is assigned as the overall "score" for a photograph. On the dating website OkCupid, users rate other users on a scale of 1–5 (if a user rates another user 4 or 5 then the rated user receives a notification)[2] (Figure 2[3]). And on the mobile application Tinder users "swipe right" (green heart) or "swipe left" (red X) to express interest in other users (two users are allowed to message each other if they mutually swipe right), which is essentially equivalent to using a binary $\{1, 2\}$ scale (Figure 3[4]). Education is another important application area requiring human ratings. For the 2016 International Joint Conference on Artificial Intelligence, reviewers assigned a "Summary Rating" score from -5–5 (equivalent to 1–10) for each submitted paper (Figure 4).[5]). The papers are then

1. http://blog.mrmeyer.com/2007/are-you-hot-or-not/
2. The likelihood of receiving an initial message is actually much more highly correlated with the variance—and particularly the number of "5" ratings—than with the average rating (Fry, 2015).
3. http://blog.okcupid.com/index.php/the-mathematics-of-beauty/
4. https://tctechcrunch2011.files.wordpress.com/2015/11/tinder-two.jpg
5. https://easychair.org/conferences/?conf=ijcai16

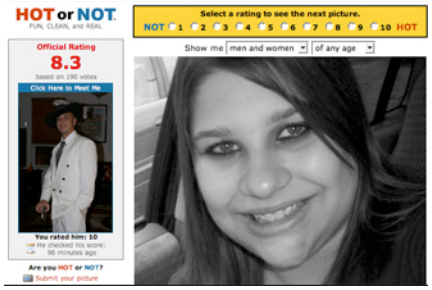

Figure 1: Hot or Not users rate attractiveness 1–10.

Figure 2: OkCupid users rate attractiveness 1–5.

Figure 3: Tinder users rate attractiveness 1–2.

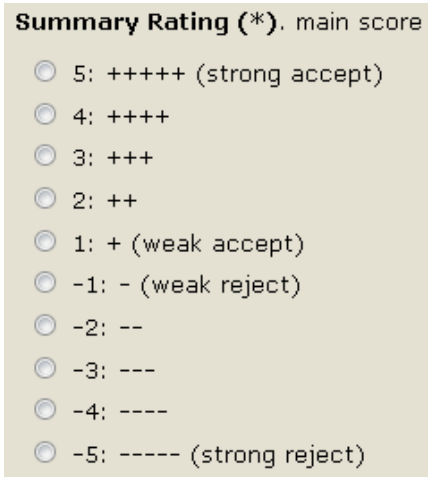

Figure 4: IJCAI reviewers rate papers -5–5.

discussed and scores aggregated to produce an acceptance or rejection decision based largely on the average of the scores.

Despite the importance and ubiquity of the problem, there has been little fundamental research done on the problem of determining the optimal number of options to allow in such settings. We study a model in which users have a underlying integral ground truth score for each item in $\{1,\ldots,n\}$ and are required to submit an integral rating in $\{1,\ldots,k\}$, for $k \ll n$. (For ease of presentation we use the equivalent formulation $\{0,\ldots,n-1\}$, $\{0,\ldots,k-1\}$.) We use two generative models for the ground truth scores: a uniform random model in which the fraction of scores for each value from 0 to $n-1$ is chosen uniformly at random (by choosing a random value for each and then normalizing), and a model where scores are chosen according to a Gaussian distribution with a given mean and variance. We then compute a "compressed" score distribution by mapping each full score s from $\{0,\ldots,n-1\}$ to $\{0,\ldots,k-1\}$ by applying

$$s \leftarrow \left\lfloor \frac{s}{\binom{n}{k}} \right\rfloor . \tag{1}$$

We then compute the average "compressed" score a_k, and compute its error e_k according to

$$e_k = \left| a_f - \frac{n-1}{k-1} \cdot a_k \right|, \tag{2}$$

where a_f is the ground truth average score. The goal is to pick $\text{argmin}_k e_k$. While there are many possible generative models and cost functions to use, these seemed like the most natural ones to start with. We leave study of alternative choices for future work.

We derive a closed-form expression for e_k that depends on only a small number (k) of parameters of the underlying distribution for an arbitrary distribution.[6] This allows us to exactly characterize the performance of using each number of choices. In computational simulations we repeatedly compute e_k and compare the average values. We focus on $n = 100$ and $k = 2, 3, 4, 5, 10$, which we believe are the most natural and interesting choices for initial study.

One could argue that this model is somewhat "trivial" in the sense that it would be optimal to set $k = n$ to permit all the possible scores, as this would result in the "compressed" scores agreeing exactly with the full scores. However, there are several reasons that would lead us to prefer to select $k \ll n$ in practice (as all of the examples previously described have done), thus making this "thought experiment" worthwhile. It is much easier for a human to assign a score from a small set than from a large set, particularly when rating many items under time constraints. We could have included an additional term into the cost function e_k that explicitly penalizes larger values of k, which would have a significant effect on the optimal value of k (providing a favoritism for smaller values). However the selection of this function would be somewhat arbitrary and would make the model more complex, and we leave this for future study. Given that we do not include such a penalty term, one may expect that increasing k will always decrease e_k in our setting. While the simulations show a clear negative relationship between k and e_k, we show that smaller values of k can actually lead to smaller e_k surprisingly often. These smaller values would receive further preference with a penalty term.

The most closely related theoretical work studies the impact of using finely grained numerical grades (e.g., 100, 99, 98) vs. coarse letter grades (e.g., A, B, C) (Dubey and Geanakoplos, 2010). They conclude that if students care primarily about their rank relative to the other students, they are often best motivated to work by assigning them to coarse categories (letter grades) than by the exact numerical exam scores. In a specific setting of "disparate" student abilities they show that the optimal absolute grading scheme is always coarse. Their model is game-theoretic; each player (student) selects an effort level, seeking to optimize a utility function that depends on both the relative score and effort level. Their setting is quite different from ours in many ways. For one, they study a setting where it is assumed that the underlying "ground truth" score is known, yet may be disguised for strategic reasons. In our setting the goal is to approximate the ground truth score as closely as possible.

While we are not aware of prior theoretical study of our problem, there have been experimental studies on the optimal number of options on a "Likert scale" (Matell and Jacoby, 1971; Wildt and Mazis, 1978; Cox III, 1980; Friedman et al., 1981). The general conclusion is that "the optimal number of scale categories is content specific and a function of the conditions of measurement" (Garland,

6. For theoretical simplicity we theoretically study a continuous version where scores are chosen according to a distribution over $(0, n)$ (though the simulations are for the discrete version) and the compressed scores are over $\{0, \ldots, k-1\}$. Continuous approximations for large discrete spaces have been studied in other settings; for instance, they have led to simplified analysis and insight in poker games with continuous distributions of private information (Ankenman and Chen, 2006).

1991). There has been study of whether including a "mid-point" option (i.e., the middle choice from an odd number) is beneficial. One experiment demonstrated that the use of the mid-point category decreases as the number of choices increases: 20% of respondents choose the mid-point for 3 and 5 options while only 7% did for $7, 9, \ldots, 19$ options (Matell and Jacoby, 1972). They conclude that it is preferable to either not include a mid-point at all or use a large number of options. Subsequent experiments demonstrated that eliminating a mid-point option can reduce social desirability bias which results from respondents' desires to please the interviewer or not give what they perceive to be a socially unacceptable answer (Garland, 1991).

2. Theoretical characterization

Suppose scores are given by continuous pdf f (with cdf F) on $(0, 100)$, and we wish to compress them to two options, $\{0, 1\}$. Scores below 50 are mapped to 0 and scores above 50 to 1.

The average of the full distribution is

$$a_f = E[X] = \int_{x=0}^{100} xf(x)dx.$$

The average of the compressed version is

$$\begin{aligned} a_2 &= \int_{x=0}^{50} 0f(x)dx + \int_{x=50}^{100} 1f(x)dx = \int_{x=50}^{100} f(x)dx \\ &= F(100) - F(50) = 1 - F(50). \end{aligned}$$

So $e_2 = |a_f - 100(1 - F(50))| = |E[X] - 100 + 100F(50)|$.

For three options,

$$\begin{aligned} a_3 &= \int_{x=0}^{100/3} 0f(x)dx + \int_{x=100/3}^{200/3} 1f(x)dx + \int_{x=200/3}^{100} 2f(x)dx \\ &= F(200/3) - F(100/3) + 2(1 - F(200/3)) \\ &= 2 - F(100/3) - F(200/3) \\ e_3 &= |a_f - 50(2 - F(100/3) - F(200/3))| \\ &= |E[X] - 100 + 50F(100/3) + 50F(200/3)| \end{aligned}$$

In general for n total and k compressed options,

$$\begin{aligned} a_k &= \sum_{i=0}^{k-1} \int_{x=\frac{ni}{k}}^{\frac{n(i+1)}{k}} if(x)dx \\ &= \sum_{i=0}^{k-1} \left[i \left(F\left(\frac{n(i+1)}{k}\right) - F\left(\frac{ni}{k}\right) \right) \right] \\ &= (k-1)F(n) - \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \\ &= (k-1) - \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \end{aligned}$$

$$\begin{aligned} e_k &= \left| a_f - \frac{n}{k-1}\left((k-1) - \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right)\right)\right| \\ &= \left| E[X] - n + \frac{n}{k-1}\sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right)\right| \end{aligned} \tag{3}$$

Equation 3 allows us to characterize the relative performance of choices of k for a given distribution f. For each k the characterization requires only knowing k statistics of f (the $k-1$ values of $F\left(\frac{ni}{k}\right)$ plus $E[X]$). In practice these could likely be closely approximated from historical data for small values of k.

As an example we see that $e_2 < e_3$ iff

$$|E[X] - 100 + 100F(50)| < |E[X] - 100 + 50F(100/3) + 50F(200/3)|$$

Consider a full distribution that has half its mass right around 30 and half its mass right around 60 (Figure 5). Then $a_f = E[X] = 0.5 \cdot 30 + 0.5 \cdot 60 = 45$. If we use $k = 2$, then the mass at 30 will be mapped down to 0 (since $30 < 50$) and the mass at 60 will be mapped up to 1 (since $60 > 50$) (Figure 6). So $a_2 = 0.5 \cdot 0 + 0.5 \cdot 1 = 0.5$. Using normalization of $\frac{n}{k} = 100$, $e_2 = |45 - 100(0.5)| = |45 - 50| = 5$. If we use $k = 3$, then the mass at 30 will also be mapped down to 0 (since $0 < \frac{100}{3}$); but the mass at 60 will be mapped to 1 (not the maximum possible value of 2 in this case), since $\frac{100}{3} < 60 < \frac{200}{3}$ (Figure 7). So again $a_3 = 0.5 \cdot 0 + 0.5 \cdot 1 = 0.5$, but now using normalization of $\frac{n}{k} = 50$ we have $e_3 = |45 - 50(0.5)| = |45 - 25| = 20$. So, surprisingly, in this example allowing more ranking choices actually significantly increases error.

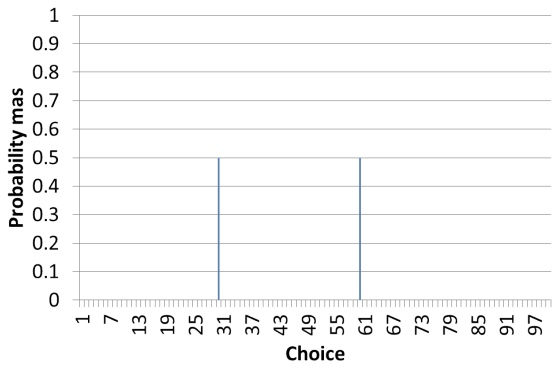

Figure 5: Example distribution for which compressing with $k = 2$ produces lower error than $k = 3$.

3. Rounding compression

An alternative model we could have considered is to use rounding to produce the compressed scores as opposed to using the floor function from Equation 1. For instance, for the case $n = 100, k = 2$, instead of dividing s by 50 and taking the floor, we could partition the points according to whether

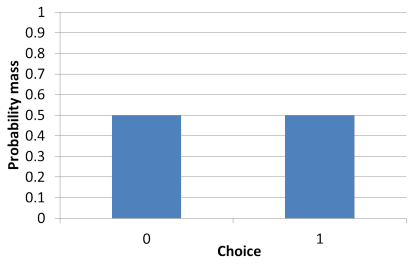

Figure 6: Compressed distribution using $k = 2$.

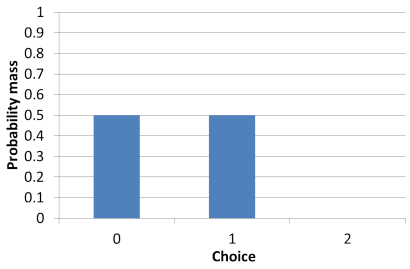

Figure 7: Compressed distribution using $k = 3$.

they are closer to $t_1 = 25$ or $t_2 = 75$. In the example above, the mass at 30 would be mapped to t_1 and the mass at 60 would be mapped to t_2. This would produce a compressed average score of

$$a_2 = \frac{1}{2} \cdot 25 + \frac{1}{2} \cdot 75 = 50.$$

No normalization would be necessary, and this would produce error of

$$e_2 = |a_f - a_2| = |45 - 50| = 5,$$

as the floor approach did as well. Similarly, for $k = 3$ the region midpoints will be $q_1 = \frac{100}{6}$, $q_2 = 50$, $q_3 = \frac{500}{6}$. The mass at 30 will be mapped to $q_1 = \frac{100}{6}$ and the mass at 60 will be mapped to $q_2 = 50$. This produces a compressed average score of

$$a_3 = \frac{1}{2} \cdot \frac{100}{6} + \frac{1}{2} \cdot 50 = \frac{100}{3}.$$

This produces an error of

$$e_3 = |a_f - a_3| = \left|45 - \frac{100}{3}\right| = \frac{35}{3} = 11.67$$

Although the error for $k = 3$ is smaller than for the floor, it is still significantly larger than the error for $k = 2$, and using two options still outperforms using three for the example in this new model.

In general, this approach would create k "midpoints" $\{m_i^k\}$:

$$m_i^k = \frac{n(2i-1)}{2k}$$

For $k = 2$ we have

$$\begin{aligned} a_2 &= \int_{x=0}^{50} 25 + \int_{x=50}^{100} 75 = 25F(50) + 75(1 - F(50)) = 75 - 50F(50) \\ e_2 &= |a_f - (75 - 50F(50))| = |E[X] - 75 + 50F(50)| \end{aligned}$$

One might wonder whether the floor approach would ever outperform the rounding approach (in the example above the rounding approach produced lower error $k = 3$ and the same error for $k = 2$). As a simple example to see that it can, consider the distribution with all mass on 0. The

floor approach would produce $a_2 = 0$ giving an error of 0, while the rounding approach would produce $a_2 = 25$ giving an error of 25. Thus, the superiority of the approach is dependent on the distribution. We explore this further in the experiments.

For three options,

$$\begin{aligned}
a_3 &= \int_{x=0}^{100/3} 100/6 f(x)dx + \int_{x=100/3}^{200/3} 50 f(x)dx + \int_{x=200/3}^{100} 500/6 f(x)dx \\
&= 100/6F(100/3) + 50(F(200/3) - F(100/3)) + 500/6(1 - F(200/3) \\
&= 100/6F(100/3) + 50F(200/3) - 50F(100/3) + 500/6 - 500/6F(200/3) \\
&= 500/6 - 100/3F(100/3) - 100/3F(200/3) \\
e_3 &= |a_f - (500/6 - 100/3F(100/3) - 100/3F(200/3))| \\
&= |E[X] - 500/6 + 100/3F(100/3) + 100/3F(200/3)|
\end{aligned}$$

For general n and k,

$$\begin{aligned}
a_k &= \sum_{i=0}^{k-1} \int_{x=\frac{ni}{k}}^{\frac{n(i+1)}{k}} m_{i+1}^k f(x)dx \\
&= \sum_{i=0}^{k-1} \int_{x=\frac{ni}{k}}^{\frac{n(i+1)}{k}} \frac{n(2i+1)}{2k} f(x)dx \\
&= \sum_{i=0}^{k-1} \left[\frac{n(2i+1)}{2k} \left(F\left(\frac{n(i+1)}{k}\right) - F\left(\frac{ni}{k}\right) \right) \right] \\
&= \frac{n}{k} \sum_{i=0}^{k-1} \left[i \left(F\left(\frac{n(i+1)}{k}\right) - F\left(\frac{ni}{k}\right) \right) \right] + \frac{n}{2k} \sum_{i=0}^{k-1} \left[\left(F\left(\frac{n(i+1)}{k}\right) - F\left(\frac{ni}{k}\right) \right) \right] \\
&= \frac{n}{k} \left[(k-1) - \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \right] + \frac{n}{2k} \\
&= \frac{n(2k-1)}{2k} - \frac{n}{k} \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \\
e_k &= \left| a_f - \left[\frac{n(2k-1)}{2k} - \frac{n}{k} \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \right] \right| = \left| E[X] - \frac{n(2k-1)}{2k} + \frac{n}{k} \sum_{i=1}^{k-1} F\left(\frac{ni}{k}\right) \right| \quad (4)
\end{aligned}$$

Like for the floor model e_k requires only knowing k statistics of f. The rounding model has an advantage over the floor model that there is no need to convert scores between different scales and perform normalization. One drawback is that it requires knowing n (the expression for m_i^k is dependent on n), while the floor model does not. In our experiments we assume $n = 100$, but in practice it may not be clear what the agents' ground truth granularity is and may be easier to just deal with scores from 1 to k. Furthermore, it may seem unnatural to essentially ask people to rate items as "$\frac{100}{6}, 50, \frac{200}{6}$" rather than "$1, 2, 3$" (though the conversion between the score and m_i^k could be done behind the scenes essentially circumventing the potential practical complication). We note that one could generalize both the floor and rounding model by using a score of $s(n,k)_i$

for the i'th region. For the floor setting we set $s(n,k)_i = i$, and for the rounding setting we set $s(n,k)_i = m_i^k = \frac{n(2i+1)}{2k}$.

4. Computational simulations

The above analysis leads to the immediate question of whether the example for which $e_2 < e_3$ was just a fluke or whether using fewer choices can actually reduce error under reasonable assumptions on the generative model. We study this question using simulations which we believe are the two most natural models. While we have studied the continuous setting where the full set of options is continuous over $(0, n)$ and the compressed set of options is the discrete space $\{0, \ldots, k-1\}$, we will now consider the perhaps more realistic setting where the full set is the discrete set $\{0, \ldots, n-1\}$, and the compressed set is $\{0, \ldots, k-1\}$ (though it should be noted that the two settings are likely extremely similar qualitatively).

The first generative model we consider is a uniform model in which the values of the pmf p_f for each of the n possible values are chosen independently and uniformly at random. Formally, we construct a histogram of n scores for p_f according to Algorithm 1. We then compress the full scores to a compressed distribution p_k by applying Algorithm 2.

The second generative model is a Gaussian model in which the values are generated according to a normal distribution with specified mean μ and standard deviation σ. This model also takes as a parameter a number of samples s to use for generating the scores. The procedure is given by Algorithm 3. As for the uniform setting, Algorithm 2 is then used to compress the scores.

Algorithm 1 Procedure for generating full scores in uniform model

Inputs: Number of scores n

```
scoreSum ← 0
for i = 0 : n do
    r ← random(0,1)
    scores[i] ← r
    scoreSum = scoreSum +r
for i = 0 : n do
    scores[i] = scores[i] / scoreSum
```

Algorithm 2 Procedure for compressing scores

Inputs: scores[], number of total scores n, desired number of compressed scores k

$Z(n,k) \leftarrow \frac{n}{k}$ ▷ Normalization

for $i = 0 : n$ **do**

    scoresCompressed $\left[\left\lfloor \frac{i}{Z(n,k)} \right\rfloor\right]$ += scores[i]

For our simulations we used $n = 100$, and considered $k = 2, 3, 4, 5, 10$, which are popular and natural values. For the Gaussian model we used $s = 1000$, $\mu = 50$, $\sigma = \frac{50}{3}$. For each set of simulations we computed the errors for all considered values of k for $m = 100,000$ "items" (each corresponding to a different distribution generated according to the specified model). The main quantities we are interested in computing are the number of times that each value of k produces the lowest error over the m items, and the average value of the errors over all items for each k value.

Algorithm 3 Procedure for generating full scores in Gaussian model

Inputs: Number of scores n, number of samples s, mean μ, standard deviation σ

```
for i = 0 : s do
    r ← randomGaussian(μ, σ)
    if r < 0 then
        r = 0
    else if r > n − 1 then
        r ← n − 1
    ++scores[round(r)]
for i = 0 : n do
    scores[i] = scores[i] / s
```

In the first set of experiments, we compared performance between using $k = 2, 3, 4, 5, 10$ to see for how many of the trials each value of k produced the minimal error. The results are in Table 1. Not surprisingly, we see that the number of victories increases monotonically with the value of k, while the average error decreased monotonically (recall that we would have zero error if we set $k = 100$). However, what is perhaps surprising is that using a smaller number of compressed scores produced the optimal error in a far from negligible number of the trials. For the uniform model, using 10 scores minimized error only around 53% of the time, while using 5 scores minimized error 17% of the time, and even using 2 scores minimized it 5.6% of the time. The results were similar for the Gaussian model, though a bit more in favor of larger values of k, which is what we would expect because the Gaussian model is less likely to generate "fluke" distributions that could favor the smaller values.

	2	3	4	5	10
Uniform # victories	5564	9265	14870	16974	53327
Uniform average error	1.32	0.86	0.53	0.41	0.19
Gaussian # victories	3025	7336	14435	17800	57404
Gaussian average error	1.14	0.59	0.30	0.22	0.10

Table 1: Number of times each value of k in $\{2,3,4,5,10\}$ produces minimal error and average error values, over 100,000 items generated according to both models.

We next explored the number of victories between just $k = 2$ and $k = 3$, with results in Table 2. Again we observed that using a larger value of k generally reduces error, as expected. However, we find it extremely surprising that using $k = 2$ produces a lower error 37% of the time. As before, the larger k value performs relatively better in the Gaussian model. We also looked at results for the most extreme comparison, $k = 2$ vs. $k = 10$. These results are provided in Table 3. Using 2 scores outperformed 10 8.3% of the time in the uniform setting, which was larger than we expected. In Figures 8–10, we present a distribution for which $k = 2$ particularly outperformed $k = 10$.

We next repeated the extreme $k = 2$ vs. 10 comparison, but we imposed a restriction that the $k = 10$ option could not give a score below 3 or above 6. (If it selected a score below 3 then we set it to 3, and if above 6 we set it to 6). These result are given in Table 4. For some settings, for instance the paper reviewing setting, extreme scores are very uncommon, and we strongly suspect that the

	2	3
Uniform number of victories	36805	63195
Uniform average error	1.31	0.86
Gaussian number of victories	30454	69546
Gaussian average error	1.13	0.58

Table 2: Number of times each value of k in $\{2,3\}$ produces minimal error and average error values, over 100,000 items generated according to both generative models.

	2	10
Uniform number of victories	8253	91747
Uniform average error	1.32	0.19
Gaussian number of victories	4369	95631
Gaussian average error	1.13	0.10

Table 3: Number of times each value of k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both generative models.

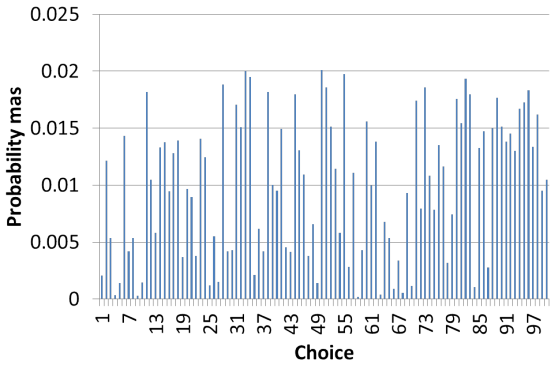

Figure 8: Example distribution for which compressing with $k = 2$ produces significantly lower error than $k = 10$. The full distribution has mean 54.188, while the $k = 2$ compression has mean 0.548 (54.253 after normalization) and the $k = 10$ compression has mean 5.009 (55.009 after normalization). The normalized errors between the means were 0.906 for $k = 10$ and 0.048 for $k = 2$, yielding a difference of 0.859 in favor of $k = 2$.

vast majority of scores are in this middle range. Some possible explanations are that reviewers who give extreme scores may be required to put in additional work to justify their scores, and are more likely to be involved in arguments with the other reviewers (or with the authors in the rebuttal). Reviewers could also experience higher regret or embarrassment for being "wrong" and possibly off-base in the review by missing an important nuance. In this setting using $k = 2$ outperforms $k = 10$ nearly $\frac{1}{3}$ of the time in the uniform model.

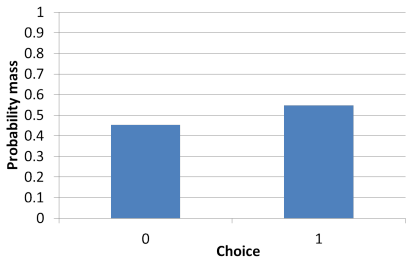

Figure 9: Compressed distribution for $k = 2$.

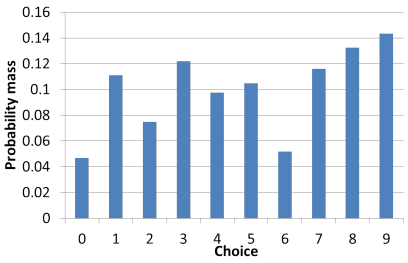

Figure 10: Compressed distribution for $k = 10$.

	2	10
Uniform number of victories	32250	67750
Uniform average error	1.31	0.74
Gaussian number of victories	10859	89141
Gaussian average error	1.13	0.20

Table 4: Number of times each k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both models. For $k = 10$, we only permitted scores between 3 and 6 (inclusive). If a score was below 3 we set it to be 3, and above 6 to 6.

We also considered the situation where we restricted the $k = 10$ scores to fall between 3 and 7 (as opposed to 3 and 6). Note that the possible scores range from 0–9, so this restriction is asymmetric in that the lowest three possible scores are eliminated while only the highest two are. This is motivated by the intuition that raters may be less inclined to give extremely low scores which may hurt the feelings of an author (for the case of paper reviewing). In this setting, which is seemingly quite similar to the 3–6 setting, $k = 2$ produced lower error 93% of the time in the uniform model!

	2	10
Uniform number of victories	93226	6774
Uniform average error	1.31	0.74
Gaussian number of victories	54459	45541
Gaussian average error	1.13	1.09

Table 5: Number of times each k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both models. For $k = 10$, we only permitted scores between 3 and 7 (inclusive). If a score was below 3 we set it to be 3, and above 7 to 7.

We next repeated these experiments for the rounding compression function. There are several interesting observations from Table 6. In this setting, $k = 3$ is the clear choice, performing best in both generative models (by a very significant margin for the Gaussian model). The smaller values of k perform significantly better in the rounding model than in the floor model (as indicated by lower average errors) while the larger values perform significantly worse, and their errors seem to

approach 0.5 for both models. Taking both compression functions into account, the optimal overall approach would still be to use the floor approach with $k = 10$, which produced the smallest average errors of 0.19 and 0.1 in the two models, while using $k = 3$ in the rounding setting produced errors of 0.47 and 0.24. The 2 vs. 3 experiments produced very similar results for the two compressions (Table 7). The 2 vs. 10 results were quite different, with 2 performing better almost 40% of the time with rounding, vs. less than 10% with the floor function (Table 8). In the 2 vs. 10 truncated 3–6 experiments 2 performed relatively better in the rounding setting for both models (Table 9), and for the 2 vs. 10 truncated 3–7 experiments $k = 2$ performed better nearly all the time (Table 10).

	2	3	4	5	10
Uniform # victories	15766	33175	21386	19995	9678
Uniform average error	0.78	0.47	0.55	0.52	0.50
Gaussian # victories	13262	64870	10331	9689	1848
Gaussian average error	0.67	0.24	0.50	0.50	0.50

Table 6: Number of times each value of k in $\{2,3,4,5,10\}$ produces minimal error and average error values, over 100,000 items generated from both models with rounding compression.

	2	3
Uniform number of victories	33585	66415
Uniform average error	0.78	0.47
Gaussian number of victories	18307	81693
Gaussian average error	0.67	0.24

Table 7: Number of times each value of k in $\{2,3\}$ produces minimal error and average error values, over 100,000 items generated according to both models with rounding compression.

	2	10
Uniform number of victories	37225	62775
Uniform average error	0.78	0.50
Gaussian number of victories	37897	62103
Gaussian average error	0.67	0.50

Table 8: Number of times each value of k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both models with rounding compression.

5. Conclusion

Settings in which humans must rate items or entities from a small discrete set of options are ubiquitous. We have singled out several important applications—rating attractiveness for dating websites and mobile applications, assigning grades to students, and reviewing academic papers for confer-

	2	10
Uniform number of victories	55676	44324
Uniform average error	0.79	0.89
Gaussian number of victories	24128	75872
Gaussian average error	0.67	0.34

Table 9: Number of times each value of k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both models with rounding compression. For $k = 10$, we only permitted scores between 3 and 6 (inclusive). If a score was below 3 we set it to be 3, and above 6 to 6.

	2	10
Uniform number of victories	99586	414
Uniform average error	0.78	3.50
Gaussian number of victories	95692	4308
Gaussian average error	0.67	1.45

Table 10: Number of times each value of k in $\{2,10\}$ produces minimal error and average error values, over 100,000 items generated according to both generative models with rounding compression. For $k = 10$, we only permitted scores between 3 and 7 (inclusive). If a score was below 3 we set it to be 3, and above 7 to 7.

ences. The number of available options can vary considerably, even within different instantiations of the same application. For instance, we saw that three popular sites for the attractiveness rating problem use completely different systems: Hot or Not uses a 1–10 system, OkCupid uses 1–5 "star" system, and Tinder uses a binary 1–2 "swipe" system.

Despite the ubiquity and importance of the problem of selecting the optimal number of rating choices, we have not seen it studied theoretically previously. Our goal is to select k to minimize the average (normalized) error between the compressed average score and the ground truth average. We studied two natural models for generating the scores. The first is a uniform model where the scores are selected independently and uniformly at random, and the second is a Gaussian model where they are selected according to a more structured procedure that gives more preference for the options near the center.

We provided a closed-form solution for continuous distributions with arbitrary cdf. This allows us to characterize the relative performance of choices of k for a given distribution. We saw that, counterintuitively, using a smaller value of k can actually produce a smaller error for some distributions (even though we know that as k approaches n the error approaches 0). We presented a specific example distribution f for which using $k = 2$ outperforms $k = 3$.

We performed numerous computational simulations, comparing the performance between different values of k for different generative models and metrics. The main metric we used was the absolute number of times for which values of k produced the minimal error. We also considered the average error over all simulated items. Not surprisingly, we observe that performance generally improves monotonically with increased k, and more so for the Gaussian model than the uniform.

However, we observe that small k values can be optimal a non-negligible amount of the time, which is perhaps counterintuitive. In fact, using $k = 2$ outperformed $k = 3, 4, 5$, and 10 on 5.6% of the trials in the uniform setting. Just comparing 2 vs. 3, $k = 2$ performed better around 37% of the time. Using $k = 2$ outperformed $k = 10$ 8.3% of the time, and significantly more as we imposed some very natural restrictions on the $k = 10$ setting that are motivated by intuitive phenomena. When we restricted the $k = 10$ to only assign values between 3 and 7 (inclusive), using $k = 2$ actually produced lower error 93% of the time! This could correspond to a setting where raters are ashamed to assign extreme scores (particularly extreme low scores).

We compared two different natural compression rules—one based on the floor function and one based on rounding—and weighed the pros and cons of each. For smaller values of k the rounding approach leads to significantly lower error than the floor approach, with $k = 3$ being the clear optimal choice, while for larger values of k rounding leads to significantly higher error.

One avenue for future study would be to extend our theoretical characterization analysis in order to get a better understanding of the specific distributions for which different values of k are optimal, as opposed to our experimental results which are in aggregate over many distributions. Specific application domains will have distributions with different properties, and improved understanding will allow us to determine which k value is optimal for the types of distributions we expect to encounter for a given domain. This improved theoretical understanding can be coupled with exploring data on specific applications of interest.

References

Jerrod Ankenman and Bill Chen. *The Mathematics of Poker*. ConJelCo LLC, 2006.

Eli P. Cox III. The optimal number of response alternatives for a scale: A review. *Journal of Marketing Research*, 17:407–442, 1980.

Pradeep Dubey and John Geanakoplos. Grading exams: 100, 99, 98, ... or A, B, C? *Games and Economic Behavior*, 69:72–94, May 2010.

Hershey H. Friedman, Yonah Wilamowsky, and Linda W. Friedman. A comparison of balanced and unbalanced rating scales. *The Mid-Atlantic Journal of Business*, 19(2):1–7, 1981.

Hannah Fry. *The Mathematics of Love*. TED Books, 2015.

Ron Garland. The mid-point on a rating scale: is it desirable? *Marketing Bulletin*, 2:66–70, 1991. Research Note 3.

Michael S. Matell and Jacob Jacoby. Is there an optimal number of alternatives for likert scale items? Study 1: reliability and validity. *Educational and psychological measurements*, 31:657–674, 1971.

Michael S. Matell and Jacob Jacoby. Is there an optimal number of alternatives for likert scale items? Effects of testing time and scale properties. *Journal of Applied Psychology*, 56(6):506–509, 1972.

Albert R. Wildt and Michael B. Mazis. Determinants of scale response: label versus position. *Journal of Marketing Research*, 15:261–267, 1978.