Mathematische Modellierung
Tip Optimization

Doris Koinegg
Daniel Kraft

June 20th, 2009

Chapter 1

The Minimal Restaurant

Let's look at a "minimal restaurant" consisting only of a single table as a first system to analyze. Assume it can serve only a single group of customers at a time, with a maximum of n people in this group (the size of the single table). So if there's already a group dining, no others can be accepted; if the table is free however, the waiters may assign it to a new group of customers to their liking.

The opening hours of a single business day (the unit we look at) are defined via an interval of two times, $t_{begin} < t_{end}$.

In order to simplify the rules even further, we assume that time is discretizised in intervals of length Δt. This means that the time to serve a group is always a multiple of Δt and also that new groups can only be accepted at times $t_i = t_{begin} + i\Delta t$, if they have entered the restaurant in the preceding time interval, because the waiters only look from time to time at the entrance to pick up new groups that may be looking for a table. This, however, is not that unrealistic, given that a real group of people seeking a table in a restaurant is not going to find one and sit down in "zero" time but this also takes some finite duration.

At each of these times t_i whenever the table is currently free, the waiters can choose one of the groups that have entered during the last interval and assign them the table. The other groups, however, have to be sent away. If the table is occupied, everyone entering has to be sent away immediatelly, too. So it is not allowed to have customers wait for the currently served group to finish, and it is also not allowed to place a second group at the single table (even if there would be enough seats!).

All the waiters know about groups when choosing the one to pick is the number of people m it consists of, so there's no nothing to suggest if a particular group "looks like" giving high or low tips. While more people of course are more likely to give a large amount of money, they usually also take longer to serve. Let g_m be the expected amount of tips for a group of size m and $d_m\Delta t$ be the length of time it will occupy the table.

As a consequence, if there are more than one groups of the same size waiting forthe table at any time, the waiters can simply discard all but one because they are identically from their point of view. Because of this, we'll further on without loss of generality assume that at most one group of size m is waiting for the table (but there may be none).

Based on these very restrictive assumptions, we can work out a quite "general" model (and algorithm to pick the best strategy) using a bottom-up approach.

1.1 Finding the best strategy

For each of the grid points t_i in time, we define $X(t_i)$ to be the expected winning on tips for the waiters from time t_i onwards until t_{end}, under the condition that the table is free at t_i (and may be assigned to a new group right away), assuming the best possible strategy picking groups of customers.

If this function X is known, it is easy to decide for each time t_i, if there are multiple groups waiting, which one to pick: If the waiters decide for the group of size m, the expected winnings will be $g_m + X(t_i + d_m)$; if they decide to wait for other groups to enter during the next interval, they will be $X(t_i + \Delta t)$. Then one simply has to evaluate all those possibilities (choosing each group or none of them) and pick the move with highest expected tips.

Calculating X is, however, nearly equally easily done (because the time is discrete). If we define $p_m(t)$ to be the probability that a group of size m enters during a single time interval around time t (because it may be higher on evenings, or in general time dependent), it follows from the proposed strategy above that:

$$X(t_i) = \max\left(X(t_i + \Delta t), \{g_m + X(t_i + d_m) \,|\, m \in G\}\right) \tag{1.1}$$

where $X(t) = 0$ for $t \geq t_{end}$ and G is the set of group-sizes waiting at time t_i. Notice that we assume the restaurant's team will continue to serve the group currently present even if t_{end} gets exceeded (but not accept a new one after that). We could of course exclude groups where $t_i + d_m > t_{end}$ from the choices, but not doing so will be consistent with the next section and making the duration into a random variable so that it is not anymore easily predictable if a group will exceed t_{end} or not. Additionally, this also seems to reflect the reality quite good...

There are 2^n possibilities to consider, as for each $1 \leq m \leq n$ a group of size m can have entered (with possibility $p_m(t_i)$) or not (possibility $1 - p_m(t_i)$). So the final relation for $X(t_i)$ is given by weighting (1.1) for all those possibilities with their respective probabilities (and G defined approprietelly):

$$X(t_i) = \sum_{q \in P} p(q) \max\left(X(t_i + \Delta t), \{g_m + X(t_i + d_m) \,|\, m \in G(q)\}\right) \tag{1.2}$$

with $P = \{0, 1\}^n$ being the set of possibilities,

$$p(q) = \prod_{j=1}^{n} \left(q_j p_j(t_i) + (1 - q_j)(1 - p_j(t_i))\right)$$

the probability of possibility $q = (q_1, \ldots, q_n) \in P$ occuring at time t_i, and

$$G(q) = \{1 \leq j \leq n \,|\, q_j = 1\}$$

the set of groups waiting for possitibility q.

Using this relation between $X(t_i)$ and X at later times, one can easily calculate the function X for all time points using the dynamic-programming technique (starting at t_{end} and working backwards in time), because there's only a finite number of time points to find X at.

1.2 Random variables and distributions

Above, we assumed that both the tips earned from a particular group as well as the time to serve one group is simply a constant (depending only on the group's size m); this, however, won't reflect how real situations are very well. In reality, both of these values are quite randomly distributed as not everyone gives the same amount of tips and not everybody takes the same time to finish dinner.

Therefore, we want to generalize both of them to be *random variables* with a given distribution instead of simply scalar constants. Let $\mathbf{g_m} : [0, \infty) \to [0, \infty)$ be the probability density function for the distribution of tips earned from a group of size m, and $\mathbf{d_m} : \{k\Delta t \,|\, k \in \mathbb{N}\} \to [0, \infty)$ be the distribution probability function for the duration to serve a group of size m.

Then, we want to find $\mathbf{X}(t_i)$ as the probability density function of the earned tips from time t_i onwards when choosing best, as above. Each "value" of $\mathbf{X}$ is now however a complete random variable with density function instead of a single value.

1.2.1 Distribute the tips given

From statistics it's known that the density function $\mathbf{f_{x+y}}$ for two independent random variables $\mathbf{x}$ and $\mathbf{y}$ is given by their density functions $\mathbf{f_x}$ and $\mathbf{f_y}$ respectively as:

$$\mathbf{f_{x+y}}(z) = \int_{-\infty}^{\infty} \mathbf{f_x}(x)\mathbf{f_y}(z - s)\, ds \tag{1.3}$$

We will not show that these are really correctly normed density functions and thus the sum of two random variables again a random variable when defined as above.

Using this formula for the expressions $g_m + X(t_i + d_m)$ one can easily translate (1.1) to random variables. Instead of the maximum function we take the random variable with highest expected value:

$$\max V = \mathbf{w} \text{ for } \langle \mathbf{w} \rangle = \max \{\langle \mathbf{v} \rangle \mid \mathbf{v} \in V\} \tag{1.4}$$

where $\langle \mathbf{v} \rangle = \int_{-\infty}^{\infty} s\mathbf{v}(s)\, ds$ is the expected value of the random variable $\mathbf{v}$. For $t \geq t_{end}$, we take $\mathbf{X}(t) = \delta_0$ to be the delta-distribution for 0.

If there's no uniquely defined element with strictly maximal expected value, any of the random variables with maximal expected value should be fine. With this definition, it's clear that the maximum of (at least a finite) set of random variables is again a well-defined random variable together with a correct probability density function.

Finally, assume we're doing a random experiment with n possible outcomes, each with probability p_i. Depending on the result, we pick a value according to some probability density function $\mathbf{d_i}$. It's quite intuitive to take the overall probability density function for the resulting random variable to be:

$$\mathbf{f}(z) = \sum_{i=1}^{n} p_i \mathbf{d_i}(z) \tag{1.5}$$

Taking together (1.3), (1.4) and (1.5), we can now formulate (1.2) using random variables for all tip-earning to get $\mathbf{X}(t_i)$ itself as random variable distribution.

1.2.2 Non-constant serving times

In order to generalize the service duration to a non-constant (that is, randomly distributed) value, we need a concept of discrete random variables here (because time is discretizised). Let $\mathbf{d_m} : \{k\Delta t \,|\, k \in \mathbb{N}\} \to [0, \infty)$ be a function with $\sum_{k\in\mathbb{N}} \mathbf{d_m}(k\Delta t) = 1$. Then $\mathbf{d_m}$ describes the time to serve a group of size m, such that for any given group $\mathbf{d_m}(k\Delta t)$ gives the probability that it will take $k\Delta t$ to serve it (other time durations are not possible!). Additionally, we require that $\mathbf{d_m}(\mathbf{n\Delta t}) = \mathbf{0}$ for all $n > n_0$ and some $n_0 \in \mathbb{N}$; it's clear that for our purpose durations can not get infinitly long, and thus this condition can well be accepted.

We then can rewrite the subexpression $X(t_i + d_m)$ of (1.2) to use the distributed duration $\mathbf{d_m}$ instead of the constant one; to sum over the possible durations with associated probabilities, we can use (1.5) again.

This leads us to the final relations describing $\mathbf{X(t_i)}$ and the best strategy, using randomly distributed tips and durations:

$$\mathbf{X(t_i)}(z) = \sum_{q\in P} p(q)\mathbf{b(q)}(z) \tag{1.6}$$

where p and P are defined as for (1.2), and $\mathbf{b(q)}$ is a random variable given as:

$$\mathbf{b(q)} = \max\left(\mathbf{X(t_i + \Delta t)}, \{\mathbf{g_m} + \mathbf{X(t_i, d_m)} \,|\, m \in G(q)\}\right)$$

with $G(q)$ as for (1.2), the maximum-function interpreted as the one described in (1.4) for random variables, and

$$\mathbf{X(t_i, d_m)}(z) = \sum_{k\in\mathbb{N}} \mathbf{d_m}(k\Delta t)\mathbf{X(t_i + k\Delta t)}(z)$$

These expressions are all well-defined because both sums are finite (P is finite by itself and per requirement $\mathbf{d_m}$ gets zero above some defined duration) and also the operations sum (1.3) and maximum (1.4) of random variables are well-defined.

We'll now show that all entities stated to be random variables really have an appropriate density function satisfying $\int_{-\infty}^{\infty} \mathbf{f}(z)\, dz = 1$:

For any t_i and duration distribution $\mathbf{d_m}$ (with requirements as above) we have:

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathbf{X}(\mathbf{t_i}, \mathbf{d_m})(z)\, dz &= \int_{-\infty}^{\infty} \sum_{k\in\mathbb{N}} \mathbf{d_m}(k\Delta t)\mathbf{X}(\mathbf{t_i} + \mathbf{k\Delta t})(z)\, dz \\
&= \sum_{k=0}^{n_0} \mathbf{d_m}(k\Delta t) \int_{-\infty}^{\infty} \mathbf{X}(\mathbf{t_i} + \mathbf{k\Delta t})(z)\, dz \\
&= \sum_{k\in\mathbb{N}} \mathbf{d_m}(k\Delta t) = 1
\end{aligned}$$

Then also all $\mathbf{g_m} + \mathbf{X}(\mathbf{t_i}, \mathbf{d_m})$ are random variables with correct density functions, and also $\mathbf{b}(\mathbf{q})$ as their maximum (the sets $G(q)$ are finite). Finally we get:

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathbf{X}(\mathbf{t_i})(z)\, dz &= \int_{-\infty}^{\infty} \sum_{q\in P} p(q)\mathbf{b}(\mathbf{q})(z)\, dz = \sum_{q\in P} p(q) \int_{-\infty}^{\infty} \mathbf{b}(\mathbf{q})(z)\, dz \\
&= \sum_{q\in P} p(q) = 1
\end{aligned}$$

This last equality is shown by induction over the table-size n. For $n = 1$ we have:

$$\begin{aligned}
\sum_{q\in P} p(q) &= p(0) + p(1) \\
&= 0p_1(t_i) + 1(1 - p_1(t_i)) + 1p_1(t_i) + 0(1 - p_1(t_i)) \\
&= 1 - p_1(t_i) + p_1(t_i) = 1
\end{aligned}$$

Assume now the equation holds for n (let P_n and $p_n(q)$ be the corresponding n-sized possibility set and associated probabilities) and we show it then also must hold for $n + 1$:

$$\begin{aligned}
\sum_{q\in P_{n+1}} p_{n+1}(q) &= \sum_{(\tilde{q},q')\in P_n\times\{0,1\}} p_{n+1}(\tilde{q}, q') \\
&= \sum_{\tilde{q}\in P_n} p_n(\tilde{q})(1 - p_{n+1}(t_i)) + \sum_{\tilde{q}\in P_n} p_n(\tilde{q})(p_{n+1}(t_i)) \\
&= \sum_{\tilde{q}\in P_n} p_n(\tilde{q})\,(1 - p_{n+1}(t_i) + p_{n+1}(t_i)) = 1
\end{aligned}$$

So finally we know that our relations in (1.6) describe well-defined random variables, and we also tried to make clear why we believe these in particular relate to the best possible strategy.

1.3 A bit of reality

As a next step, we're going to think about real values and distributions to use for our formulas in order to bring them as close to reality as possible for our actual simulations. We chose to use **hours** as unit for time-measurements and **Euros** as unit for monetary values (tips, that is). Then we assumed these values for our time-frame:

$$\begin{aligned} t_{begin} &= 9.00 \\ t_{end} &= 23.00 \\ \Delta t &= \frac{5}{60} \end{aligned}$$

The size of the single table is fixed at 4 persons.

1.3.1 Tips

Tips are naturally very widely distributed, with some people giving hardly anything and others being very generous. For groups of multiple people, the tip payout seems to be pretty proportional to the group size; this is true both when a single person pays for all or each one for herself. Thus we assume a Gaussian distribution for the tips $\mathbf{g_m}$, which is of course clipped to exclude negative amounts and re-normalized accodingly:

$$\begin{aligned} \tilde{\mathbf{g_m}}(z) &= \begin{cases} 0 & z < 0 \\ \exp\left(-\frac{1}{2}\left(\frac{x-m\mu}{m\sigma}\right)^2\right) & z \geq 0 \end{cases} \\ \mathbf{g_m}(z) &= \frac{\tilde{\mathbf{g_m}}(z)}{\int_{-\infty}^{\infty} \tilde{\mathbf{g_m}}(y)\, dy} \\ \mu &= 1.00 \\ \sigma &= 0.80 \end{aligned}$$

Another possibility would be to construct a distribution function that is naturally asymmetric so that it falls down for amount towards zero faster and for amount greater than the mean value more slowly in order to resemble the minimum amount of no tips at all as well as some people giving two or three (and even more) times the mean value. We think, however, that this clipped Gaussian distribution describes quite well what an actual distribution might look like.

1.3.2 Durations

For the duration distributions (depending on the group-size m) we concluded this empirical relation for the mean duration, depending on the time of day:

$$60\mu(t) = \begin{cases} 15 + 25m & \text{for } t \geq 18.00 \\ \min(15 + 15m, 45) & \text{for } t \in [12.50, 13.50] \\ 15 + 15m & \text{otherwise} \end{cases}$$

$60\mu(t)$ is the duration in minutes, of course. The first case means that on evenings, people usually take more time for dinner than they would spend eating their meal at other times of the day, while the second should handle employees' lunch-breaks where we assume a maximal duration of 45 minutes (not only for the mean duration, see later).

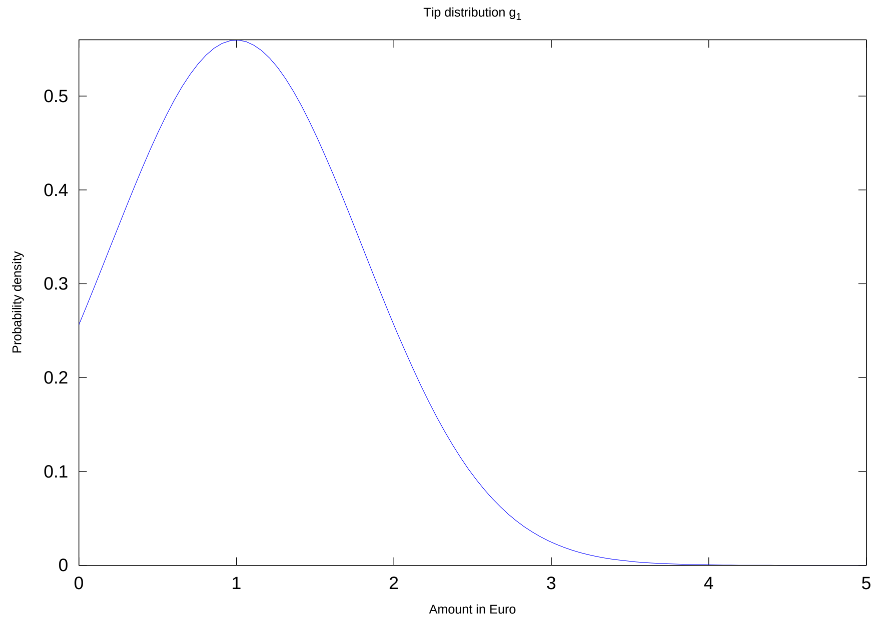

Figure 1.1: One of those tip distributions

It may be quite astonishing, but experience shows that there's not much variation in these durations (e.g. restaurants really put reservations in a tight schedule but it works out); because of this, we assume the standard deviation to be only $\sigma = \frac{7.5}{60}$.

With these nominal values, again a Gaussian distribution is assumed; as for the tips, we cap this distribution for negative durations and normalize accordingly. In addition, for the lunch-break time-frame we also cap it above 45 minutes.

Finally, in order to map these continuous distributions to our discrete time grid, we simply pack "duration probabilities" to their nearest time-point t_i. Let $\tilde{\mathbf{d_m}}$ be the continuous Gaussian distribution as described above. Then the discrete duration distribution $\mathbf{d_m}$ we require is defined as:

$$\mathbf{d_m}(k\Delta t) = \int_{k\Delta t - \frac{\Delta t}{2}}^{k\Delta t + \frac{\Delta t}{2}} \tilde{\mathbf{d_m}}(z)\, dz$$

1.3.3 Group probabilities

For the probabilities p_m of groups entering, we first decided on a relative ratio of the group-sizes themselves. Our assumption is that groups of 2, 3, 1 and 4 persons will make up 40%, 30%, 20% and 10% of all groups.

That is, for some total rate q of groups entering, we have these values for our p_m:

$$p_1 = 0.2q,\ p_2 = 0.4q,\ p_3 = 0.3q,\ p_4 = 0.1q$$

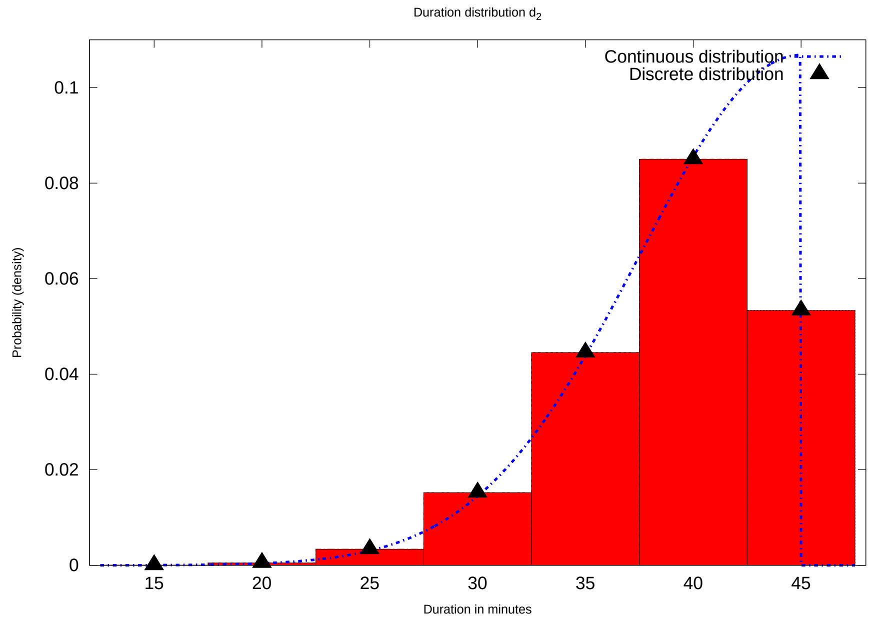

Figure 1.2: Duration for lunch-time

We do not yet, however, fix an absolute value of q. As the rate of possible customers will play a significant role for analysing our method, we will run simulations using different values.

1.4 Results

Finally, we implemented our method to both predict a distribution of tips using the arithmetic with random variables (approximated by a table numerically) and handle a real simulation using our strategy. For the simulation, we randomly "worked" for 50.000 days and generated a histogram of tip earnings; both the predicted curve and real histogram can then be compared.

While this of course is not a prove for "optimality" of our decision method, we'll see that our prediction matches the simulation pretty well, so at least our theory fits our assumptions and our implementation seems correct. This means that it really capable to make predictions, and so the decisions based on our model should be accurate.

We did however also simulate 50.000 days where the waiters always "misdecide" rather than follow our strategy. But that does not mean sending away groups (which is of course a quite fatal error), but misdecision only means to pick (another) group if there are several to choose from or if the best option would be to send all groups away. Comparison between the simulation results for our strategy and the misdecision simulation should reveal that we really advise the right thing.

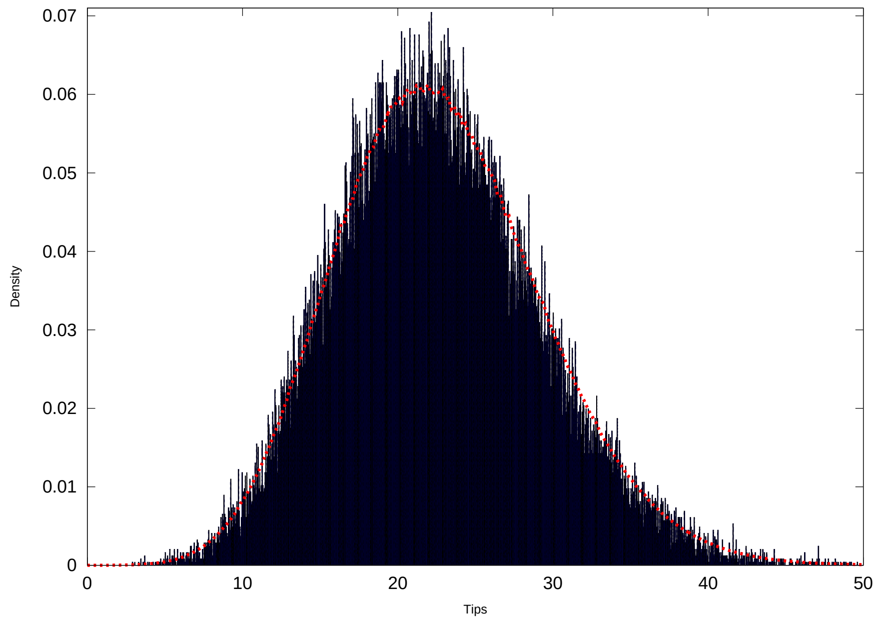

Figure 1.3: Tip distributions in theory and simulation for low customer rate

1.4.1 Low group rate

For the first simulation presented here, we assumed the group-rate q (see 1.3.3) to be $q = 0.1$. That means, we expect a fresh group roughly every 50 minutes. This corresponds to a fairly low rate. Consequently, during the simulations no group is sent away in hope for a better one to come soon.

Our theory predicts the expected value of the tip distribution (for one day) to be **22.74 Euro**, whereas the mean earnings over the simulations were **22.57 Euro**. So here's a quite good match. The resulting curve shows a roughly Gaussian distribution of the expected tips, and also a very good match between theory curve and histogram.

The earnings for misdecisions are **22.09 Euro**, so a little inferior to using our strategy, but not much in this case (as there's rarly the chance to decide any different when usually no groups are send away anyways and there aren't often multiple groups waiting).

1.4.2 High group rate

In contrast, $q = 2$ means that even more than one group is expected each 5 minute time step (for instance, the probability that a group of 2 persons enters is 80% per time step). Here, the predicted average tip earnings are **47.99 Euro** while the mean of the simulation is **47.72 Euro**. So our model fits quite well even for large amounts of possible customers.

The mean earnings when misdeciding are **38.06 Euro**, so here's quite a difference between our strategy and not following it!

The theory curve shows some small "oscillations", see 1.4.3 for an explanation of those. But overall, we once again get a bell-shaped curve and histogram,

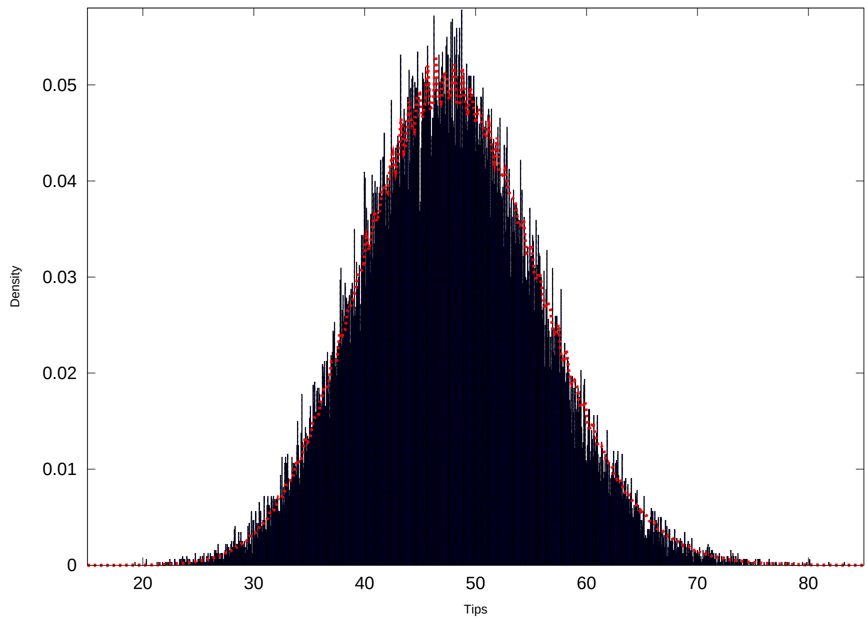

Figure 1.4: Tip distributions in theory and simulation for high customer rate

and both match as before.

For a customer rate that high, we can also observe some interesting (but quite plausible) decisions. Here's the log for one of our simulated days:

```
 9.00: Opening our doors.
 9.00: Groups are waiting:
    1, 2, 3
 9.00: We chose  3, best is  3.
10.17: It took  1.17 and we earned    5.10.
10.17: Groups are waiting:
    2, 3
10.17: We chose  3, best is  3.
11.17: It took  1.00 and we earned    3.54.
11.17: Groups are waiting:
    2, 3, 4
11.17: We chose  4, best is  4.
12.42: It took  1.25 and we earned    9.32.
12.42: Groups are waiting:
    1, 2
12.42: We chose  0, best is  0.
12.50: Groups are waiting:
    2, 3
12.50: We chose  0, best is  0.
12.58: Groups are waiting:
    2, 3, 4
12.58: We chose  4, best is  4.
13.33: It took  0.75 and we earned    4.62.
```

```
13.33: Groups are waiting:
    2
13.33: We chose  0, best is  0.
13.42: Groups are waiting:
    1, 2, 3
13.42: We chose  3, best is  3.
14.17: It took  0.75 and we earned    2.99.
14.17: Groups are waiting:
    2, 3
14.17: We chose  3, best is  3.
15.25: It took  1.08 and we earned    4.85.
15.25: Groups are waiting:
    1, 2, 3
15.25: We chose  3, best is  3.
16.25: It took  1.00 and we earned    1.38.
16.25: Groups are waiting:
    2, 3
16.25: We chose  3, best is  3.
17.33: It took  1.08 and we earned    0.78.
17.42: Groups are waiting:
    2, 3
17.42: We chose  3, best is  3.
18.25: It took  0.83 and we earned    5.72.
18.25: Groups are waiting:
    3, 4
18.25: We chose  4, best is  4.
20.17: It took  1.92 and we earned    2.93.
20.17: Groups are waiting:
    1, 3
20.17: We chose  3, best is  3.
21.67: It took  1.50 and we earned    6.18.
21.67: Groups are waiting:
    2, 3
21.67: We chose  2, best is  2.
22.75: It took  1.08 and we earned    2.11.
22.75: Groups are waiting:
    2
22.75: We chose  0, best is  0.
22.83: Groups are waiting:
    1, 2
22.83: We chose  0, best is  0.
22.92: Groups are waiting:
    1, 2, 3
22.92: We chose  3, best is  3.
24.58: It took  1.67 and we earned    1.82.
24.58: Closing.  Total tips today:   51.35.
```

Beware that all times are coded as fractional hours, so 13.50 would mean "half past 1 pm" rather than 13:50! Choosing group 0 means to send away all and wait for better ones to show up.

It is easy to see that in general, larger groups are preferred. Tips are proportional to the group size, whereas the duration also has a fixed offset and thus the tip-to-duation ratio is better for larger groups.

Generally, it seems that groups of size 2 or less are send away, while groups of size 3 or larger are accepted for this scenario. However, there are some irregularities:

At 12.50, even a group of size 3 is dismissed; that's probably because that's already in the noon-timeframe where all durations are capped at 45 minutes. Because of this, a group of 4 persons gets even more attractive, as its tip-to-duration ratio is even higher (as the longer duration is capped away). So for lunch, even larger groups are clearly advantageous. This gamble was successful, as 5 minutes later at 12.58 a group of 4 can be served. At 13.42 however, after this time-frame is over, the group-size of 3 is happily accepted again.

The next interesting thing happens at 21.67, when 2 is preferred over 3. The explanation for this behaviour is probably that we're nearing 23.00, when the restaurant will close, and with a group of 3 we risk already running over time, while 2 ensures we can still accept a group after finishing that one off. This happens at 22.75 shortly before closing, and then we once again wait for a larger group by sending away the small ones; only at the very last moment at 22.92 the final group is accepted, and served over time (as our rules allow), so the available hours are used as excessively as possible.

1.4.3 Stress-testing the model

For some special (unrealistic) parameters one can make another interesting observation that shows a good fit between our theory and the simulation results. If the restaurant's table size is assumed to be 1 (effectively only allowing 1 person groups of course), $q = 0.5$ and the standard deviation for tip distributions is reduced to 5 cents (so a very narrow distribution), the graph below results.

Our theory predicts **10.85 Euro** while the simulation mean is also **10.85 Euro** (in fact, half a cent higher). For misdecisions we get **10.86 Euro**, but that's not astonishing as there's nearly no room for any misdecisions at all in this situation, and it was by chance one cent higher than the previous simulation.

Really interesting, however, is that the tip distribution is no longer approximately Gaussian but rather has a lot of peaks (in fact, a trend to this form is already found in 1.4.2)!

In close examination, one finds that the distance between two of those peaks is 1 Euro which corresponds to the tips given by one "group". So each peak corresponds to a certain number of groups that could be served during a day (the central one to 11, for instance). Because the tip distribution is that narrow now and because also only one specific type of group is allowed, these cases are clearly distinguished from each other. Note that we did not change the service durations, but they were already fixed to a narrow distribution!

We find that even this special case is predicted well by our model and implementation, as an additional "stress-test".

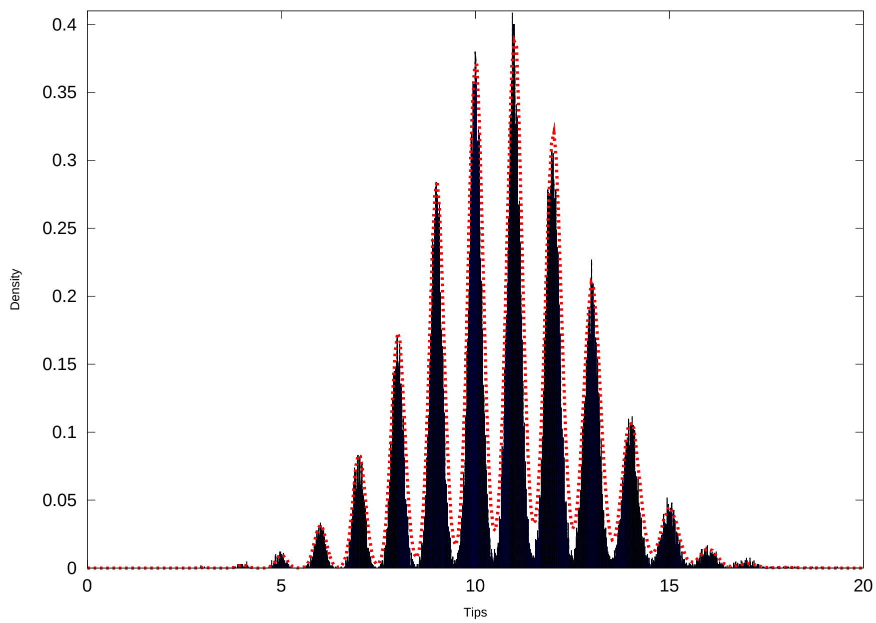

Figure 1.5: Tip distributions in theory and simulation for the stress test

Chapter 2

Multiple Tables

As a single table is not very exciting, we wanted to extend our model to a restaurant containing multiple tables, probably of varying sizes. We thought about different ideas on how to model this new situation, and while a bottom-up approach similar to the one table case would in theory be possible (see 2.1), we concluded that it was not practical for our purposes (and the way we thought about it).

So we decided to go a top-down way this time, making some assumptions and trying to get a sensible strategy out of it that would test reasonable in simulations. In fact we ended up both with a very simple approach and a more sophisticated one that is based on reducing the multiple table case to multiple single-table cases as we already solved them.

2.1 Brute-force analysis

If we leave time discretizised, once again we could try to calculate an expected tip distribution for all possible "situations" that might occur during a day, where a situation is characterized by current time of day, the occupied and free tables, and finally the remaining dining time for all those tables that are currently in use.

As there's a finite number of time-steps $T = \frac{t_{end} - t_{begin}}{\Delta t}$, a fixed number of tables anyways and also an upper bound and thus finite number of discretizised durations, say D, one can estimate the number of states of this "system" as TD^n and thus get at least a finite number. So it is in theory possible to work out relations and a dynamic programming method just as for the single-table case. This, however, will become computationally intensive soon, as D is rather large and for a non-trivial number of tables the factor D^n grows rapidly. Because of this problem, we don't think this can be practically implemented directly, and thus we did not further consider this method.

2.2 Probabilities of groups appearing

Before we start practical work on the multiple-table problem, we have to rethink our approach specifying how groups of people enter the restaurant. For a single table, we were only interested if *at least one* group of size m has entered in the last time-interval, and did specify the probability for this event to be p_m (and we assumed some "realistic" values for these probabilities, but they themselves were simply parameters).

For multiple tables, we also want to know how many groups of a given size did exactly enter. For this, we'll use a discrete random variable distribution $\mathbf{p_m}(k)$ giving the probability that exactly k groups of size m will enter ($k \in \mathbb{N}$). This is similar to our specification of serving durations in 1.2.2.

We require these properties of the distributions in order to be well-defined for our purpose and additionally reflect the fact that the probability for at least one group is already defined to be p_m (for sake of consistency):

$$\begin{aligned} \forall k \in \mathbb{N} &: \quad 0 \leq \mathbf{p_m}(k) \leq 1 \\ \sum_{k \in \mathbb{N}} \mathbf{p_m}(k) &= 1 \\ \sum_{k=1}^{\infty} \mathbf{p_m}(k) &= p_m \end{aligned}$$

Of course there are many ways to choose a distribution satisfying the basic requirements, but we want to motivate a special choice. It's probably a good assumption that there are "lots" of possible groups around, and each of them independently chooses to visit our restaurant. Let there be n such potential groups and let $p \in [0, 1]$ be the probability that each of these groups chooses to visit our restaurant. It's clear that the probability of at least one group entering is then given by $p_m = 1 - (1-p)^n$ and thus

$$p = 1 - \sqrt[n]{1 - p_m}$$

Then, for the distribution values $\mathbf{p_m^n}(k)$, we have to apply the Binomial distribution:

$$\mathbf{p_m^n}(k) = \binom{n}{k} p^k (1-p)^{n-k} = \binom{n}{k} (1 - \sqrt[n]{1-p_m})^k (1-p_m)^{1-\frac{k}{n}}$$

Because we assume that n is very large (that is, a nearly infinite pool of possible customers in the city), we'll take the limit $n \to \infty$ to arrive at the final resulting distribution $\mathbf{p_m}(k)$:

$$\begin{aligned} \mathbf{p_m}(k) &= \lim_{n\to\infty} \mathbf{p_m^n}(k) \\ &= \lim_{n\to\infty} \binom{n}{k} (1 - \sqrt[n]{1-p_m})^k (1-p_m)^{1-\frac{k}{n}} \\ &= \frac{1-p_m}{k!} \lim_{n\to\infty} \frac{n(n-1)\cdots(n-k+1)}{\left(1 - \sqrt[n]{1-p_m}\right)^{-k}} \\ &= \frac{1-p_m}{k!} \lim_{n\to\infty} \frac{n^k(1 + \frac{a_1}{n} + \ldots + \frac{a_k}{n^k})}{\left(1 - \sqrt[n]{1-p_m}\right)^{-k}} \end{aligned}$$

after expanding the binomial coefficient and for some sufficiently defined a_i. We'll see soon that the limit taking into account the constant term of the numerator polynomial is a finite number, and thus all limits containing the other factors $\frac{a_i}{n^i}$ will be zero and left out of consideration from now on. Using l'Hôpital's rule, one can finally show the existence and value of this limit:

$$\begin{aligned}
\mathbf{p_m}(k) &= \frac{1-p_m}{k!} \lim_{n\to\infty} \left(\frac{n^{-1}}{1-\sqrt[n]{1-p_m}} \right)^{-k} \\
&= \frac{1-p_m}{k!} \left(\lim_{n\to\infty} \frac{n^{-1}}{1-\sqrt[n]{1-p_m}} \right)^{-k} \\
&= \frac{1-p_m}{k!} \left(\lim_{n\to\infty} \frac{\frac{-1}{n^2}}{-\exp\left(\log(1-p_m)\frac{1}{n}\right)\log(1-p_m)\left(\frac{-1}{n^2}\right)} \right)^{-k} \\
&= \frac{1-p_m}{k!} \left(\lim_{n\to\infty} -\exp\left(\log(1-p_m)\frac{1}{n}\right)\log(1-p_m) \right)^{k} \\
&= \frac{1-p_m}{k!} \left(-\log(1-p_m)\right)^k
\end{aligned}$$

So, the distribution we'll choose for groups of size m entering our restaurant — according to our assumptions of a mass of potential groups deciding individually — is:

$$\mathbf{p_m}(k) = \frac{1-p_m}{k!} \left(-\log(1-p_m)\right)^k \tag{2.1}$$

But still it does not matter for our purpose how many groups exactly entered as long as their number is already larger than the count of available tables overall. Because of this, we later need to know the probability that *at least* s groups of some size entered, and this is given as:

$$\begin{aligned}
\mathbf{p_m}(\geq s) &= \sum_{k=s}^{\infty} \mathbf{p_m}(k) = (1-p_m) \sum_{k=s}^{\infty} \frac{\left(-\log(1-p_m)\right)^k}{k!} \\
&= (1-p_m) \left(\sum_{k=0}^{\infty} \frac{\left(-\log(1-p_m)\right)^k}{k!} - \sum_{k=0}^{s-1} \frac{\left(-\log(1-p_m)\right)^k}{k!} \right) \\
&= (1-p_m) \left(e^{-\log(1-p_m)} - \sum_{k=0}^{s-1} \frac{\left(-\log(1-p_m)\right)^k}{k!} \right) \\
&= (1-p_m) \left(\frac{1}{1-p_m} - \sum_{k=0}^{s-1} \frac{\left(-\log(1-p_m)\right)^k}{k!} \right) \\
&= 1 - (1-p_m) \sum_{k=0}^{s-1} \frac{\left(-\log(1-p_m)\right)^k}{k!}
\end{aligned}$$

In particular, one can verify that $\mathbf{p_m}(\geq 0) = 1$ as required for a random variable distribution, and this result could also have been obtained by using the complement probability for the values $k = 0, \ldots, s-1$.

2.3 Tip to duration ratio

If we dismiss sending groups away despite leaving a table empty (because we could want to save it for a "better" group that is likely to come soon), which seems like a good assumption at least if the probabilities for groups to appear are not too high and which also reflects how things are in reality, we can rate possible customers (or groups thereof) by their "tip to duration" ratio r_m (for groups of size m):

$$r_m = \frac{\langle \mathbf{g_m} \rangle}{\langle \mathbf{d_m} \rangle}$$

i.e. the expected amount of tips per time-unit that we'll have to spend for serving them.

Then, if a number of groups is waiting and some tables are free, we'll place that subset of the groups to some of the empty tables that gives the maximum sum of tip-to-duration ratios, but without further considerations.

One can easily verify that the obvious Greedy Algorithm works for this task:

1. Pick waiting group of size m such that r_m is the maximal ratio for all groups available.
2. If there's no free table of size at least m (that is, large enough) dismiss it and repeat step 1 for next group; otherwise, place the selected group at the smallest free table that has enough seats.
3. Repeat with step 1 until no longer any groups are waiting.

This gives the "optimal" strategy with our restrictions for this "top-down" approach.

We would also like to predict the expected tip random variable as we did for the single-table case, but unfortunately we can not simply apply the method described here together with the techniques in 1.2 to do so, because of the same problems described above regarding the brute-force method; namely because the search space of all possibilities gets by far too large in the multi-table case. This means that there's no direct and practical approach to finding the distribution. But for one table, this distribution itself was the basis for our decision strategy, and here we already do have the method, so the distribution is nothing totally necessary.

2.4 Reduced probabilities

If we considered the multiple-table case in such a way that for each table there would be a seperate queue of people, it would obviously be simply multiple times the single table in parallel; the difficulty is of course that this is not how it works in reality, but there's "a single queue" of people interested in a table for the whole restaurant, and in the new case it is shared among multiple tables.

For multiple tables, sending away groups is obviously less useful than for the single table case, because there's simply more capacity to handle them; and even if we hope for a better group the next round, accepting another group may be better than sending it away because there's still some table free for the larger group when it shows up.

This leads to the idea that we could try to adapt the single-table strategy to multiple tables by preventing this type of problem: For instance, when there are two tables, each table is used for roughly half of all groups (and each table can expect about half the group rate to be "its" customers). So we could simply use the single-table strategy for each table, but do all calculations with reduced group-probabilities p_m (in fact $\frac{p_m}{r_m}$ where r_m is some reduction factor).

As we already did in 2.3, it's probably best to fill free tables from smallest to largest (because a small table for some group — if sufficient — safes us the larger tables for larger groups). So for a given choice of reduction factors r_m we get this strategy:

0. For each table i of size n in the restaurant, calculate $\mathbf{X_i}(t)$ using $\frac{p_m}{r_m}$ as the group probabilities and n as "the" table size with the single-table method described above. This step is preparation and can be done once before the actual decisions have to be made.

1. Go through all free tables from smallest to largest and for each one and the waiting groups, use the single-table decision method together with the already calculated $\mathbf{X_i}$ function for this table to pick a group (or leave the table empty).

2. Repeat the last step until for each table still free the decision is to leave it empty. Then send away all remaining groups.

Finally, we present three possible methods for defining the reduction factors themselves:

No reduction The simplest strategy is to do no reduction at all, that is $r_m = 1$ for all m.

Table count reduction Define r_m to be the number of tables with at least m seats (that is, for a given group size reduce its rate by the number of tables capable of handling it).

In order reduction As before, but let r_m be the number of tables for at least m people, *that will additionally be tried before the currently calculated one or are the current table.* That is, for the first table tried (the smallest one) do no reduction at all, for the second in order reduce as if only that two tables were present, and so on.

For the special case of only one table, this method degrades (regardless of the chosen reduction strategy) to the same as for the real single-table case (because it is based on that strategy of course).

Note that here we could also try to estimate the expected tip distribution by taking all single-table estimations for all tables and summing them (as random variables). But this does not give the right estimation in general (too low for reduction and too high for no reduction); this trick of probability reduction works quite well for finding the right decisions (as we'll see later), but it gives a wrong absolute estimation.

The three reduction strategies described are of course not all possible methods, and in fact there may well be some reduction, that is somewhere "in between" no reduction and the other two strategies, and could even lead to a

Strategy	Run 1	Run 2	Run 3
Tip-to-duration ratio	426.80	426.84	426.99
No reduction	424.54	424.45	424.63
Table count reduction	426.95	426.88	426.81
In order reduction	429.41	429.37	429.24
Inverse order	427.13	427.04	427.16

Table 2.1: Multi-table results for different strategies

good fit between the resulting estimation and histogram, and/or better results than all three of our methods. But those are all we found plausible and tried ourselves so far.

2.5 Results

Once again, we tried our methods with some simulations (each run consisted now of 100.000 days simulated in our restaurant). We assumed the distributions already described in 1.3, but of course took into account our results for group appearance probabilities from 2.2. We chose $q = 2$ as the overall group rate to get quite some customers in order to fill our larger restaurant. Finally, in the simulations our restaurant had a total of 10 tables with three 2-person ones, two 3-person tables and the final five tables for 4 persons.

All resulting histograms showed a bell-shaped distribution (to no surprise of course), and all tested methods produced nearly, but not fully, the same mean tip earnings. We did each simulation three times in order to get some very rough idea of the statistical fluctuation.

The first nice thing to notice is that the resulting tips are now a little less than ten times those for a single table with $q = 2$ (50 Euro) — this is consistent with that we now have ten tables, but some smaller than 4 persons and also of course the same customer rate to fill ten tables now instead of only one; but overall, the numbers look plausible.

Notice also that the statistical differences between the three runs of a single method have a span of about 20 cents, which we'll take for a very rough approximation of the statistical error of our simulations.

Although at least the tip-to-duration strategy is fundamentally different to either probability reduction, all methods give largely the same return; this is a good consistency check that we've gotten "near" to some "best" strategy, despite the fact that all methods here are only based on "reasonable" assumptions and are of a top-down nature.

There are, however, some small differences between the simulated methods; and those differences are of single Euro order, so only within one percent of the total sum, but also larger than the statistical error, and thus mean real qualitative differences between the methods.

Probability reduction method with the no reduction strategy performs poorest, tip-to-duration and table count reduction perform better and quite similar to each other, and the most sophisticated in order reduction gives the best performance. "Inverse order" finally is an in order reduction, but here the tables are ordered largest to smallest (that is, in the opposite direction); this gives a worse result, so the order is really (a little) significant.

But all in all, we get the impression that all of our empirical methods for multiple tables probably work quite well (and are thus already close), although there are real differences. Also quite interesting is our finding that the "most obvious" method, namely tip-to-duration ratio, *is* slightly outperformed by in order reduction and thus not optimal. And here our prior analysis of the single-table case together with the probability reduction trick did help us find a better strategy.