

Efficiently Computing Bridgelets

Kevin Buchin
@kbuchin

Aleksandr Popov
@aleqss

October 2022

1 Counting Paths Through a Vertex

We can solve the following question: given a time limit T and the point (x, y) , count the paths in $W(x, y, T)$ that go through a vertex (a, b) . We can compute these counts efficiently for all possible vertices (a, b) .

A first attempt at a solution might count the paths as in the previous question starting from $(0, 0)$; and separately count the paths from (x, y) while going back in time. Suppose the cell $P_1(a, b, t)$ for some arbitrary a, b , and t holds the value s , and suppose the cell of the second DP $P_2(a, b, T - t)$ holds the value t ; then there are st paths in $W(x, y, T)$ that pass through (a, b) at time t . If we aggregate these values over time, though, we will count visits to (a, b) instead of the paths that visit (a, b) . Instead, we can let P_1 be the DP that counts the paths ending in (a, b) at time t such that these paths have *not* visited (a, b) previously. So, we can count the paths that first visit (a, b) at time t . If we leave P_2 unchanged, then the technique we used to combine P_1 and P_2 gives us the required outcome.

Finally, note that any path from $(0, 0)$ to (a, b) that first visits (a, b) at time t can be viewed in reverse; then it is a path from (a, b) to $(0, 0)$ that never visits (a, b) after leaving. Translating it, we get a path from $(0, 0)$ to $(-a, -b)$ that never revisits $(0, 0)$. We can formulate a simple DP to count this for all (a, b) at once:

$$\begin{aligned} N(0, 0, 0) &= 1 \\ N(x, y, 0) &= 0 \quad \text{for all } x \neq 0 \text{ or } y \neq 0, \\ N(0, 0, t + 1) &= 0 \quad \text{for all } t \geq 0, \\ N(x, y, t + 1) &= N(x, y, t) + N(x - 1, y, t) + N(x + 1, y, t) \\ &\quad + N(x, y - 1, t) + N(x, y + 1, t) \quad \text{for all } (x, y) \neq (0, 0) \text{ and for all } t \geq 0. \end{aligned}$$

To summarise, we can compute $N(-a, -b, t)$ for all (a, b) and t ; compute $P(a, b, T - t)$; then elementwise multiply these to get the count of paths in $W(x, y, T)$ that visit (a, b) at time t for the first time. For any (a, b) of interest, we can compute the sum of the results over all values of t . This way, in $O(T^3)$ time, we can count the paths in $W(x, y, T)$ that visit point (a, b) , for all possible (a, b) .

2 Counting with Obstacles

If we are given a particular cell (a, b) , then we can count paths in $W(x, y, T)$ that visit (a, b) for all (x, y) and $t \leq T$, also in the presence of obstacles. Note that this is different from the task we solved previously: here we fix (a, b) beforehand instead of (x, y) . To achieve this, we

simply compute $P_o(x, y, t)$ for all x, y , and t ; and compute $P'_o(x, y, t)$ with $S \cup \{(a, b)\}$. The second DP counts the paths that never visit (a, b) . Then taking $P_o(x, y, t) - P'_o(x, y, t)$ counts the paths in $W(x, y, T)$ that go through (a, b) at some time.

If we want to use the counts for computing visit probabilities, or e.g. the probability that the path ends in a certain cell, there is some subtlety here. This model can be conceptualised as follows: we take the trajectories from $(0, 0)$ in T steps uniformly at random, and then we discard any trajectories that pass through obstacles. In the figures one can see, in particular, that the counts right next to the wall are lower than the counts in symmetric cells w.r.t. $(0, 0)$. The reason is that we discard the paths that would go through the wall, so there are fewer paths ending up in the cells next to the wall. If we generate trajectories using this model, this behaviour needs to be considered. An alternative model could instead be that from any cell, we propagate to all available neighbours with equal probability. Under normal circumstances, the probabilities are all $1/5$, which is equivalent to what we get from path counting. If one neighbour is blocked, though, we can make all probabilities $1/4$; or we can keep them at $1/5$ and make the probability of staying in the current cell be $2/5$. All three models will yield subtly different outcomes, and the choice depends on the desired probabilistic interpretation. The two models briefly described here can also be implemented using a dynamic program, but propagating rational numbers representing probabilities instead of path counts.

3 Closed-Form Expressions

In this section, we provide closed-form expressions. The expressions in this section are well-known in the context of simple random walks. We merely set them into the context of bridgelets.

3.1 One-Dimensional Walks

The elementary building block is a random walk in 1D. We want to count the number of walks starting at 0 and ending at a 'location' $x \in \mathbb{N}$ after T time steps. We denote the set of these walks by $W^-(x, T)$. First we consider the case that in each time step, the walk either does a step -1 or a step $+1$. Observe that we need to have $x \leq T$, since otherwise there is no such walk. More specifically, we need to have $T = x + k$, where $k = T - x \in \mathbb{N}_0$ is even. Any walk to x in time T will make $x + k/2$ $+1$ -steps and $k/2$ -1 -steps. The walks only differ in the order in which these steps occur, i.e. picking in which of the T steps the $k/2$ -1 -steps happen. The number of such walks is:

$$|W^-(x, T)| = \binom{T}{k/2}. \quad (1)$$

Now assume we want to allow the walk to stay at the same location, i.e. we also have a $'0'$ -step. We denote this set of walks by $W(x, T)$. We keep the notation $T = x + k$, but now k does not need to be even. Consider the walks that do exactly i -1 -steps with $0 \leq i \leq \lfloor k/2 \rfloor$. This means that the number of $'0'$ -steps is $k - 2i$. For a given i , we have $\binom{T-k+2i}{i}$ possibilities to arrange the $+1$ - and -1 -steps (using Equation 1), and for each of these possibilities, we have $\binom{T}{k-2i}$ ways to pick when the $'0'$ -steps happen. Summing over all possible i , we get the expression for the number of walks to x in time T :

$$|W(x, T)| = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{T}{k-2i} \binom{T-k+2i}{i}. \quad (2)$$

3.2 Two-Dimensional Walks

As in the previous section, we first consider walks that do not stay at the same location in consecutive time steps, that is, the only possible steps are $(+1, 0)$, $(-1, 0)$, $(0, +1)$, and $(0, -1)$. These are classical random lattice walks. Given a location (x, y) and a number of time steps T , we are interested in the number of walks $|W^-(x, y, T)|$ from $(0, 0)$ to (x, y) in T steps. Similarly to the one-dimensional case, we define $k = T - x - y$. As above, k needs to be even, since otherwise the number of walks is 0. For even k , it is well known that the number of walks is

$$|W^-(x, y, T)| = \binom{T}{k/2} \binom{T}{x + k/2}. \quad (3)$$

In the following sections, we give two derivations of this equation, one that is more direct and another that provides a better explanation of the expression. These derivations are well-known, in particular in the context of random walks starting and ending at $(0, 0)$, i.e. in the special case that $x = y = 0$.

3.2.1 Direct Derivation

We can decompose any walk from $(0, 0)$ to (x, y) into two one-dimensional walks, one from 0 to x on the x -axis and another from 0 to y on the y -axis. If the walk from 0 to x takes $x + k_1$ steps, then the other walk takes $y + k_2$ steps with $k_2 = k - k_1$. Given two such one-dimensional random walks, there are $\binom{T}{x+k_1}$ possibilities of interleaving them to obtain a two-dimensional random walk.

Thus, for a given k_1 , the number of walks is $\binom{T}{x+k_1} \binom{x+k_1}{k_1/2} \binom{y+k-k_1}{(k-k_1)/2}$. To obtain the total number of walks, we need to sum over all possible values of k_1 . Defining $i = k_1/2$, we obtain

$$\begin{aligned} |W^-(x, y, T)| &= \sum_{i=0}^{k/2} \binom{T}{x+2i} \binom{x+2i}{i} \binom{y+k-2i}{(k-2i)/2} \\ &= \sum_{i=0}^{k/2} \frac{T!}{(x+2i)!(y+k-2i)!} \frac{(x+2i)!}{i!(x+i)!} \frac{(y+k-2i)!}{(k/2-i)!(y+k/2-i)!} \\ &= \frac{T!}{k/2!(x+y+k/2)!} \sum_{i=0}^{k/2} \frac{k/2!}{i!(k/2-i)!} \frac{(x+y+k/2)!}{(x+i)!(y+k/2-i)!} \\ &= \binom{T}{k/2} \sum_{i=0}^{k/2} \binom{k/2}{i} \binom{x+y+k/2}{x+i} \\ &= \binom{T}{k/2} \binom{T}{x+k/2}, \end{aligned}$$

where the last equality follows from the Chu–Vandermonde identity: $\binom{a+b}{j} = \sum_{i=0}^j \binom{a}{i} \binom{b}{j-i}$.

3.2.2 Alternative Derivation

Let $u := (1/2, 1/2)$ and $v := (1/2, -1/2)$. Observe that $\pm u \pm v$ gives exactly the four steps $(+1, 0)$, $(-1, 0)$, $(0, +1)$, and $(0, -1)$. We can therefore interpret a random walk on the lattice as the sum of two independent one-dimensional random walks, one making steps u and $-u$ and the other making steps v and $-v$. Now we have $(x, y) = (x+y)u + (x-y)v$, thus the walk in the direction of u starts at 0 and has to end at $(x+y)$, and the walk in the direction of v starts at 0

and has to end at $(x - y)$. Again denoting $T = x + y + k$ and using Equation 1, we get that the number of walks in u -direction is $\binom{T}{k/2}$, and the number of walks in v -direction is

$$\binom{T}{1/2(T - (x - y))} = \binom{T}{1/2(2y + k)} = \binom{T}{y + k/2} = \binom{T}{x + k/2}.$$

Since these walks are independent, the number of 2D walks from $(0, 0)$ to (x, y) is the product of the number of 1D walks, i.e.

$$|W^-(x, y, T)| = \binom{T}{k/2} \binom{T}{x + k/2}. \quad (4)$$

3.2.3 Incorporating $(0, 0)$ -Steps

As in the one-dimensional case, we can incorporate the $(0, 0)$ -steps by interleaving these steps with a walk without $(0, 0)$ -steps. Replacing the second term in Equation 2 by the corresponding term for a two-dimensional walk (see Equation 3), we obtain the number of walks:

$$|W(x, y, T)| = \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{T}{k - 2i} \binom{T - k + 2i}{i} \binom{T - k + 2i}{x + i}. \quad (5)$$

3.3 Visiting a Location

In this section, we show how to derive an expression for the number of walks that start at $(0, 0)$ and end at (x, y) in time T while visiting (x', y') on the way. The difficulty here lies in counting walks that visit (x', y') multiple times *once*. Since the expression as such is rather complex, we only show its building blocks here. As before, we make the derivation without $(0, 0)$ -steps, since these can be incorporated using an additional summation.

We can decompose a walk from $(0, 0)$ to (x, y) via (x', y') into a walk from $(0, 0)$ to (x', y') in t time steps and a walk from (x', y') to (x, y) in $T - t$ time steps. To obtain a unique decomposition, (x', y') should be visited at time t for the first time. Let $W^1(x', y', t)$ be the set of walks to (x', y') in time t that visit (x', y') only once. The number of walks we are interested in can be calculated as

$$\sum_{t=x'+y'}^{T-x-y+x'+y'} |W^1(x', y', t)| |W^-(x - x', y - y', T - t)|. \quad (6)$$

What is left is to obtain an expression for $|W^1(x', y', t)|$. We can calculate it as $|W^-(x', y', t)|$ minus the number of walks that visit (x', y') at least twice in time t . By symmetry, the latter number is the same as the number of walks from $(0, 0)$ to (x', y') in time t that revisit $(0, 0)$. As done previously, we can decompose these walks into walks from $(0, 0)$ to $(0, 0)$ in time t' and walks from $(0, 0)$ to (x', y') in time $t - t'$, where t' is the first time that $(0, 0)$ is revisited.

Now the only quantity we still need an expression for, is the number of walks from $(0, 0)$ to $(0, 0)$ in time t' that do not intermediately visit $(0, 0)$. Let $n = t'/2$. Then the number of such walks as a function of n is a known integer sequence with known generating function.¹

Putting all of this together would result in a rather involved expression. The simple dynamic programming approach we presented can compute all of the expressions above, with the additional advantage that it can be easily adapted to other situations.

¹See <http://oeis.org/A054474>.