Случайные блуждания на плоскости и их пределы: простое случайное блуждание, LERW и SAW

Ольга Ромаскевич, Дмитрий Челкак

Задачи к лекции №3: случайное блуждание с удаленными петлями

1. Случайное блуждание и остовные деревья: алгоритм Вилсона

Напоминание: *остовным деревом* на графе G называется дерево, содержащее *все* вершины графа G, ребра которого суть какое-то подмножество ребер графа G. Множество остовных деревьев на графе G будем обозначать как $\mathcal{T}(G)$. *Алгоритм Вилсона (Wilson's algorithm)* генерации случайного остовного дерева на (конечном) графе G состоит в построении (конечной) растущей последовательности деревьев $T_k \subset G$ следующим образом:

- T_0 – дерево из одной вершины (корня) r нашего графа, которая может быть выбрана как детерминистически, так и произвольным случайным образом;
- T_{k+1} получается из T_k выбором произвольной вершины графа $r_{k+1} \notin T_k$ и добавлением к T_k *случайной(!)* траектории LERW, построенной по случайному блужданию на G, начатому из r_{k+1}, и остановленному в момент первого удара в T_k.

Для конечных графов, докажите, что вероятность события, что остовное дерево, построенное с помощью алгоритма Вилсона, совпадает с заранее выбранным, равна $|\mathcal{T}(G)|^{-1}$ ***(все остовные деревьям появляются с равными вероятностями)***. Для этого:

1a. Осознайте, что *бесконечные* траектории случайного блуждания на графе G находятся во взаимно-однозначном соответствии с бесконечными таблицами следующего вида: столбцы таблицы пронумерованы вершинами графа, строк в таблице – бесконечное число. В столбце, отвечающем вершине v, написаны ребра $e_v^1, e_v^2, \ldots$, по которым случайное блуждание пойдет, попавши в вершину v, соответственно, в первый, второй и т.д. раз. Столбец, отвечающий вершине v, мы будем называть *стеком* этой вершины: пройдя соответствующее ребро, мы выкидываем его из стека и переходим к следующему.
Замечание: для построения остовного дерева с корнем r в соответствии с алгоритом Вилсона нам не нужно знать стек вершины r так что этот стек можно считать пустым.

1b. По каждой таблице из пункта **1a.** построим *начальный видимый граф*: для этого проведем ориентированные ребра (написанные в первой строке таблицы) из каждой вершины графа G, кроме r. Покрасим эти ребра в цвет номер 1. Если в полученном графе нет циклов, то он является остовным деревом с корнем r (почему?).

Если же в графе есть циклы, то *схлопнем* один из них. А именно, выкинем все ребра, образующие цикл и добавим ребра, идущие за выкинутыми в соответствующих стеках вершин. Добавленные ребра покрасим в цвет номер 2. Будем продолжать такой алгоритм и выкидывать циклы. В итоге возможны два случая: или процедура остановится и мы получим *набор покрашенных циклов* o, который привел к *остовному дереву* τ; или же процедура не останавливается и мы получаем всё новые и новые циклы.

Заметьте, что схлопнутые ориентированные циклы сами по себе могут повторяться (приведите пример), однако *покрашенные циклы не повторяются.* Приведите пример таблицы, в которой ребра в одном цикле могут быть покрашены в разные цвета.

1c. Рассмотрите алгоритм схлопывания циклов из пункта **1b** для заданной таблицы. Докажите, что если алгоритм остановился (т.е. число циклов в наборе o – конечно), то τ *и* o *не зависят от порядка выкидывания окрашенных циклов*. Для этого достаточно доказать (почему?), что если цикл C мог быть схлопнут в порядке $C_1, \ldots, C_n = C$, но на первом шаге был схлопнут другой цикл $C' \neq C_1$, то цикл C всё ещё может быть схлопнут.

1d. Заметьте, что алгоритм Вилсона является частным случаем описанной выше процедуры: удаление циклов "по мере появления" задает один из возможных порядков схлопывания. При этом алгоритм Вилсона останавливается с вероятностью 1 (почему?). Заметьте также, что произвол выбора новых вершин r_{k+1} оказывается несущественным: распределение вероятностей получающихся пар (τ, o) будет одно и то же (см. пункт **1c**).

1e. Рассмотрите множество X, состоящее из пар (τ, o), где τ – остовное дерево, а o – набор покрашенных циклов, приводящий к τ в результате действия алгоритма схлопывания. Докажите, что если $(\tau, o) \in X$, то $(\tau', o) \in X$ для всех остовных деревьев τ'. Докажите, что вероятность получить пару (τ, o) в результате генерации случайной матрицы на шаге **1a**, и применения к ней алгоритма **1b** *не зависит от* τ. В частности (см. **1d**), вероятность получить данное остовное дерево τ при помощи алгоритма Вилсона не зависит от τ.

2. Функция Грина. Зафиксируем дискретную область Ω (на квадратной решетке) и пусть $u \in \Omega$ – внутренняя вершина. *Дискретной функцией Грина* называется (вещественная) функция $G_\Omega(\,\cdot\,; u)$, заданная на вершинах Ω и обладающая следующими свойствами:

(i) для всех внутренних $v \neq u$ выполнено условие $G_\Omega(v; u) = \frac{1}{4}\sum_{*=+,-,\sharp,\flat} G_\Omega(v^*; u)$;

(ii) $G_\Omega(\cdot; u)$ обращается в нуль на границе области: $G_\Omega(a; u) = 0\ \forall a \in \partial\Omega$;

(iii) $G_\Omega(u; u) = 1 + \frac{1}{4}\sum_{*=+,-,\sharp,\flat} G_\Omega(u^*; u)$.

2a. Докажите, что (для заданных Ω и u) такая функция единственна.

2b. *Вероятностная интерпретация:* докажите, что $G_\Omega(v; u)$ – среднее значение числа посещений вершины u случайным блужданием, стартовавшим из вершины v (при этом мы считаем, что случайное блуждание останавливается, выйдя на границу Ω).

2c. Докажите, что $G_\Omega(v; u)$ можно задать как сумму по всем путям, соединяющим вершины u, v и остающимися *внутри области* Ω: $G_\Omega(v, u) = \sum_{\gamma: v \leftrightarrow u} \left(\frac{1}{4}\right)^{|\gamma|}$ (как всегда, $|\gamma|$ обозначает количество шагов в γ). Заметьте, что отсюда вытекает симметричность G_Ω по аргументам (u, v): $G_\Omega(v; u) = G_\Omega(u; v)$.

2d. Пусть, как на лекции, область $\Omega \setminus \{a\}$ получается из Ω удалением одного граничного ребра $a = (a_{\mathrm{in}} a_{\mathrm{out}})$. Докажите, что, для всех $u, v \in \Omega \setminus \{a\}$, верно следующее равенство:

$$G_\Omega(u; v) = G_{\Omega\setminus\{a\}}(u; v) + 16 P_\Omega(u; a) P_\Omega(v; a) / G_\Omega(a_{\mathrm{in}}, a_{\mathrm{in}}), \tag{1}$$

где $P_\Omega(u; a) = \omega_\Omega(u; a)$ – это вероятность выйти из Ω через a для случайного блуждания, запущенного из u (напоминание: эта величина также представляется в виде суммы по путям в Ω). "Непрерывный" аналог (1) называется *вариационная формула Адамара.*

2e. Пусть $N_\Omega(v; a, 0)$ – среднее количество посещений вершины v случайным блужданием $\mathrm{RW}_\Omega(a; 0)$, соединящим a и 0 внутри Ω (как и на лекции, это означает что каждый из возможных путей $\gamma : a \leftrightarrow o$ имеет вероятность, пропорциональную $(\frac{1}{4})^{|\gamma|}$). Докажите, что

$$N_\Omega(v; a, 0) = G_\Omega(0; v) P_\Omega(v; a) / P(0; a).$$

2f. Пусть, как на лекции, p_* (где $* \in \{\rightarrow, \uparrow, \leftarrow, \}$) – это вероятности сделать соответствующий первый шаг для $\mathrm{LERW}_\Omega(a; 0)$. Убедитесь в том, что

$$N_\Omega(v; a, 0) = (4 P_\Omega(v; a))^2 / G_\Omega(a_{\mathrm{in}}; a_{\mathrm{in}}) + \textstyle\sum_* p_* \cdot N_{\Omega\setminus\{a\}}(v; a_*, 0)$$

(*подсказка:* первое слагаемое – это возможные посещения v первой частью траектории (петлями $\mathrm{RW}_\Omega(a; 0)$) "от a и до последнего посещения a_{in}"), а $\sum_*$ – это вклад оставшейся части $\mathrm{LERW}_\Omega(a; 0)$, разбитый на отдельные слагаемые в зависимости от первого шага.

2g. Опираясь на пункты **2d** и **2e**, докажите (без явного вычисления p_*), что

$$P_\Omega(v; a) / P_\Omega(0; a) = \textstyle\sum_* p_* \cdot P_{\Omega\setminus\{a\}}(v; a_*) / P_{\Omega\setminus\{a\}}(0; a_*)\,.$$

Это дает другое (более длинное, но и более естественное, без "угадывания") доказательство ключевого тождества для вероятностей p_*, использованного на лекции.