Long-range percolation in one-dimension

1984 J. Phys. A: Math. Gen. 17 2349

(http://iopscience.iop.org/0305-4470/17/11/029)

J. Phys. A: Math. Gen. 17 (1984) 2349–2350. Printed in Great Britain

COMMENT

Long-range percolation in one dimension†

John L Spouge

Group T-10, MS K710, Theoretical Biology and Biophysics, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 21 February 1984

Abstract. For lattice sites at integer values, the probability of a bond between sites m and n is $a_{|m-n|}$ where $a_0 = 0$. We strengthen Schulman's criterion for absence of an infinite cluster and suggest a Monte Carlo method for determining critical probabilities. The results derive from comparison of Schulman's Bethe lattice to branching processes. We note that $a_n = pa^{|n|}$ (pa, $a < 1$) is probably the most mathematically tractable example of long-range percolation although it does not permit an infinite cluster.

Long-range percolation in one dimension was considered by Zhang *et al* (1983). The sites are the integers $\{0, \pm 1, \ldots\}$. The probability of a bond existing between two sites m and n is $a_{|m-n|}$ ($a_0 = 0$; a_1, $a_2, \ldots$ are pre-assigned). A bond's existence is independent of other bonds. The problem here is to determine when, with positive probability, an infinite cluster can occur.

Schulman (1983) gives a criterion for excluding an infinite cluster. He compared long range percolation to a Bethe lattice having an infinite number of branches emanating from each site. For each branch, (numbered $0, \pm 1, \pm 2, \ldots$) from a site, a bond joins the two sites on the branch with probability a_n ($a_n = a_{-n}$). Let us label the sites of the Bethe lattice: the root site is 0; proceeding inductively through the lattice generations, if a site labelled m gives rise to a branch with assigned probability a_n, we label the site on the other end of the branch $(m+n)$. This procedure assigns each lattice site a label, although this label will appear on infinitely many other sites. To relate the Bethe lattice to our percolation problem, consider any particular realisation of the Bethe lattice. In this realisation, delete any bond leading to a site with a label which has appeared in a previous generation of the lattice. If a label appears on more than one vertex in the same generation, leave the bond to one such vertex intact and delete the rest. The result is a Bethe lattice which remembers its history, so to speak.

A little reflection shows that the historical Bethe lattice corresponds exactly to the determination of the cluster containing 0 in the percolation problem. We determine 0's connections, then connections of the connections, etc, keeping track of where we are and where we have been, and not bothering to test connections to those sites already in the cluster. If the label n appears in the kth generation of the historical Bethe lattice, the shortest percolation path between 0 and n contains k bonds.

Instead of deleting Bethe lattice bonds according to the history back to the 0th generation, we could delete bonds in blocks of k generations, i.e., delete as before up to and including the kth generation, then view what remains of the kth generation as

† This work was performed under the auspices of the US Department of Energy.

roots for new lattices. We delete vertices on a new lattice (which consists of a kth generation root up to its $2k$th generation progeny) independently of the old lattice and the other new lattices (so a given label appears at most once in a new lattice, but may also appear in other new lattices or in the old lattice). Repeat this process at the $(2k)$th generation, then the $(3k)$th, and so on.

Because of translation invariance in the percolation problem, the (mk)th lattice generation generated by this process may be viewed as the mth generation of a branching process (e.g. Athreya and Ney 1972). We call this the k-generation branching process.

Because k-generation lattice stripping is less ruthless than stripping the whole lattice, if the k-generation branching process ultimately goes extinct, there is no infinite percolation cluster. Branching process extinction occurs if the expected number of progeny from a single progenitor is $\leqslant 1$.

Let $A_{n,k}$ be the indicator random variable for label n appearing in the kth generation if the entire lattice is stripped. The number of progeny in the k-generation branching process is $\Sigma_{(n)} A_{n,k}$. But $E(A_{n,k}) = P(A_{n,k} = 1)$, so for $k = 1$ the extinction condition

$$\sum_{n=-\infty}^{\infty} E(A_{n,1}) = \sum_{n=-\infty}^{\infty} a_n \leqslant 1 \tag{1}$$

ensures the absence of an infinite cluster. This is Schulman's (1983) result. For $k = 2$,

$$\sum_{n=-\infty}^{\infty} \left[(1 - a_n) \left(1 - \prod_{\imath=-\infty}^{\infty} (1 - a_\imath n_{-\imath} \right) \right] \leqslant 1 \tag{2}$$

ensures extinction. Analytic criteria for $k \geqslant 3$ appear prohibitively complex, but Monte Carlo (or perhaps direct computer analysis of lattice paths) can replace them.

For $a_n = pa^{|n|} (pa, a < 1)$, when a is small, (2) gives markedly better estimates of p_c (for fixed a) than (1). Schulman (1983) also noted that estimate (1) worsened for rapidly decreasing a_n. $a_n = pa^{|n|}$ are good a_n for mathematical analysis because, for them, the probability that a particular path exists has a simple form. They do not, however, allow formation of an infinite cluster.

References

Athreya K B and Ney P E 1972 *Branching Processes* (New York: Springer) ch 1
Schulman L S 1983 *J. Phys. A: Math. Gen.* **16** L639–41
Zhang Z Q, Pu F C and Li B Z 1983 *J. Phys. A: Math. Gen.* **16** L85–90