Projekt 4: Perkolation

2 april 2019

En graf är en mängd av noder, V, och en mängd kanter $E \subset \{\{u, v\} : u, v \in V\}$, dvs en delmängd av mängden av par av noder, där $\{u, v\} \in E$ ska tolkas som att grafen har en kant mellan noderna u och v. Kort sagt består grafen av ett antal noder och kanter mellan en del av paren av noder och man skriver $G = (V, E)$.

I det här projektet ska vi betrakta några olika grafer. I samtliga fall ska vi ha $V = L_n \times L_n$, där $L_n = \{-n, -(n-1), \ldots, n-1, n\}$. Vi betraktar nu

- Grafen B_n. Här består E av alla $\{(i, j), (k, l)\}$ där antingen $i = k$ och $l = j + 1$ eller $j = l$ och $k = i + 1$. M.a.o. är B_n ett vanligt kvadratiskt gitter.
- Grafen T_n. Här består E av alla kanter i B_n plus de där $k = i + 1$ och $l = j + 1$. Till skillnad från B_n har alltså även diagonala kanter åt ena hållet lagts till och T_n blir alltså ett triangelgitter.
- Grafen D_n där man till kanterna i D_n även lägger diagonalerna åt andra hållet, dvs de där $k = i + 1$ och $l = j - 1$.

Alla tre graferna är alltså en stor box med olika mängder av kanter. Skapa nu, i alla tre fall, en slumpgraf genom att för varje kant behålla den (och den kallas då *öppen*) med sannolikhet p, oberoende för olika kanter.

Använd nu Matlab till att simulera dessa processer och bestäm approximativt precis hur stort p behöver vara för att det med sannolikhet nära 1 ska finnas en väg av öppna kanter från vänster till höger sida av boxen. (Det blir förstås olika svar i de olika fallen.) ”Nära 1” beror på n, men välj ett stort n, förslagsvis $n = 500$, då det bör synas mycket tydligt för vilka p detta sker. *Tips:* Att lagra slumpgrafen genom en grannmatris är ogörligt eftersom den skulle få storlek $10^6 \times 10^6$, vilket Matlab inte klarar. Lagra istället genom att för varje nod minnas vilka av dess kanter som är öppna. För att sedan bestämma om det finns en väg från vänster till höger, tag en nod till vänster och ta, steg för steg, reda på vilka noder som ligger i dess komponent. Om den komponenten når ända till högersidan är det klart. Annars ta en nod på vänstersidan som inte var i den första nodens komponent och upprepa. Fortsätt så tills alla noder på vänstersidan är slut eller högersidan blivit nådd.

Titta gärna i G. Grimmett: Percolation (1999) för att få veta två av de exakta resultaten. Det finns ett enkelt geometriskt skäl till att svaret för B_n är vad det är. Se om ni kan finna det (man måste inte lista ut det själv utan det finns i litteraturen).