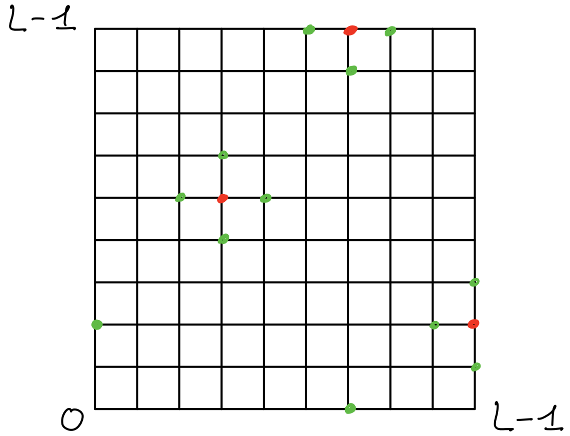
Applicazioni del Markov Chain Monte Carlo

1) Modello di Ising 2D (Metropolis):

$V = \{0, 1, \dots, L-1\}^2$



Si ha:

$$u \sim v \Leftrightarrow \|u - v\| = 1$$

Useremo le condizioni al bordo periodiche (le coordinate sono in modulo L): $L \equiv 0$

$\Rightarrow$ se $u = (i, j) \in V$, i suoi vicini sono $(i, j-1)$, $(i, j+1)$, $(i-1, j)$, $(i+1, j)$ con le operazioni $+/-$ mod L

Sia $E = \{-1, 1\}^V$, $x = (x_v)_{v \in V} \in E$, $H(x) = -\sum_{u,v : u \sim v} x_u x_v$

Se $x \in E$, le "configurazioni vicine ad x" sono tutte e sole quelle del tipo:

$$(x^v)_u = \begin{cases} x_u & \text{se } u \neq v \\ x_v & \text{se } u = v \end{cases}$$

Questa nozione di "inversione" fornisce E di una struttura di grafo regolare di grado L^2. Quindi usiamo l'algoritmo di Metropolis usando come matrice di riferimento quella della passeggiata aleatoria su tale grafo, cioè gli unici elementi non nulli di ψ sono $\psi(x, x^v) = \frac{1}{L^2}$. L'algoritmo di metropolis per $\pi_\beta(x) = \frac{1}{Z_\beta} e^{-\beta H(x)}$ ha matrice di transizione

$$Q(x, x^v) = \frac{1}{L^2} \exp\left(-\beta\left(H(x^v) - H(x)\right)^+\right), \quad \beta > 0$$

Notare che $H(x^v) - H(x) = 2 x_v \sum_{u : u \sim v} x_u$. In particolare, se $v = (i, j)$ si ha $H(x^v) - H(x) = 2 x_v (x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j-1})$.

Se $x \in E$, poniamo $m(x) = \frac{1}{L^2} \sum_{u \in V} x_u =:$ MAGNETIZZAZIONE, l'obiettivo è studiare l'evoluzione di $m(X_n)$!!!

PSEUDOCODICE:

$$X_0 = \begin{cases} \text{tutti gli spin uguali tra loro} \\ (X_0)_v = \pm 1 \text{ con probabilità } \frac{1}{2} \text{ indipendenti.} \end{cases}$$

For $n = 0, \dots, N-1$:
  scegli un vertice casuale
  if $H(X_n^v) - H(X_n) \leq 0$:
    $X_{n+1} = X_n^v$
  else:
    genera $U \sim Unif(0,1)$
    if $U < e^{-\beta(H(X_n^v) - H(X_n))^+}$:
      $X_{n+1} = X_n^v$
    else:
      $X_{n+1} = X_n$
  visualizza $m(X_{n+1})$, $n = 1, \dots, N$

---

2) Problema del commesso viaggiatore:

$E = \{$permutazioni di n oggetti$\}$. Ogni oggetto denota una città. Se $i, j \in \{1, \dots, n\}$ sono 2 città, denotiamo con $d(i,j)$ la loro distanza. Assumiamo $d(i,j) = d(j,i)$ $\forall i,j$.
Se $\sigma \in E$, definiamo:

$$H(\sigma) = d(\sigma(1), \sigma(2)) + \dots + d(\sigma(n-1), \sigma(n)) + d(\sigma(n), \sigma(1))$$

L'obiettivo è minimizzare H. Si può usare l'algoritmo di Metropolis per $\pi_T(\sigma) = \frac{1}{Z_T} \exp\left(-\frac{H(\sigma)}{T}\right)$. 2 permutazioni σ, η si dicono vicine se $\exists\, i,j$ t.c.:

$$\eta(k) = \begin{cases} \sigma(k) & k \neq i,j \\ \sigma(i) & k = j \\ \sigma(j) & k = i \end{cases}, \quad \eta = \sigma^{i,j}$$

Ciò dota E di una struttura di grafo regolare con deg $= n(n-1)$ e possiamo quindi costruire l'algoritmo di Metropolis come segue:

$$Q(\sigma, \sigma^{i,j}) = \frac{1}{n(n-1)} \exp\left(-\frac{1}{T}\left(H(\sigma^{i,j}) - H(\sigma)\right)^+\right)$$

Facciamo variare nel tempo $T = T_n$ t.c. $T_{n+1} = \alpha T_n$, $\alpha \in (0,1)$
