Tema d'esame MSN

<u>**PERCOLAZIONE**</u>

Come sapete, l'esame finale del corso è orale, ma è inteso partire dalla discussione di una relazione che presenterete su un progetto. Quest'ultimo è inteso essere una palestra in cui potrete mettere all'opera ciò che avete imparato nel corso. In particolare, non si tratterà di un progetto da iniziare da zero, ma di qualcosa da portare a termine a partire da quanto già fatto insieme a lezione.

Uno dei possibili argomenti per il vostro progetto è la **PERCOLAZIONE**. In tutto quanto segue resta inteso che studieremo il fenomeno su **reticolo quadrato bidimensionale**. Segnatamente, il progetto prevede due aspetti: uno algoritmico (la *implementazione dell'algoritmo di cluster finding HK* – dal nome degli autori Hoshen-Koperlman) e uno di comprensione dei risultati previsti dal modello (la *determinazione della soglia di percolazione*).

<u>Quanto all'aspetto algoritmico:</u>

A lezione abbiamo implementato un algoritmo (ricordate? Quello che visita il reticolo facendo crescere l'intero cluster che passa per ogni sito colorato e non ancora etichettato che incontriamo), che ci ha consentito di iniziare a studiare il modello, arrivando ad un livello di comprensione che sarà quello da cui ripartirete. Abbiamo però anche descritto la strategia per la implementazione di un algoritmo "che non si volta mai indietro e che non torna mai ad esaminare un sito già incontrato" (HK, appunto).

<u>Quanto alla determinazione della soglia di percolazione:</u>

Abbiamo visto cosa voglia dire determinare la probabilità di percolazione

- ad una fissata taglia del reticolo
- ad un dato valore della probabilità di colorazione dei siti

L'obiettivo è determinare la frequenza di percolazione a partire dalla ripetizione (per un numero sufficientemente grande di volte) dell'esperimento che consiste nelle operazioni di

- generare un reticolo e colorarlo in accordo alla probabilità di colorazione fissata
- riconoscere i cluster
- verificare la presenza di (almeno) un cluster percolante

Quanto alla determinazione della probabilità di percolazione a partire dal calcolo della frequenza di percolazione, sappiamo che la legge dei grandi numeri ci assicura che la frequenza tenderà alla probabilità nel limite in cui il numero di esperimenti tende a diventare grande. Segnatamente, *la probabilità dell'evento "si verifica percolazione" è esprimibile nel senso di valor medio della sua variabile caratteristica (quella che assume il valore 1 se il reticolo è percolante, 0 in caso contrario)*. Questo modo di esprimere le cose è estremamente utile perché ci consente sia di calcolare la frequenza come un valor medio, sia di calcolare (a motivo della definizione appena data) l'errore da associare a detta frequenza. Per la precisione, <u>se chiamiamo *dd* il vettore che raccoglie tutte le variabili caratteristiche (per *N* esperimenti indipendenti, *dd* sarà un vettore di *N* entrate, tutte a valore 0 o 1), Matlab ci farà calcolare la *frequenza* come *mean(dd)* e l'*errore* ad essa associato come *std(dd)/sqrt(lenght(dd))*</u>. Spero abbiate riconosciuto entrambe le formule (e siate in grado di ricostruire gli argomenti teorici, legati alla legge dei grandi numeri e al teorema limite centrale, che stanno alla base dell'utilizzo delle formule stesse).

Come ricorderete, il calcolo delle frequenze non è stato l'unico strumento che abbiamo utilizzato per cercare di determinare la soglia di percolazione: abbiamo definito e misurato altre osservabili (ricordate? Legate alla dimensione dei cluster e ad una qualche probabilità che un sito cadesse entro il cluster di massima taglia…) Come vedrete, vi verrà chiesto di riprendere questo modo di procedere.

La logica è ora che voi partiate da quanto visto insieme e (in un certo senso) portiate lo studio a compimento. Raccoglierete quanto farete in una RELAZIONE scritta che consegnerete (a mezzo posta elettronica) entro il mezzogiorno della giornata precedente l'esame (dovete lasciarmi il tempo di leggere…) L'esame sarà strutturato a partire dalla discussione della relazione da voi presentata.

Di seguito, le CONSEGNE d'ESAME in dettaglio.

<u>ALGORITMICA</u>

Abbiamo discusso insieme la logica e qualche spunto pratico per la implementazione dell'algoritmo HK (quello che "non si volta mai indietro").

[**c1**] Dovrete implementare l'algoritmo, ovvero scrivere un codice Matlab che lo implementi. Lo scopo è quello di confrontarlo con quanto scritto a lezione, auspicabilmente verificando che è più efficiente, ovvero più veloce nel riconoscere i cluster. Attenzione! La consegna ha due aspetti.

[**c1.1**] Per prima cosa dovrete verificare la CORRETTEZZA della vostra implementazione. Sta a voi stabilire quale sia un buon criterio a riguardo. Il consiglio che vi viene dato è di considerare affidabile l'algoritmo di ricerca cluster visto ed implementato insieme a lezione. Se volete una traccia da cui partire:

- Ovviamente potete pensare di far analizzare lo stesso reticolo ai due algoritmi.
- Dovete però pensare a cosa voglia dire verificare che essi ritornano gli stessi risultati.

[**c1.2**] Quanto ad operare il confronto fra gli algoritmi, questo vuol dire alla fine CONFRONTARE I TEMPI DI ESECUZIONE, che a sua volta vuol dire due cose

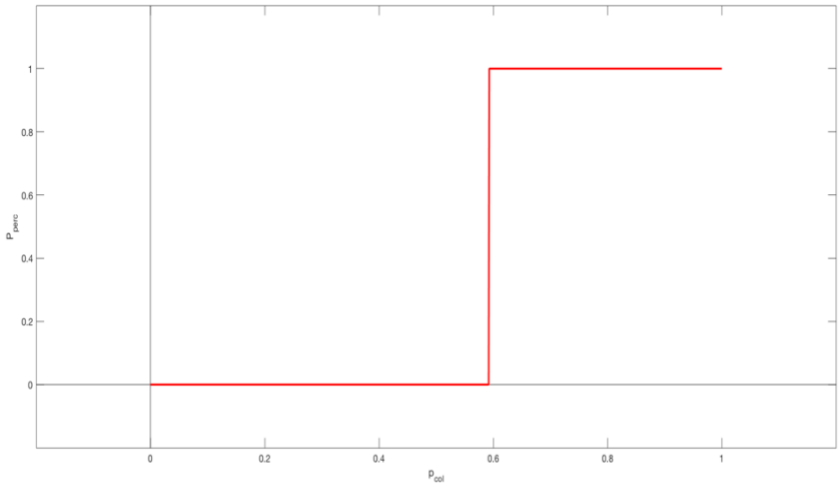
- [**c1.2.1**] Studiare l'andamento dei tempi di calcolo a probabilità di colorazione dei siti fissata e al variare della taglia del reticolo.
- [**c1.2.2**] Studiare l'andamento dei tempi di calcolo a taglia del reticolo fissata e al variare della probabilità di colorazione dei siti.

Chiaramente, i due studi comporteranno la ripetizione di tanti esperimenti a condizione fissata, in modo da calcolare tempi di esecuzione medi ed errori associati alle medie. Come detto, se la vostra implementazione sarà efficiente (oltre che corretta…) dovete attendervi che HK sia più veloce dell'algoritmo implementato insieme a lezione. Questo vuol dire che potete pensare (in quanto segue) di usare HK per condurre lo studio della soglia di percolazione.

<u>DETERMINAZIONE DELLA SOGLIA DI PERCOLAZIONE</u>

Ricordate la (prima) definizione data della soglia di percolazione? <u>La soglia è il valore della probabilità di colorazione (p_{col}) dei siti (chiamiamola p_c) tale che per valori di probabilità di</u>

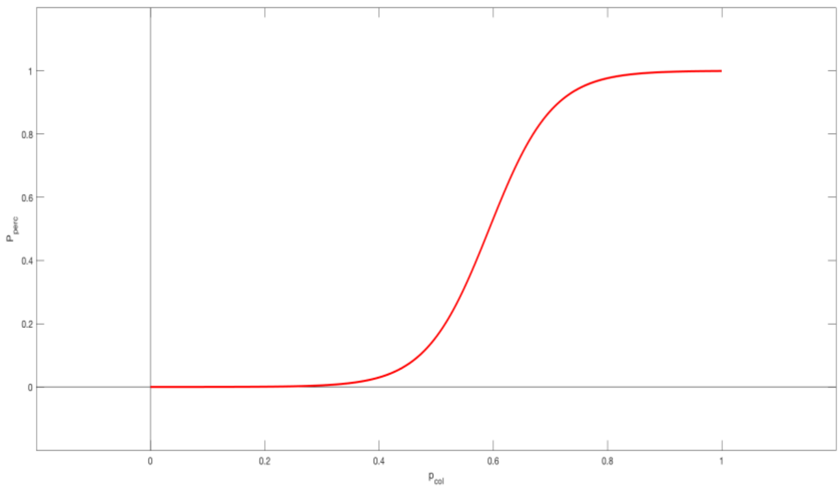
colorazione minori (p_{col} < p_c) non registriamo percolazione (attenzione! l'affermazione è molto forte: vuol dire probabilità di percolazione NULLA, $P_{perc} = 0$) mentre per valori di probabilità maggiori ($p_{col} > p_c$) registriamo percolazione (attenzione! ancora una volta l'affermazione è molto forte: vuol dire probabilità di percolazione UNO, $P_{perc} = 1$). In termini grafici, le affermazioni precedenti equivalgono a



Ancora una volta, attenzione! *Questo è quello che dobbiamo attenderci per un reticolo infinito*. Non entriamo per ora nella questione di cosa voglia dire percolazione per un reticolo di taglia infinita (pare una contraddizione: se il reticolo è infinito, non ci sono pareti e dunque … cosa vuol dire per un cluster percolare da una parete all'altra? cfr oltre…)

Con un approccio molto pragmatico, noi possiamo comunque accettare di convivere con l'idea che ci sia un risultato per il reticolo infinito se registriamo che, al crescere della taglia del reticolo, registriamo la tendenza dei risultati ad "assomigliare sempre più alla figura di sopra"…

Ciò detto, *cosa dobbiamo attenderci per un reticolo di taglia finita?* Se ricordate, i nostri esperimenti a lezione si avevano restituito qualcosa di simile a quello che vedete nella figura sotto



Attenzione! Ho barato … non sono davvero partito da una simulazione … e la figura è solo qualitativa! … la cosa importante è che la figura è (sostanzialmente) una sigmoide, non certo una funzione a gradino (quest'ultima, dovreste saperlo, ha a che fare con la funzione di Heaviside incontrata in altro contesto). La cosa ancora più importante è che *troverete che, al crescere della taglia del reticolo, la curva si fa sempre più simile ad una funzione a gradino*.

Come è facile capire, per una funzione a gradino la *definizione di soglia di percolazione* è banale: è davvero quella data poco sopra… Ma cosa potete dire nel caso di un reticolo di taglia finita? Come visto a lezione, *possiamo pensare a diverse definizioni, ad esempio*:

1. Il più piccolo valore di p_{col} (chiamiamolo p_1*) per cui risulta $P_{perc}(p_{col}) \neq 0$.
2. Il più grande valore di p_{col} (chiamiamolo p_2*) per cui risulta $P_{perc}(p_{col}) \neq 1$.
3. Il valore di p_{col} a cui troviamo il punto di inflessione.
4. Il valore di p_{col} (chiamiamolo p_3*) per cui risulta $P_{perc}(p_3^*) = 0.5$.

Dovreste convincervi che (quasi) *tutte queste definizioni tendono a coincidere nel processo di avvicinamento al limite di taglia infinita*.

La consegna di base per la determinazione della soglia di percolazione è

[**c2**] *<u>Determinare la probabilità di percolazione per diversi valori della probabilità di colorazione dei siti, ovvero (sostanzialmente) riprodurre qualcosa di simile alla seconda figura di sopra.</u>* Attenzione! Voi non otterrete una curva continua, ma una successione di punti, ciascuno corredato dal suo errore (come abbiamo imparato, questo è reso molto bene in Matlab dalla funzione *errorbar*, che abbiamo imparato ad usare a lezione). *<u>Dovete fare quanto detto per diversi valori della taglia del reticolo, per taglie sempre maggiori, con l'obiettivo di poter dire qualcosa sul risultato per la soglia nel limite di taglia infinita.</u>*

[**c2.1**] In soldoni, condurre l'esperimento di determinare la frequenza di percolazione in funzione di p_{col} (con relativi errori!) per tanti valori della taglia del reticolo comporta

- prendere in considerazione diversi valori della taglia L del reticolo (intendiamo che il reticolo ha L^2 siti);
- per ogni fissato valore della taglia L del reticolo, prendere in considerazione diversi valori della probabilità di colorazione dei siti (p_{col}) per determinare $P_{perc}(p_{col})$ (ancora una volta, i valori devono essere accompagnati dagli errori);
- per ogni fissato valore della taglia L del reticolo e per ogni fissato valore della probabilità di colorazione dei siti, il calcolo del punto precedente richiede che voi ripetiate N volte l'esperimento di base (creazione e colorazione del reticolo, ricerca dei cluster, verifica della presenza di un cluster percolante).

Come visto a lezione, *dovrete verificare che il valore che ottenete per la probabilità di percolazione in direzione orizzontale (diciamo sinistra-destra) è lo stesso (entro gli errori) che ottenete per la percolazione in direzione verticale (diciamo alto-basso)*.

La determinazione della soglia dovrebbe ora essere un compito ben definito, ma, come visto a lezione, la determinazione diretta a partire dal calcolo della probabilità di percolazione non è l'unico elemento che abbiamo a disposizione. Per questo, <u>*calcolerete anche altro*</u> e segnatamente

- [**c2.2**] calcolerete
$$P_1 = \frac{s_{max}}{L^2}$$
dove s_{max} è la massima taglia di cluster che avete trovato (mentre a denominatore leggete il numero di siti del reticolo)
- [**c2.3**] calcolerete
$$P_2 = \frac{s_{max}}{pL^2}$$
dove rispetto alla formula precedente cambia solo il denominatore (cosa rappresenta pL^2?)

- [**c2.4**] calcolerete

$$P_3 = \frac{s_{max}}{\sum_s s\, n_s}$$

dove il denominatore conta il numero di siti colorati (sommiamo su tutte le taglie *s* il valore della taglia *s* moltiplicato per il numero n_s di cluster di taglia *s*)
- [**c2.5**] calcolerete infine

$$RACS = \frac{\sum_{s \neq s_{max}} s\,(s\, n_s)}{\sum_{s'} (s'\, n_{s'})}$$

Dove RACS sta per *Reduced Average Cluster Size*, ovvero taglia media dei cluster *ridotta*. Sul fatto che si tratti di una media (e per la precisione della media della taglia dei cluster) non dovreste avere dubbi: potete riconoscere la struttura di una media pesata (la riscriviamo come media della stessa quantità rispetto a pesi generici…)

$$\frac{\sum_s s\, \mu_s}{\sum_{s'} \mu_{s'}} = \sum_s s\, \frac{\mu_s}{\sum_{s'} \mu_{s'}} \qquad \text{con} \qquad \sum_s \frac{\mu_s}{\sum_{s'} \mu_{s'}} = 1$$

Chi sia il peso dovrebbe esservi chiaro dalla struttura della formula. Perché ridotta? Come vedete, a numeratore escludiamo il contributo alla media che viene dal cluster di dimensione massima… Ma il cluster di dimensione massima a cavallo della soglia raccoglie la maggior parte dei siti colorati … e allora … che succede?

Perché mai pensiamo che studiare le quantità che abbiamo elencato ci faccia capire qualcosa sulla soglia? Guardate gli andamenti di ciascuna, ovvero considerate le quantità in questione come funzioni di p_{col}, ovvero come delle $f(p_{col})$, esattamente come fatto per la frequenza/probabilità di percolazione; dovreste sapervi dare una risposta… Per tutte queste quantità (ancora una volta) ricordate che dovete determinare anche gli errori.

[**c2.6**] Attenzione! un'altra cosa che vi viene richiesta è quale sia la interpretazione probabilistica delle P_1, P_2, P_3 (ne abbiamo peraltro parlato a lezione). Alla luce (anche) di queste, capite qualcosa di quanto succede a cavallo della soglia? Scoprirete che una risposta corretta a quest'ultima domanda vi rende ragione anche del comportamento della *RACS* … Di più: dovreste poter scoprire che una risposta corretta vi consente di dare una possibile interpretazione di cosa sia la percolazione nel caso di reticolo di taglia infinita (ricordate? se non ci sono pareti, non ha più senso parlare di cluster che percola da una parete alla parete opposta).

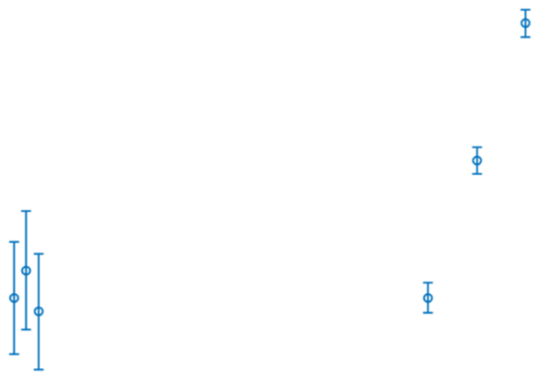
[**c2.7**] L'obiettivo ultimo è la <u>*determinazione della soglia, che (come più volte detto) otterremo considerando L che diventa sempre più grande*</u>.

Attenzione! Un aspetto fondamentale per procedere in modo efficiente è operare una scelta corretta nel campionamento dei valori di p_{col}. Tenete presente che

- è inutile dedicare troppa attenzione a valori di p_{col} per cui i risultati sono banali (il caso clamoroso sono i valori $p_{col} = 0$ e $p_{col} = 1$; sapete già la risposta)
- è invece utile avere maggiore risoluzione (e quindi valori distribuiti in modo "più fitto") dove succedono le cose significative (ovvero nella regione in cui la frequenza/probabilità di percolazione passa da *0* a *1*)

- nell’intervallo dove accadono le cose significative, non fatevi però prendere la mano: valori troppo fitti di p_{col} sono inutili se non avete un sufficiente controllo dell’errore!

I punti che vedete di sotto vi fanno capire il significato della ultima affermazione



(Sono graficamente “sottintesi” gli assi coordinati: dovete pensare l’asse *x* come quello della probabilità di colorazione; l’asse y quello di una delle osservabili prese in considerazione.) Se state cercando di determinare un andamento, i tre punti a sinistra non vi sono molto utili: i punti sono sostanzialmente indistinguibili entro l’errore. La stessa cosa non si può dire per i tre punti a destra: i punti sono molto ben distinguibili entro l’errore!

Come avete visto, le consegne fin qui assegnate vi chiedono di portare a compimento il percorso iniziato insieme a lezione. Ci può essere modo di fare qualcosa di più di questo? I più curiosi possono provare a cimentarsi con il modello in tre dimensioni. Ci aspettiamo che la soglia si sposti: si sposterà verso un valore minore o maggiore? Ripeto, questa è una variante dell’esperimento per i più curiosi.

QUALCHE NOTA SULLA RELAZIONE

Nella relazione dovete

- Illustrare in modo conciso (ma chiaro) quanto discutete (chi legge in linea di principio potrebbe non conoscere nulla del problema della percolazione).
- Illustrare il codice da voi sviluppato (la implementazione dell'algoritmo HK).
- Presentare con chiarezza la vostra comprensione del problema ogni volta che dovete discutere un punto concettuale (ad esempio, la interpretazione della *RACS*).
- Presentare con chiarezza i risultati numerici, in particolare calcolando sempre gli errori; ricordate che le figure devono essere di facile interpretazione e devono essere sempre commentate adeguatamente (in una legenda o nel corpo testo); allo stesso modo, se riportate tabelle, il significato delle entrate deve essere chiaro.
- Fate attenzione ad usare una notazione consistente; poche cose mettono in difficoltà chi legge come trovarsi di fronte ad una stessa quantità chiamata in modo diverso in punti diversi…

Se avete dubbi, fatevi vivi: possiamo riguardare insieme qualcosa, se serve.

Ricordate anche che in sede di esame è importante avere sotto mano Matlab ed essere pronti a dimostrare praticamente qualcosa (un tipico esempio: chiedo molto spesso di mostrare che la implementazione di HK sia corretta, facendo analizzare uno stesso reticolo per mezzo del codice sviluppato a lezione e per mezzo del codice sviluppato da voi).

BUON LAVORO!