

Percolation - Econophysics: Fluctuations of the stock market

Pierfrancesco Menia
pierfrancesco.menia@gmail.com

1 Introduzione e metodo utilizzato

L'obiettivo del progetto è quello di simulare le variazioni di valore "R" di un bene nel mercato azionario utilizzando algoritmi di percolazione applicati ad un reticolo quadrato di lato L. Nel modello adottato, ogni sito nel reticolo viene occupato con probabilità p e rappresenta un agente di borsa. I vari siti, raggruppati in cluster di percolazione (in contatto tra loro per almeno uno dei 4 lati), corrispondono ad agenti che si influenzano a vicenda nelle scelte di mercato. Ogni cluster ha una probabilità p_a di essere attivo e, se attivo, ha una probabilità p_b di comprare e $1 - p_b$ di vendere. La variazione di prezzo R viene calcolata ed ottenuta nel seguente modo:

$$R = \sum_{buy} sn_s - \sum_{sell} sn_s \quad (1)$$

dove s indica la dimensione di un cluster, ovvero il numero di siti al suo interno, e $n_s(p)$ è la "cluster size distribution" o CSD, così definita:

$$n_s(p) = \frac{\text{numero di cluster di dimensione } s}{\text{numero totale di siti nel reticolo}} \quad (2)$$

In particolare, si vuole verificare come in un mercato poco attivo (piccoli p_a) le fluttuazioni di R siano maggiori e la sua distribuzione $P(R)$ sia caratterizzata da un maggiore coefficiente di curtosi γ , mentre come in un mercato caratterizzato da un'alta attività le fluttuazioni di R si regolino tra loro avvicinando $P(R)$ ad una Gaussiana e quindi a $\gamma \sim 0$ (1).

2 Funzionamento del programma: prima parte

2.1 Riempimento del reticolo

Per prima cosa viene creato un reticolo quadrato di lato $L+2$ di valori interi inizialmente posti a zero. In questo modo si ottiene un reticolo di lato L su cui lavorare con un contorno di siti vuoti, in modo da rendere più semplice l'algoritmo di ricerca dei cluster come descritto in seguito.

Per ogni sito viene generato un numero casuale compreso tra 0 e 1, ed esso viene confrontato con il valore di probabilità d'occupazione p . Il relativo sito viene quindi posto a 0 oppure ad 1 sulla base del risultato del confronto.

2.2 Ricerca dei Cluster

Il passo successivo si occupa della ricerca dei vari cluster. Allo scopo è stato implementato un algoritmo di "labelling" che si svolge in due passaggi:

- Il primo ("labelling") si occupa di trovare siti connessi rilevando anche eventuali connessioni in coppia.
- Il secondo ("relabelling") che si occupa di unire i cluster connessi che al primo passaggio presentino connessioni in coppia.

Questo algoritmo appartiene alla classe di "labelling a connettività 4".

2.2.1 Labelling

Viene inizializzata al valore 0 una variabile "*tag*" che verrà utilizzata come indice per i vari cluster identificati, e che verrà incrementata opportunamente al bisogno.

Vengono anche create due liste per contenere i valori delle eventuali connessioni in coppia.

La routine itera tra tutti i siti occupati (quelli a valore 1), ricercando dei punti di contatto. A tal fine per ognuno di essi vengono controllati i siti vicini, ovvero i due siti immediatamente adiacenti alla sua sinistra e soprastante (chiamati West e North). Avendo creato inizialmente un reticolo di dimensione superiore, non bisogna preoccuparsi di porre delle condizioni nel caso in cui il sito preso in esame si trovi lungo i bordi.

Infine al sito viene associata la "*label*" secondo le seguenti regole eseguite in ordine di priorità:

- Se solamente uno dei due siti vicini è non nullo, viene assegnato al sito il valore di *label* del sito non nullo.
- Se entrambi i siti vicini sono non nulli, viene assegnato al sito il valore di *label* corrispondente al minimo tra i due (il caso è banale se i due siti vicini hanno lo stesso valore di *label*).
Nel caso in cui i valori siano diversi, essi vengono anche inseriti nelle due liste rispettive, una contenente il maggiore dei due, "*pair_hi*", e l'altra contenente il minore, "*pair_lo*".
- Se i due siti vicini sono entrambi nulli, il valore corrente della variabile *tag* viene incrementato ed assegnato al sito in oggetto.

Alla fine del primo passaggio, tutti i siti occupati (valore 1) sono stati reindicizzati con un valore di *label* come descritto, ed inoltre le due liste "*pair_hi*" e "*pair_lo*" contengono i valori di coppia di connessione ritrovati in questo passaggio. La variabile *tag* conterrà quindi il numero massimo delle *label* utilizzate.

Lo step successivo (il "relabelling") utilizzerà questi valori per unire cluster connessi, ma che inizialmente erano stati valorizzati con due *label* differenti.

2.2.2 Relabeling

Il secondo passaggio si occupa di generare una tabella di equivalenza tra le varie label che durante la fase precedente erano state associate a cluster differenti, ma nello stesso tempo anche riconosciute come connesse tramite le coppie "*pair_hi-lo*". Di conseguenza i relativi siti devono essere etichettati come siti appartenenti allo stesso cluster. L'operazione viene eseguita iterando per ogni valore di label, partendo dal valore più basso fino al massimo delle label identificate al passo precedente, richiamando la routine "*find_tree*" descritta nel dettaglio nel paragrafo sottostante.

2.2.3 Routine *find_tree*

Questa è la routine fondamentale che si occupa di generare la tabella di equivalenze utilizzata per il relabeling dei siti.

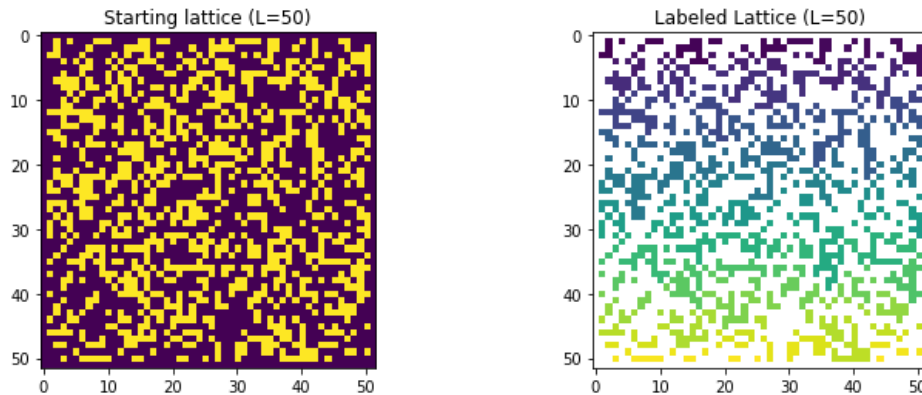
Essa prende in input le liste *pair_hi* e *pair_lo* contenenti le coppie (di cluster) inizialmente identificate come legate, ed il parametro "*label_target*" corrispondente al valore di *tag* sul quale viene effettuata la ricerca di equivalenza. In uscita viene generata una lista di label che vengono riconosciute come connesse al "*label_target*" passato in ingresso.

Di conseguenza tutti i siti con label in equivalenza vengono aggiornati e quindi uniti alla label di ingresso. Inoltre tutte le label equivalenti vengono settate con un flag a "done" perché a quel punto non più necessarie essendo già state unite.

Entrando nel dettaglio, l'algoritmo della routine di ricerca è implementato nel modo seguente:

- Viene creata una lista vuota *s* alla quale viene inizialmente aggiunto il "*label_target*" preso in considerazione.
- Viene ricercato ogni elemento di *s* all'interno di *pair_lo* e, per ogni caso di esito positivo, se non già presente, viene aggiunto ad *s* l'elemento di *pair_hi* corrispondente allo stesso indice. Di seguito la stessa cosa viene ripetuta ma invertendo la ricerca su *pair_hi* ed inserendo da *pair_lo*. Ogniqualvolta viene effettuato un nuovo inserimento in *s* viene settato un flag ad indicare la necessità di reiterare la ricerca dato che sono stati introdotti elementi nuovi in *s*.
- Al termine, quando la lista *s* rimane invariata, ne viene rimosso il valore iniziale "*label_target*". In uscita quindi, la lista *s* contiene tutti i *tag* (label) equivalenti e quindi collegati a *label_target*, e che, di conseguenza, devono appartenere allo stesso cluster.
- A questo punto, viene eseguito un loop su tutti i siti del reticolo assegnando il valore *label_target* a tutti quelli le cui label sono contenute in *s*.

Al termine, tutti i siti all'interno di uno stesso cluster sono correttamente etichettati con lo stesso *tag* e si può procedere con la seconda parte del programma.



3 Funzionamento del programma: seconda parte

3.1 Identificazione di cluster percolanti

Si esegue la ricerca di eventuali cluster percolanti orizzontali e verticali all'interno del reticolo. Per fare ciò viene eseguito un loop per ogni sito presente su un lato del reticolo verificando se lo stesso *tag* (label) sia eventualmente presente su uno qualunque dei siti del lato opposto. Questo viene eseguito sia in orizzontale che in verticale (*lato_sx* vs *lato_dx* e *lato_hi* vs *lato_lo*). In caso affermativo, quel *tag* corrisponde ad un cluster percolante e viene aggiunto ad una lista dedicata, in modo da poter essere eventualmente trascurato nei calcoli futuri.

3.2 Calcolo della dimensione dei cluster e riduzione ai soli cluster attivi

Il calcolo della dimensione dei cluster (del numero dei siti al suo interno) viene implementato nel modo seguente:

- Viene inizializzato un array di zeri "*flag_list*" di dimensione pari a *max_tag*, il valore massimo di *tag* ottenuto durante la prima fase di ricerca "labelling".
- Viene eseguito un ciclo che scorre tutti i siti del reticolo. Per ogni sito con label diversa da zero, viene incrementato di 1 il valore dell'array di indice relativo a quel *tag*. L'array contiene quindi la lista delle dimensioni dei vari cluster. Va notato che visto che gli indici degli array iniziano da zero mentre i valori delle label iniziano da 1, l'indice utilizzato viene decrementato di 1 prima del suo utilizzo. Non avendo minimizzato i *tag* (label) nel reticolo, all'interno di *flag_list* si troveranno anche valori nulli.
- Viene quindi creato un ulteriore definitivo array "*lab*" nel quale vengono inseriti solamente *tag* (label) non nulli e quindi realmente da utilizzare per il nostro scopo, ed i cui elementi sono il numero di siti (la dimensione) del *tag* (cluster) relativo.

3.3 Calcolo della distribuzione di R nel tempo

Prima di passare alla descrizione del calcolo della variazione di prezzo *R* bisogna calcolare ancora una grandezza, la "cluster size distribution" $n_s(p)$. La CSD è la distribuzione normalizzata dei cluster aventi la stessa dimensione.

3.3.1 Calcolo della CSD

Il calcolo di $n_s(p)$ avviene eseguendo un ciclo su tutti gli elementi di *lab*, ricavando quante volte lo stesso valore compare al suo interno (quanti elementi della stessa dimensione sono contenuti in *lab*), e dividendo per il numero totale di siti del reticolo. In questo modo, ad ogni cluster di dimensione *s* viene associata la sua $n_s(p)$.

3.3.2 Calcolo di R

A questo punto il calcolo di R viene eseguito nel modo seguente:

- Viene inizializzato $R = 0$.
- Per ogni elemento in lab , viene verificato che sia diverso da 1, poiché non si considerano cluster banali, cioè costituiti da un unico elemento.
- Se diverso da 1, viene generato un numero casuale da 0 e 1 e confrontato con p_a per determinare l'attivazione del cluster. Se non attivo, il cluster non contribuisce alla variazione di prezzo.
- In caso di esito positivo, il cluster è attivo e viene generato un secondo numero casuale tra 0 e 1 da confrontare con p_b :
 - Se maggiore di p_b , il cluster compra e quindi viene sommata ad R la dimensione del cluster preso in considerazione moltiplicata per la cluster size distribution corrispondente.
 - Se minore di p_b , il cluster vende e quindi sottratta ad R la stessa quantità sopra descritta.

Questo processo viene eseguito all'interno di un ciclo su un intervallo di tempo Δt predeterminato, ottenendo quindi una distribuzione dei valori di R per ogni time step.

3.4 Grafici

Viene quindi costruito un istogramma dei valori di R ottenuti e viene calcolato il "coefficiente di curtosi" γ della distribuzione tramite la formula:

$$\gamma = \frac{m_4}{m_2^2} - 3 \quad (3)$$

dove con m_k viene indicato il momento di ordine k della distribuzione.

Per una distribuzione Gaussiana si ha $\gamma = 0$, per cui questo coefficiente permette di determinare quanto la distribuzione $P(R)$ si discosti da un andamento gaussiano.

4 Analisi dei dati e risultati

Il programma prevede di poter impostare con facilità i parametri su cui eseguire l'elaborazione. In particolare:

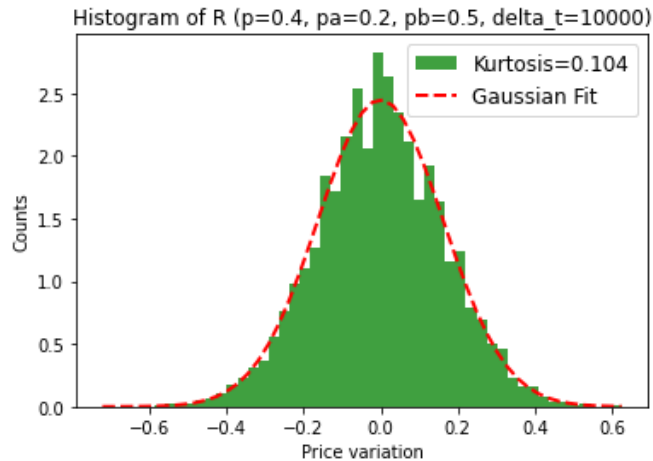
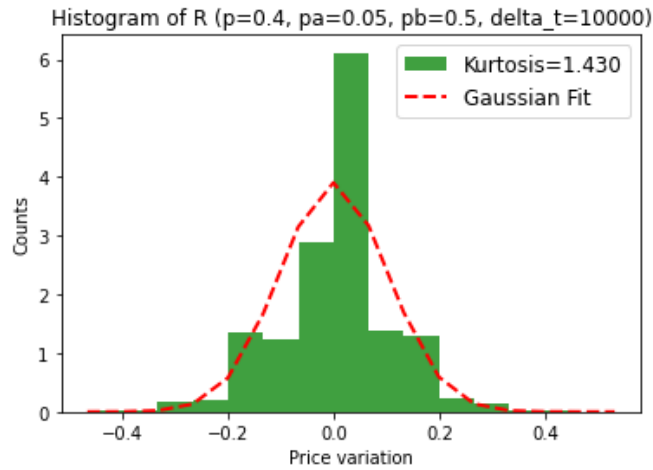
- L = valore del lato del reticolo
- p = probabilità di occupazione
- p_a = probabilità di attivazione
- p_b = probabilità di acquisto
- $time_iter$ = numero di iterazioni utilizzato per simulare il tempo che scorre

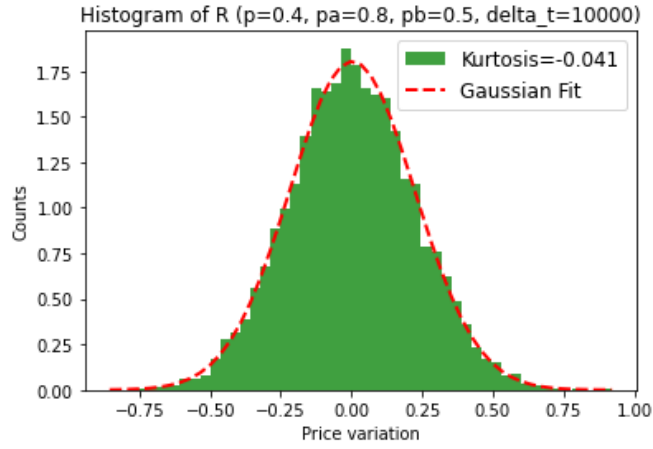
Nei paragrafi che seguono vengono riportati alcuni esempi significativi di istogrammi relativi ai risultati ottenuti variando il parametro p_a , il tutto per tre diversi valori di L . Ogni grafico rappresenta una distribuzione di dati iterata per 10.000 step.

Inoltre, ogni grafico riporta in testa le informazioni sul valore dei parametri utilizzati per i calcoli.

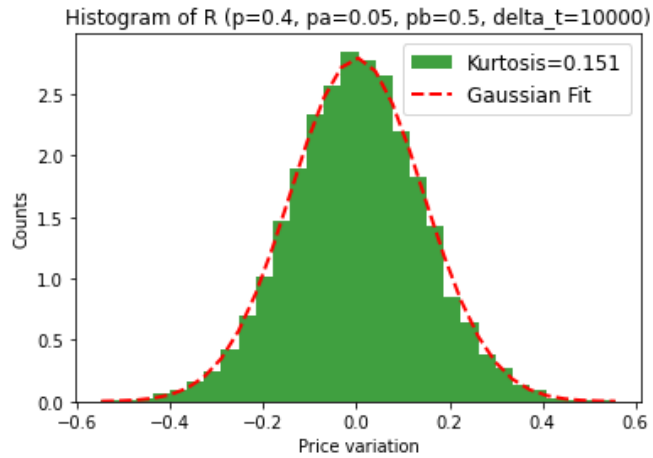
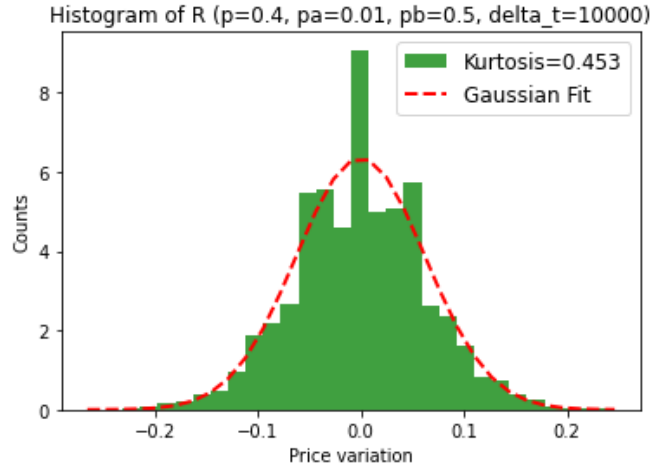
4.1 Reticolo $L=16$

In ogni istogramma viene riportato il valore di γ ottenuto per quella distribuzione. Si può notare come, all'aumentare di p_a , esso tenda sempre più a 0 a supporto della tesi che $P(R)$ tenda ad una gaussiana all'aumentare dell'attività del mercato.

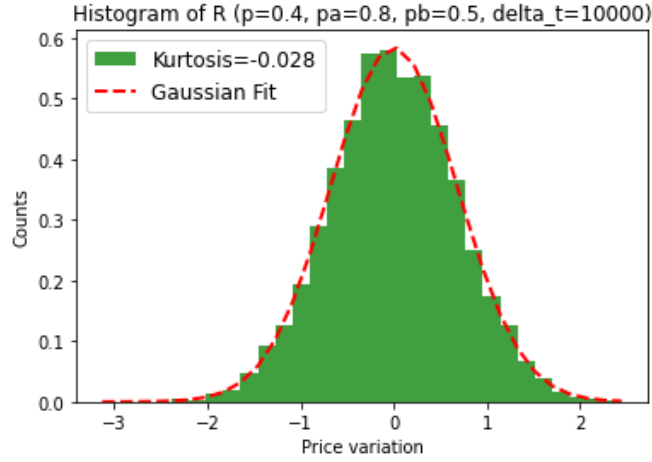




4.2 Reticolo $L=50$

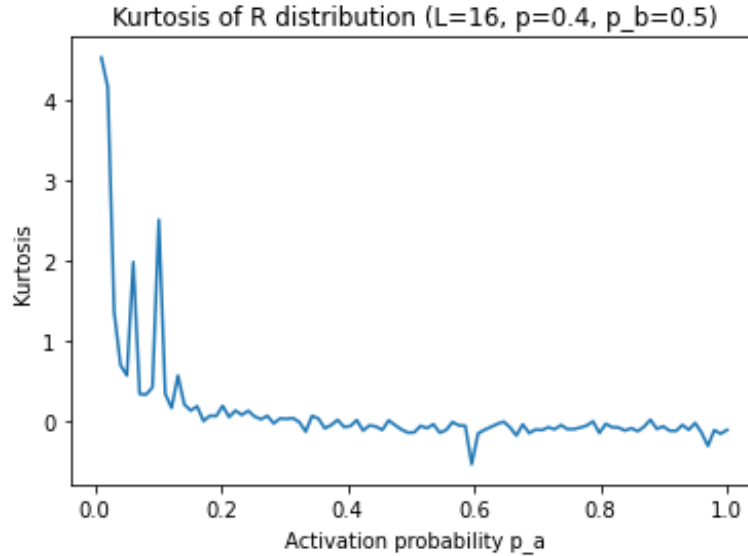


Anche in questo caso si osserva il comportamento atteso al variare di p_a ma è interessante notare come, per poterlo apprezzare, ci sia bisogno di scendere a valori più bassi di probabilità di attivazione rispetto al caso precedente e che la stabilizzazione della distribuzione avvenga già intorno a $p_a = 0.05$, valore per cui $P(R)$ nel reticolo con $L = 16$ presentava ancora un valore di coefficiente di curtosi γ lontano da 0.



4.3 Analisi dell'andamento di γ

Viene riportato l'andamento del coefficiente di curtosi γ al variare della probabilità di attivazione p_a ottenuto per $L = 16$:



Si osserva come già oltre $p_a \simeq 0.2$ il valore di γ si stabilizzi attorno a 0.

Alla luce di questo risultato, viene riportato l'andamento di γ in funzione di $p_a \leq 0.2$ per $L = 50$ e per $L = 100$ (Figura 2).

Vengono presi 500 valori di p_a in questo intervallo in modo da poter osservare la decrescita di γ anche per reticoli di queste dimensioni, apprezzabile rispettivamente attorno a $p_a \sim 0.005$ e $p_a \sim 0.0005$.

La scelta di non prendere in considerazione $0 \leq p_a \leq 1$ ma di limitarsi ad una frazione di tale intervallo in quest'ultima analisi viene da necessità computazionali. Viene nuovamente osservato il comportamento atteso, ma è interessante notare come il valore massimo di p_a sotto il quale è possibile osservare il discostamento dall'andamento Gaussiano di $P(R)$ decresce con l'aumentare delle dimensioni del reticolo.

4.4

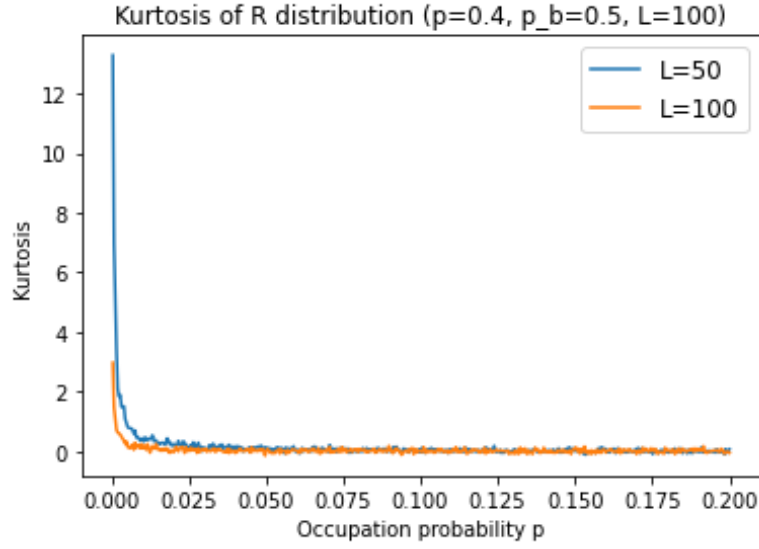


Figura 2: Andamento di γ in funzione di p_a per $L = 50$ e $L = 100$

4.5 Risultati per $p \geq p_c$

La teoria sulla percolazione ci dice che esiste un valore critico di probabilità d'occupazione p_c oltre il quale, in un contesto ideale ($L \rightarrow \infty$), si otterrebbe sempre almeno un cluster percolante per ogni configurazione di reticolo generata. Nel caso di reticolo quadrato, per il quale non esiste una soluzione analitica ma solamente computazionale, si ottiene un valore di $p_c = 0.59 \pm 0.05$. Vengono riportati il reticolo etichettato e gli istogrammi di R ottenuti per $p = p_c$ e $p_a = 0.25$ per un reticolo dove $L=50$:

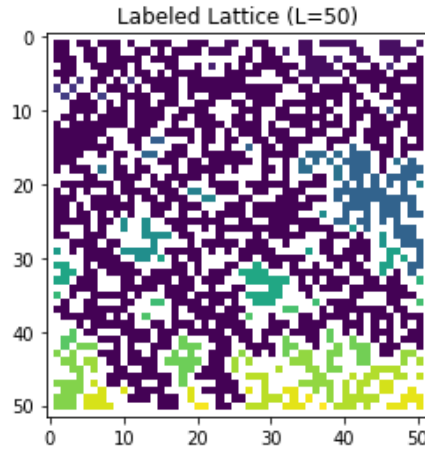


Figura 3: Reticolo etichettato in cui è ben visibile il cluster percolante in color viola

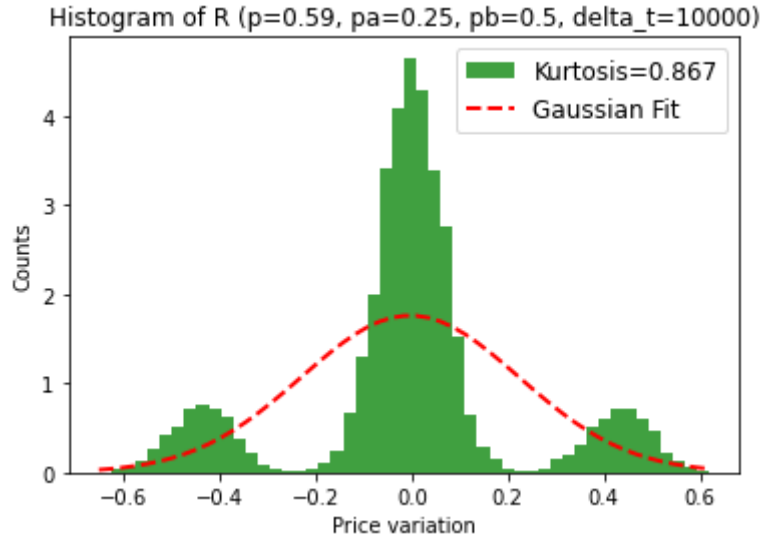


Figura 4: Istogramma di R mantenendo il cluster percolante

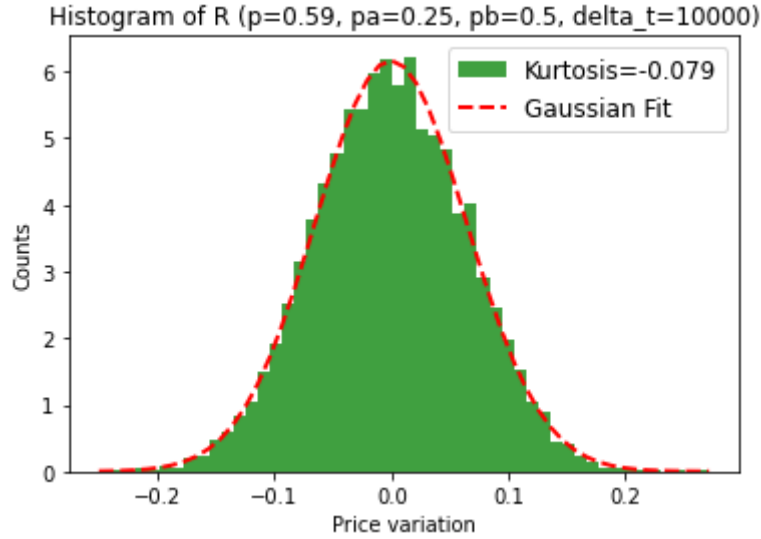


Figura 5: Istogramma di R eliminando il cluster percolante

È interessante notare come, a parità di configurazione nel reticolo, la presenza del cluster percolante comporti una deviazione di $P(R)$ dalla distribuzione gaussiana (Figura 4). Questo è probabilmente dovuto al fatto che i cluster percolanti contengano buona parte dei siti occupati nel reticolo: nel caso in cui essi siano attivi contribuiscono considerevolmente al valore finale di R sul singolo time step, compromettendo l'uniformità della distribuzione $P(R)$. Questo risultato trova riscontro nel contesto del modello economico che si vuole riprodurre: in un mercato in cui è presente un player che ne controlli una considerevole maggioranza, la variazione del prezzo di un bene sarà influenzata in maniera predominante dalla sua attività. Infatti, eliminando il cluster percolante come visibile in Figura 5, viene ripresa l'uniformità della distribuzione.

Quanto detto è ovviamente amplificato per valori limite di p e p_a , come possiamo osservare nei casi seguenti:

- Esempio 1: per $p = p_a = 0.9$, quasi tutti i siti sono occupati, e la maggior parte di essi appartiene allo stesso cluster percolante di dimensione s e cluster size distribution (SCD) $n_s(p)$. Il risultato di ciò è che la distribuzione di R dipende quasi unicamente dal fatto che questo cluster compri o venda, risultando quindi in due picchi simmetrici a $\pm sn_s(p)$, considerando $p_b=0.5$. (Vedi Figura 6)
- Esempio 2: per $p = 0.9$ e $p_a = 0.1$, come prima la maggior parte dei siti è occupata ed appartenente allo stesso cluster, ma l'attività del mercato è molto bassa. Ne consegue che il mercato rimarrà quasi del tutto inattivo per buona parte del tempo e quindi la distribuzione $P(R)$ presenterà un picco sullo 0. (Vedi Figura 7)

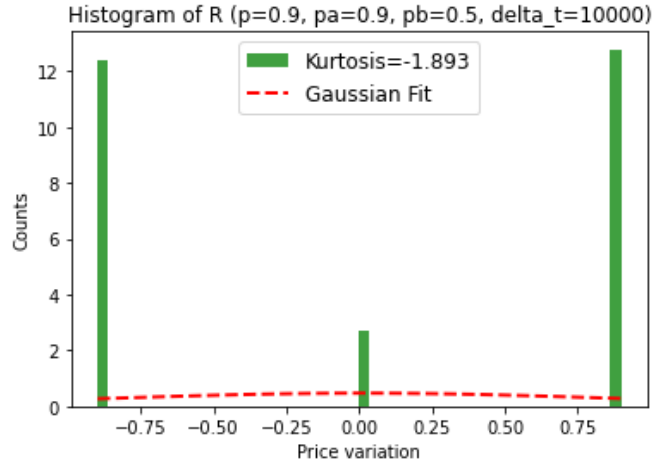


Figura 6: Istogramma di R per $p = p_a = 0.9$.

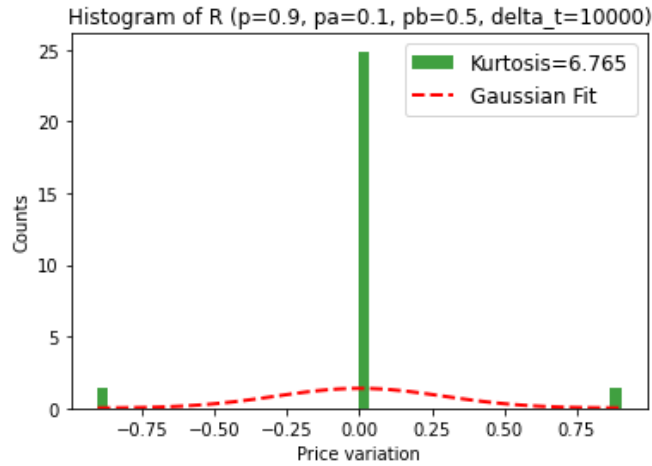
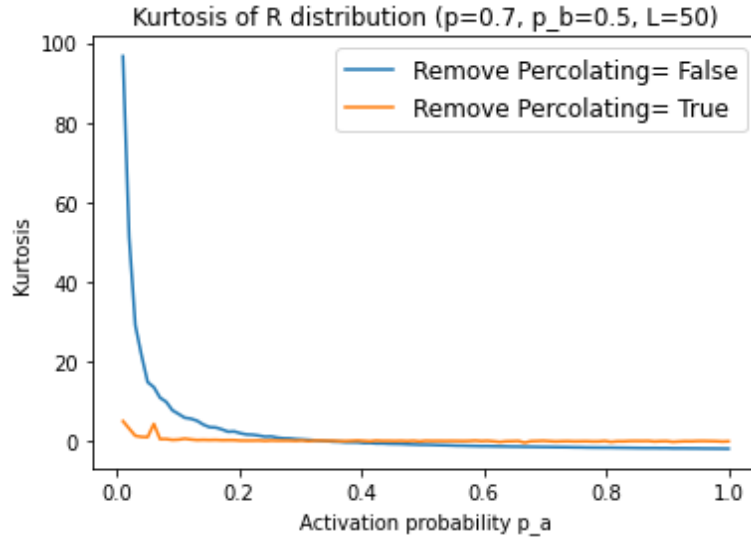


Figura 7: Istogramma di R per $p = 0.9$ e $p_a = 0.1$.

Infine, nel grafico sottostante, viene riportato l'andamento del coefficiente di curtosi γ in funzione di p_a per $p = 0.7 > p_c$ in due diverse simulazioni: la prima include il contributo dei cluster percolanti che si sono venuti a creare, mentre nella seconda tale contributo è stato escluso forzando il calcolo in modo da non considerarlo nel calcolo di R.



Si osserva come, nel caso in cui i cluster percolanti vengano rimossi, la curtosi, dopo una prima fase di decrescita nella zona di piccoli p_a , si stabilizzi attorno a 0 come atteso, mentre nella simulazione in cui i cluster percolanti sono inclusi parta da valori più alti e continui a decrescere senza mai stabilizzarsi.

5 Conclusioni

Alla luce dei risultati riportati, si può concludere che l'andamento della distribuzione $P(R)$ rispecchi quanto atteso: rimanendo al di sotto del valore di probabilità di occupazione critica p_c , per piccoli valori di p_a e quindi in un mercato poco attivo $P(R)$ presenta delle cosiddette "fat tail" rappresentate da una curtosità che si discosta da 0, allontanandosi quindi dall'andamento Gaussiano; per valori più alti di p_a invece, il contributo al calcolo di R è dato dalla somma su diversi cluster e quindi, per il teorema del limite centrale, la distribuzione $P(R)$ tende ad una gaussiana con una curtosità prossima a 0.

Inoltre, dalle simulazioni emerge come le dimensioni del reticolo influiscano sull'attività minima che deve avere il mercato affinché le fluttuazioni di prezzo R si regolino tra loro e si distribuiscano uniformemente secondo una gaussiana.

Infine, le simulazioni per $p \geq p_c$ hanno mostrato come la presenza di cluster percolanti contenenti una parte maggioritaria dei siti occupati influenzi significativamente la distribuzione di R a seconda della loro attività, come ci si aspetta che accada in un mercato in cui è presente un player che ne controlli una fetta considerevole.

6 Bibliografia

1. R. Cont and J.-P. Bouchaud, "Herd behavior and aggregate fluctuations in financial markets," cond-mat/9712318.