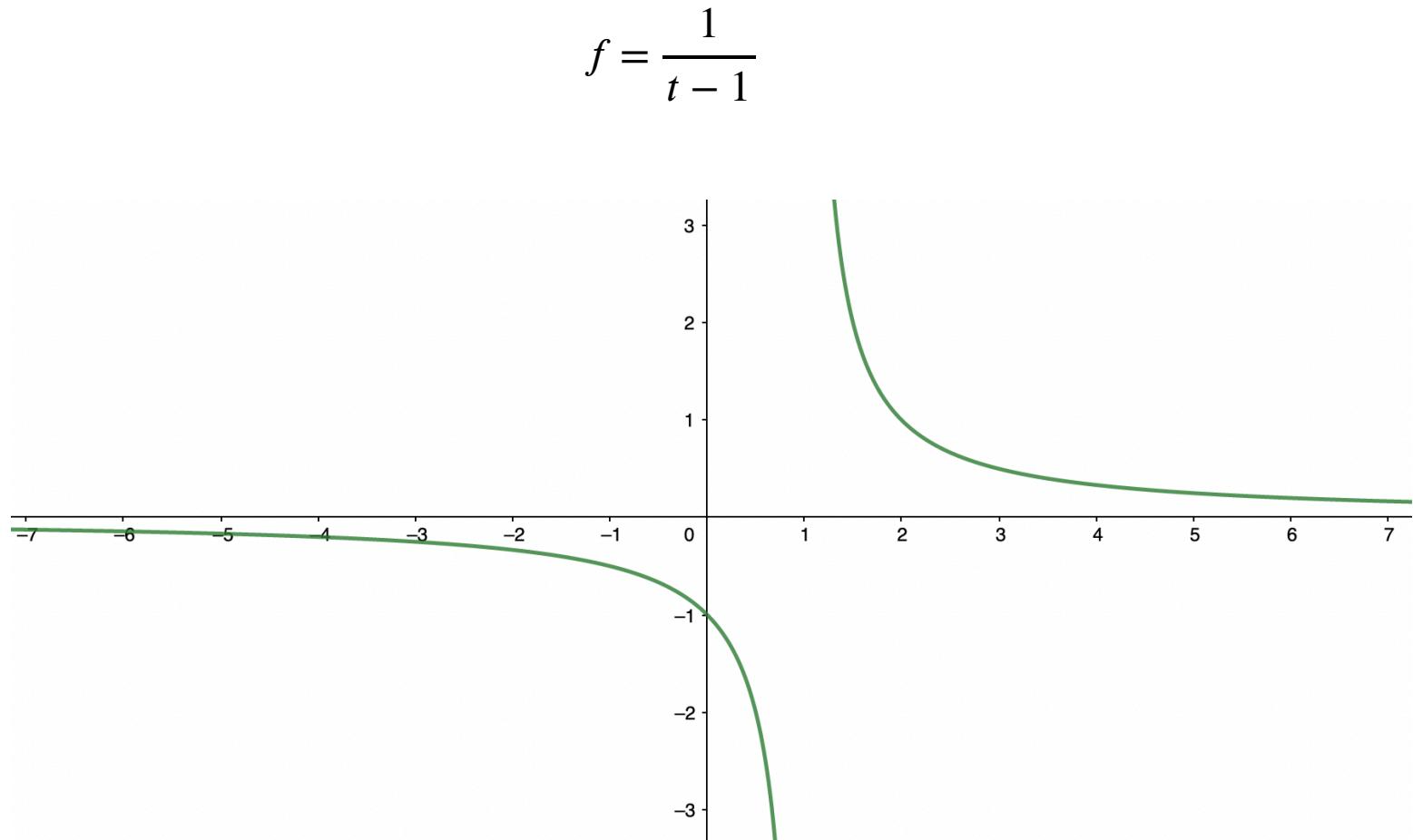
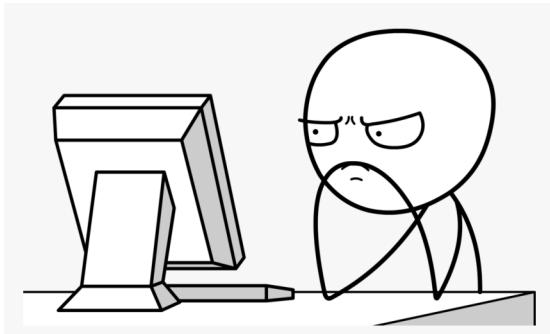


Programma la tua prima intelligenza artificiale

Alessandro Cossard

Introduzione

- Problema dei tre corpi
- Uso del **computer**
- NON è un corso di puro Python

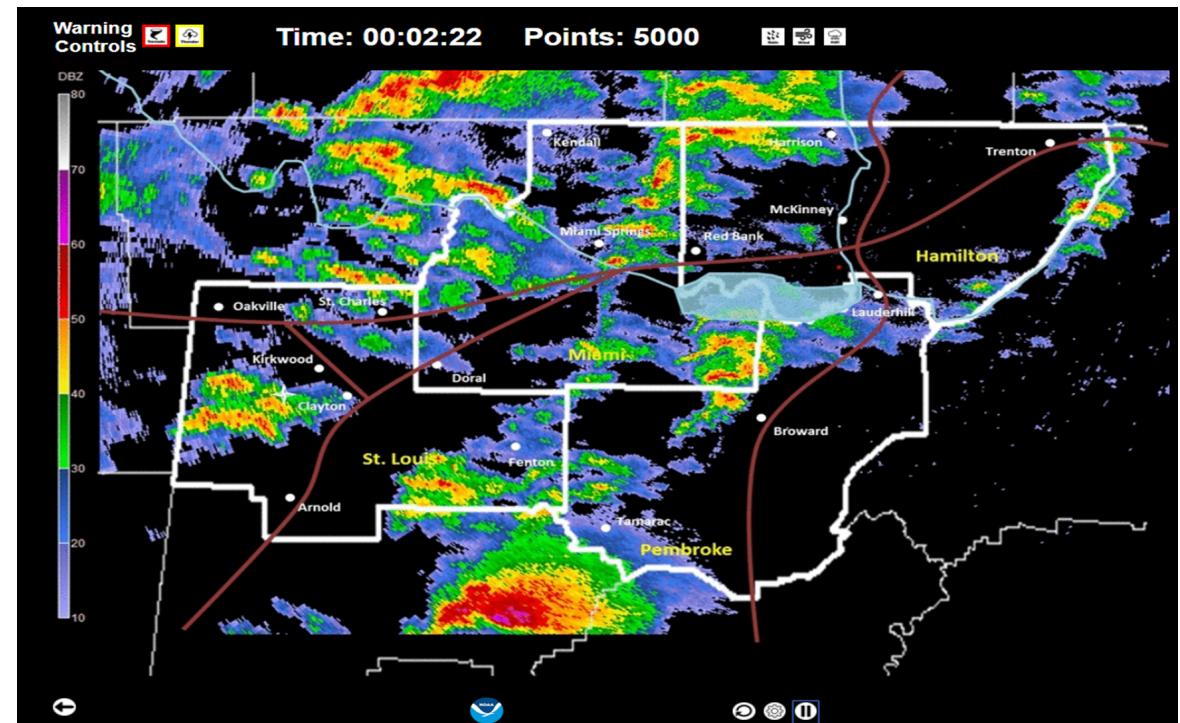


Modellizzazione

- **Descrivere** un sistema complesso
- Trovare gli aspetti significativi
- Adattarlo ad altri sistemi e non specifico
- Simulazioni al pc = programmare



Giorgio Parisi



Programmare

- Parte teorica
- Anaconda e Google Colab per tensorflow
- Notebook

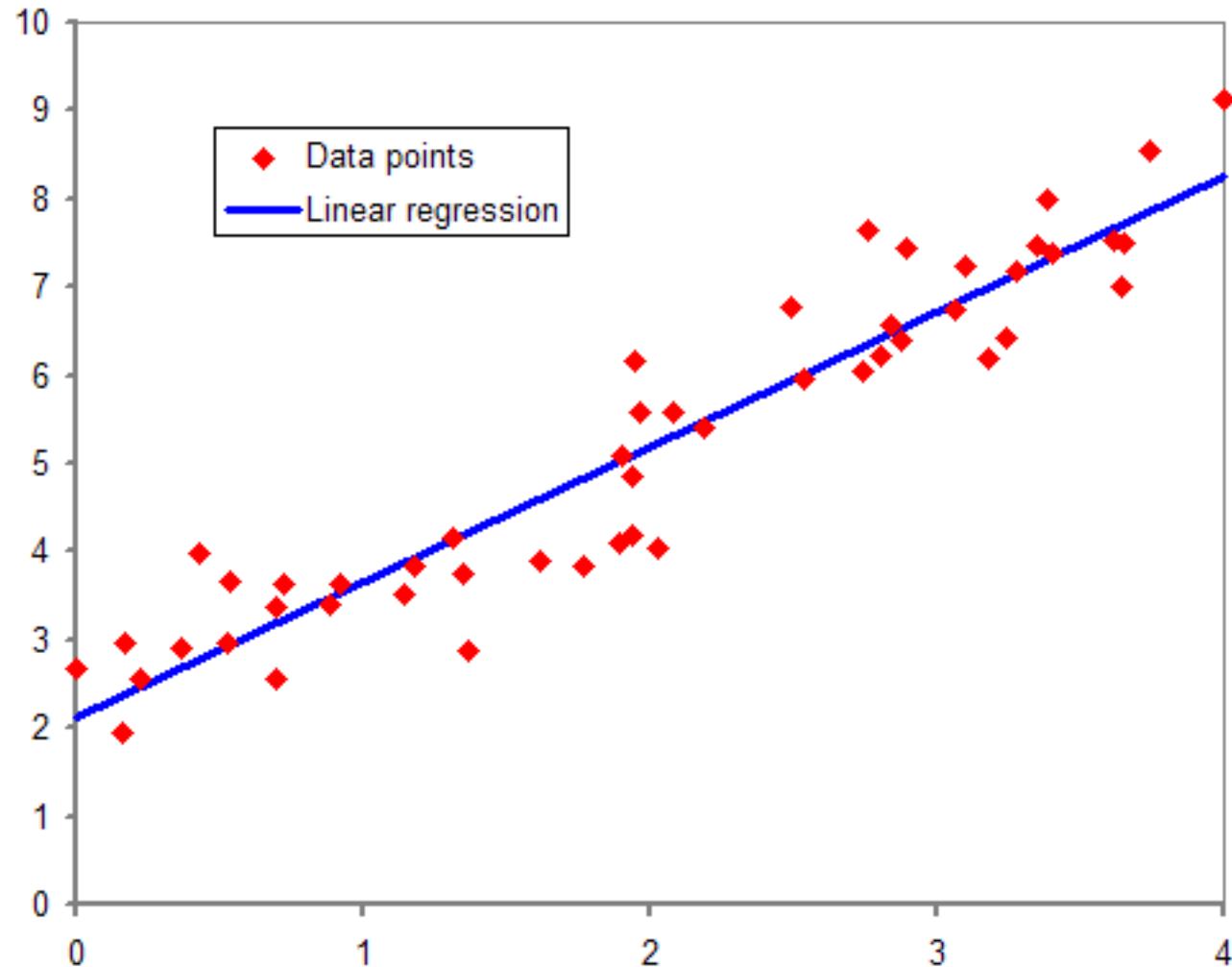


Data Science

- Machine Learning Vs Artificial Intelligence
 - Machine Learning: apprendimento automatico dai dati (predizione)
 - Usato **ovunque** (riconoscimento volti Facebook, suggeriti Spotify, importante l'applicazione in medicina)
 - Scopo: creare un modello **predittivo** generalizzabile e automatico
 - Fondamentale l'utilizzo e lo studio dei **dati**
-
- **Notebook**

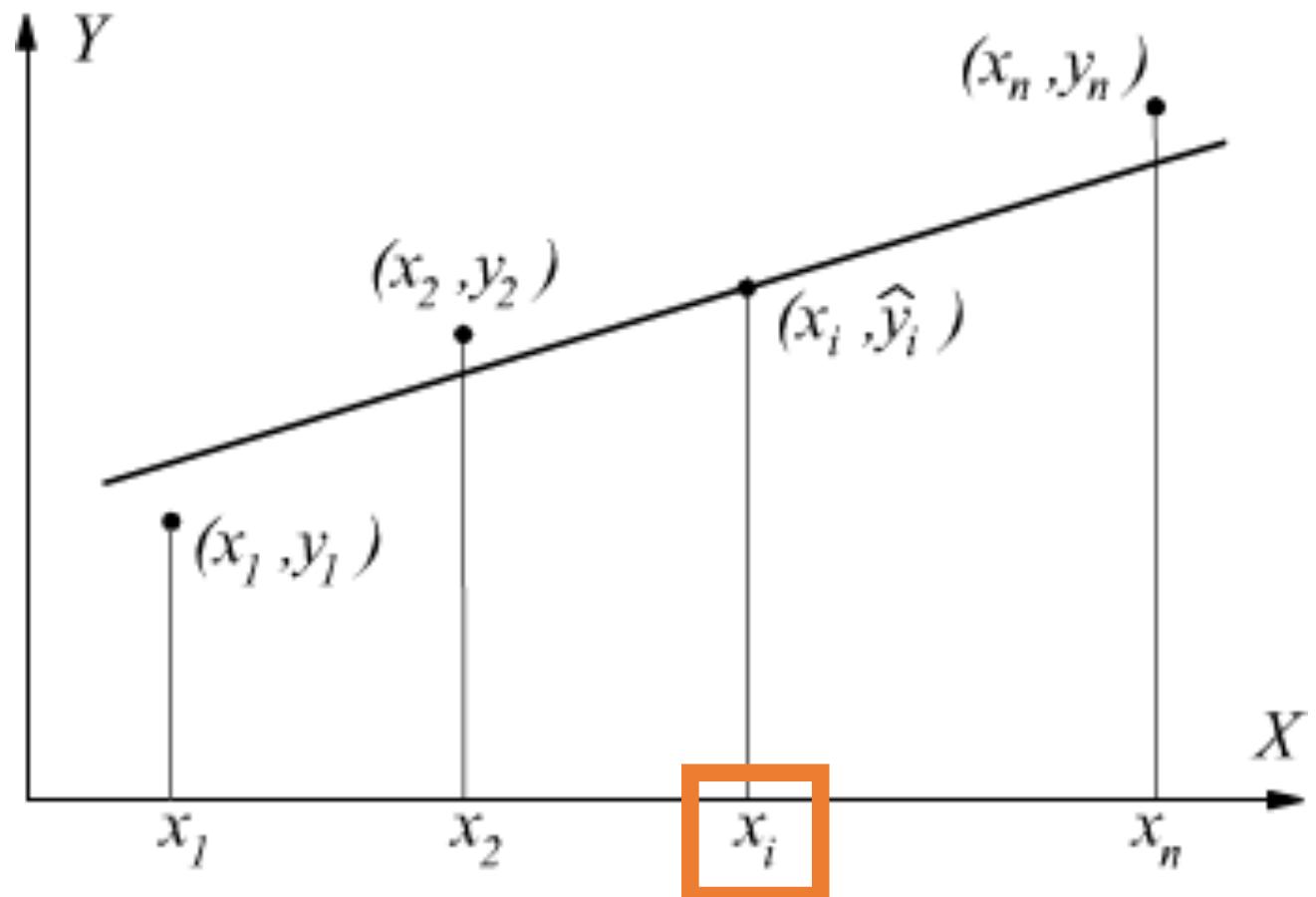
Regressione lineare

- Metodo **supervised**
- Regressione perché valore reale
- Quantificare la **dipendenza** tra due variabili



Regressione lineare

- Predire valori nuovi a partire da quelli noti
- Uso la retta di regressione come **predict**



Regressione lineare: l'algoritmo

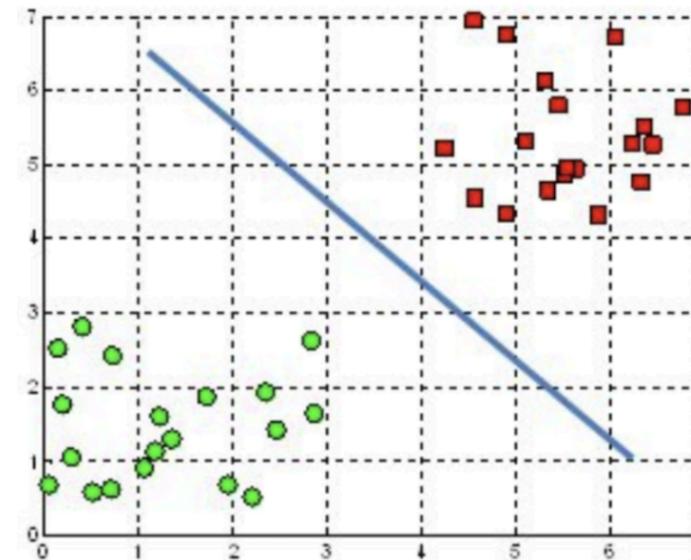
- $h(x) = \sum_{i=0}^d w_i x_i$ (NO sign())
- Quanto $h(x)$ **approssima** bene $f(x)$
- Uso l'errore quadratico
- Applico il metodo dei minimi quadrati
- Training Vs Test

$$E = \frac{1}{N} \sum_{i=1}^n [h(x_n) - y_n]^2 = \frac{1}{N} \sum_{i=1}^n \left[\sum_{i=0}^d w_i x_i - y_n \right]^2$$

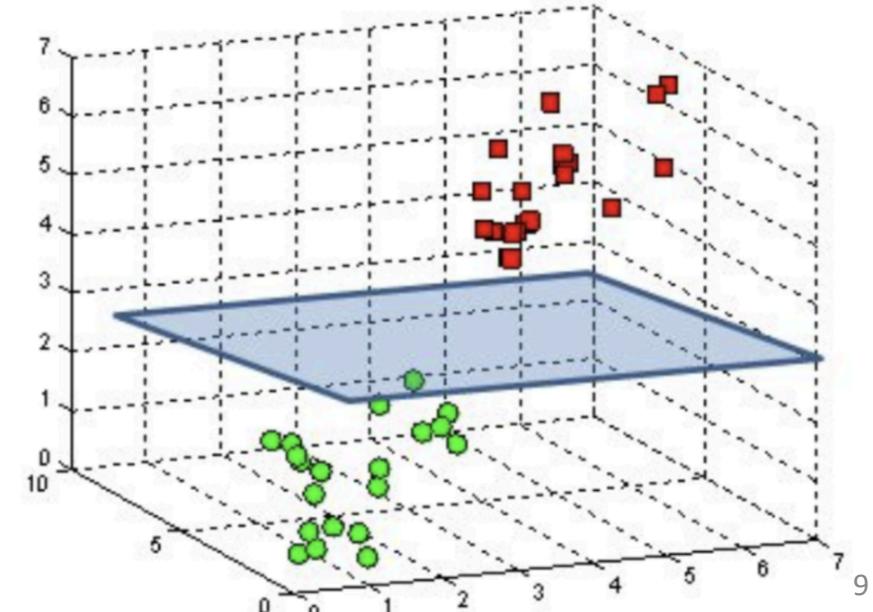
Statistica Multivariata

- La statistica multivariata guarda come cambiano più variabili in modo **indipendente**
- Quindi dopo i 3 assi non posso più fare i plots perchè è come se fossi in dimensioni maggiori a 3

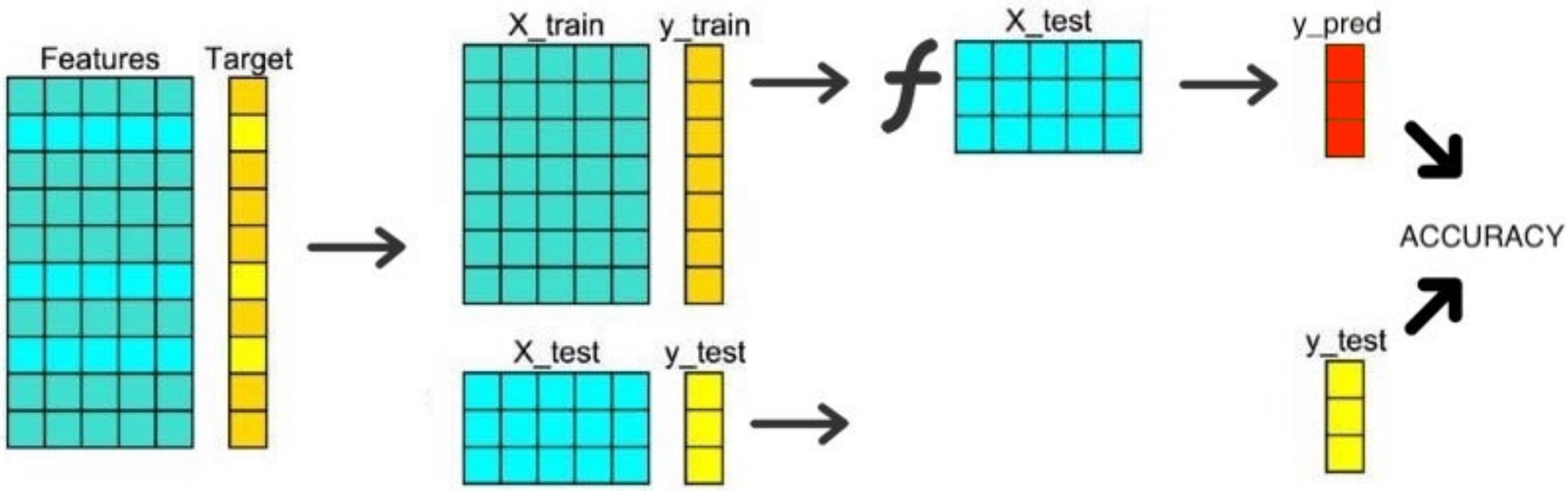
A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane

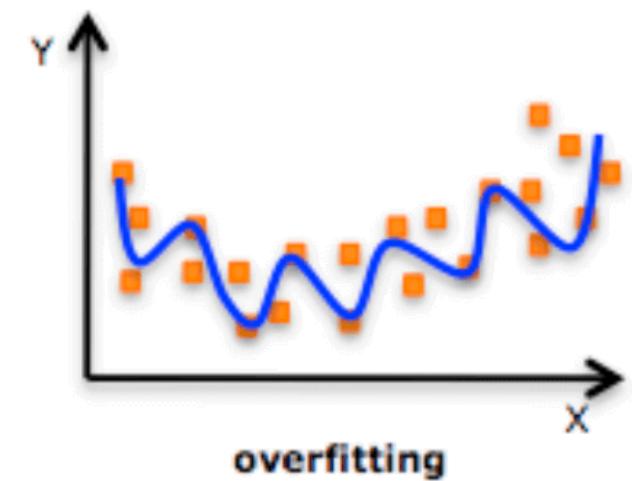
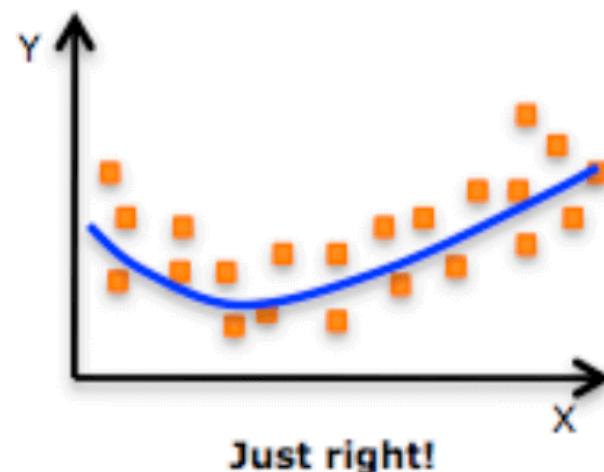


Regressione lineare - Train Vs Test



Overfitting

- Non voglio il 100% di accuracy perchè significa che il mio modello, classificando tutto correttamente, si è **adattato troppo ai dati**
- Quindi **perde** di generalizzazione nel caso di out-of-sample data



Esempio della banca

- Problema
- Dati
- Soluzione
- Banca

$$h(x) = \text{sign}\left(\sum_{i=0}^d w_i x_i\right)$$



Clienti	Casa	Macchina	Stipendio	Nucleo	Salute	Label
Cliente 1	si	si	15k	4	si	si
Cliente 2	no	Si	12k	1	si	si
Cliente 3	si	no	8k	3	si	no
Cliente N	no	si	16k	6	no	no

Perceptron

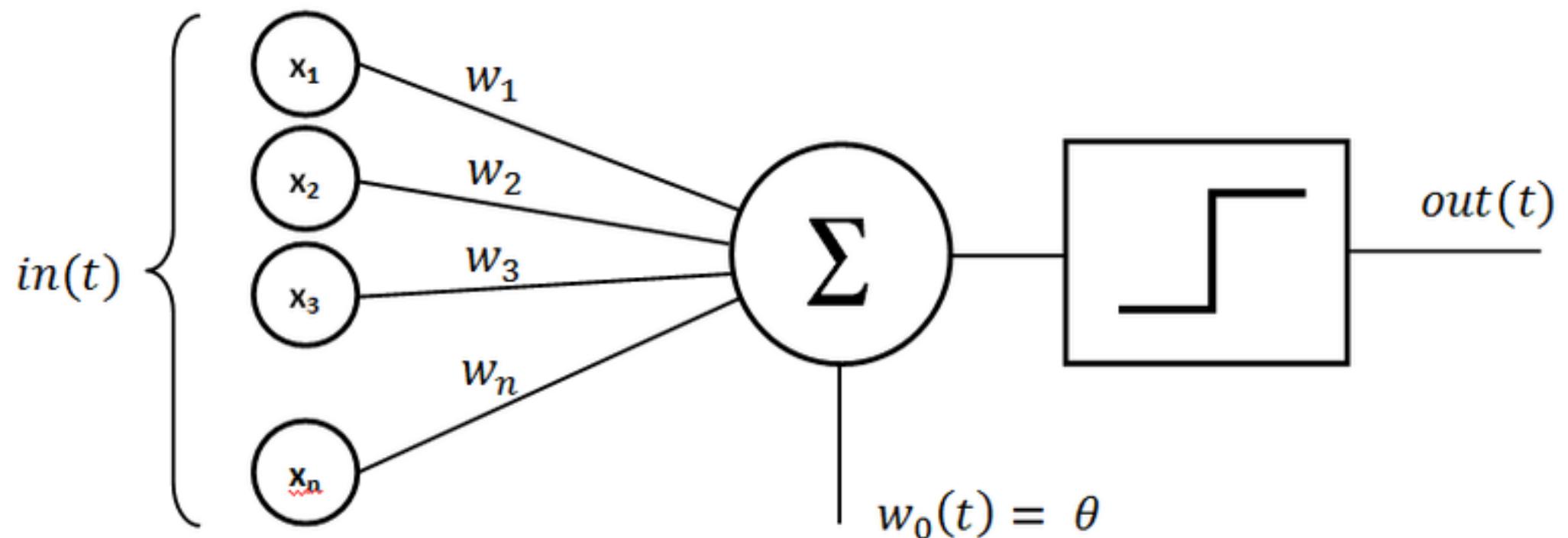
- $X = (x_1, \dots, x_n)$: n attributi di un cliente normalizzati tra 0 e 1

- Si al prestito se $\sum_{i=1}^n w_i x_i > 0$, NO al prestito se $\sum_{i=1}^n w_i x_i < 0$

- $Sign(\sum_{i=1}^n w_i x_i)$: + approvato, - respinto

Perceptron

- Neurone artificiale
- Input, pesi, somma, funzione di attivazione, output

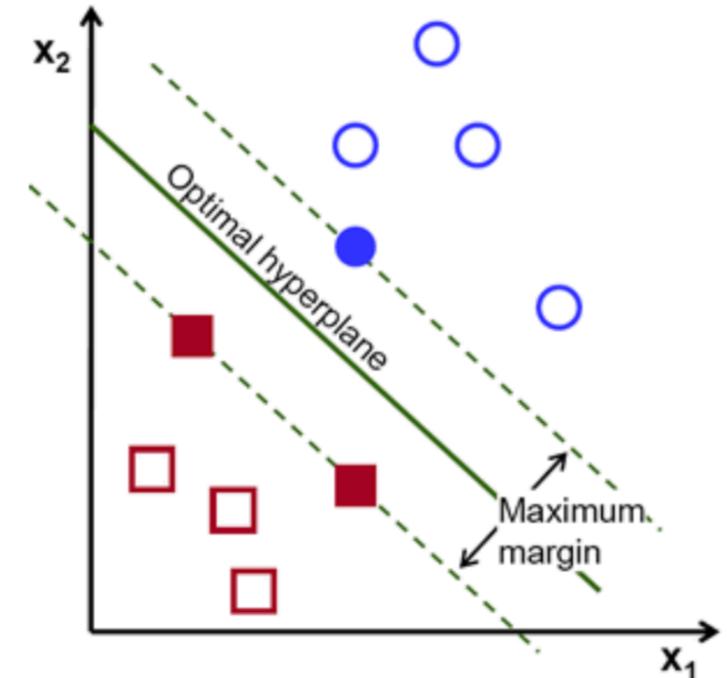
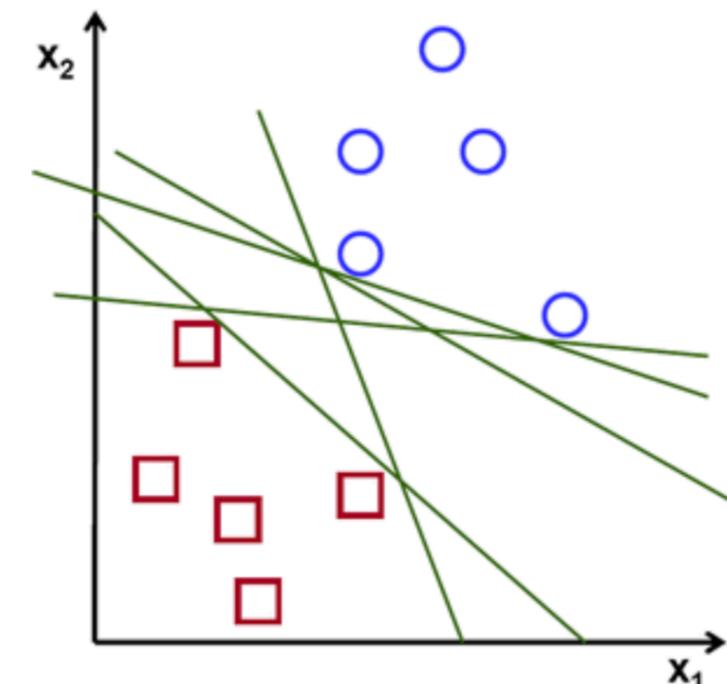


Scelta e Morale

- Neuromarketing
- Eyetracking: <https://www.webmarketingstrategico.net/eye-tracking/>
- https://en.wikipedia.org/wiki/Vance_Packard
- <https://www.snopes.com/fact-check/subliminal-advertising/>
- Morale?

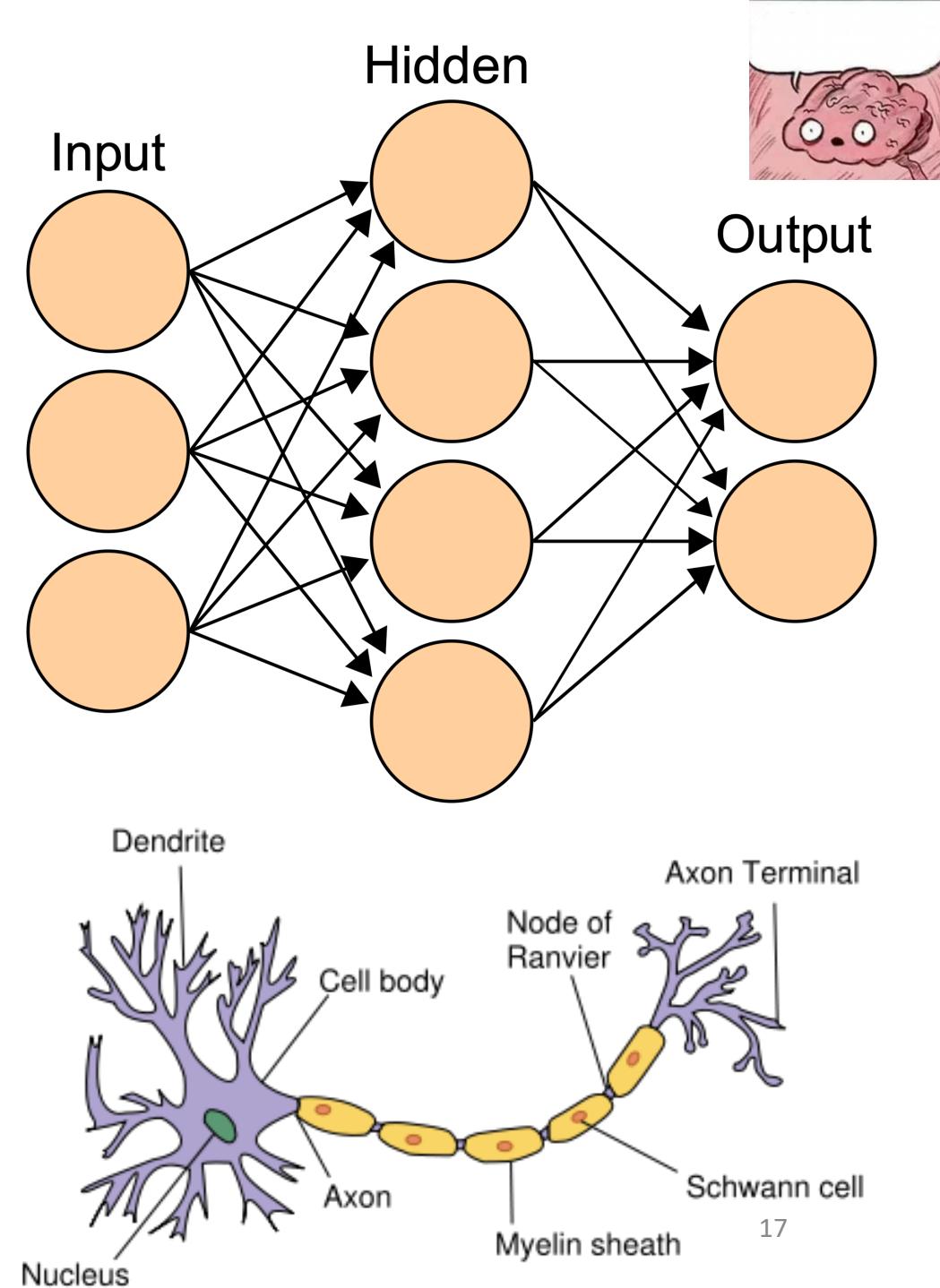
Support Vector Machine

- Simile al Perceptron MA
1) Non richiede la separabilità dei dati perchè li **proietta** usando il Kernel
2) Al perceptron basta un piano qualunque mentre SVM cerca i **Support Vectors** per selezionarlo



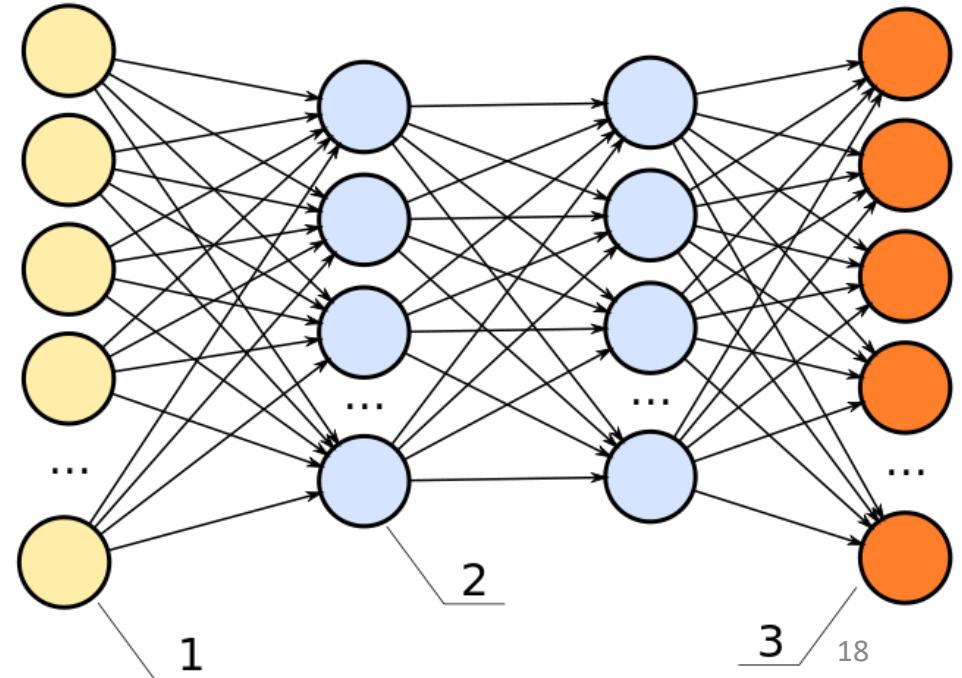
Rete Neurale

- Multilayer perceptron
- Sistema di apprendimento formato da diverse sottocomponenti, alcune di input, altre di output e altre ancora nascoste (comunicative)
- Simile al cervello umano: se il segnale dei dendriti supera una certa soglia, il neurone emette un segnale elettrico
- La rete va allenata modificando i pesi (Black Box)



Deep Neural Network

- Almeno due o tre **hidden layers**
- Risultati migliori ma più difficile da gestire
- Supervised: posso controllare il risultato



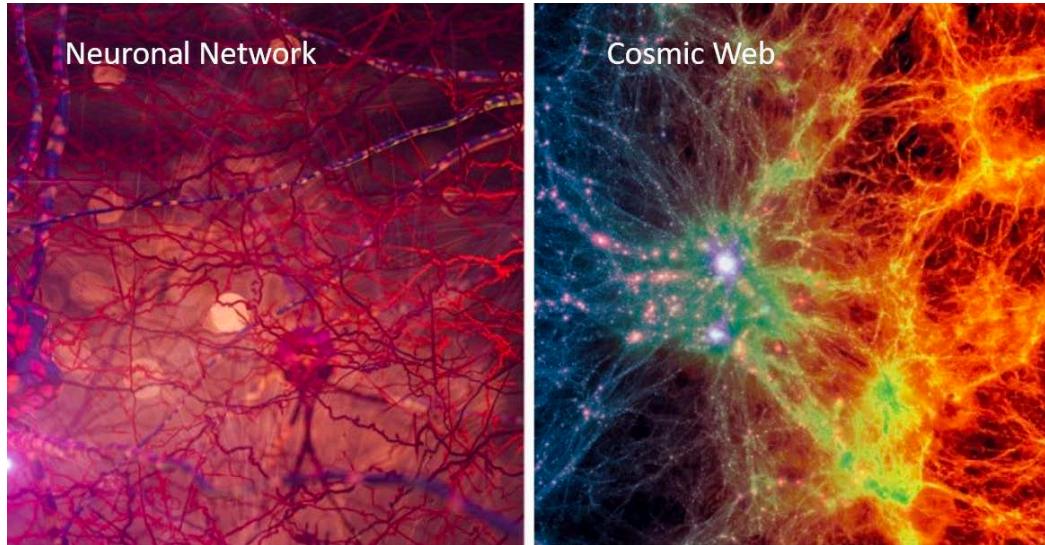
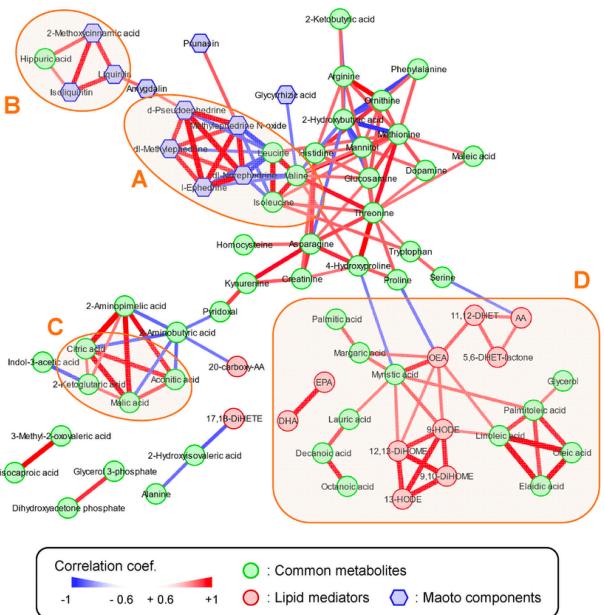
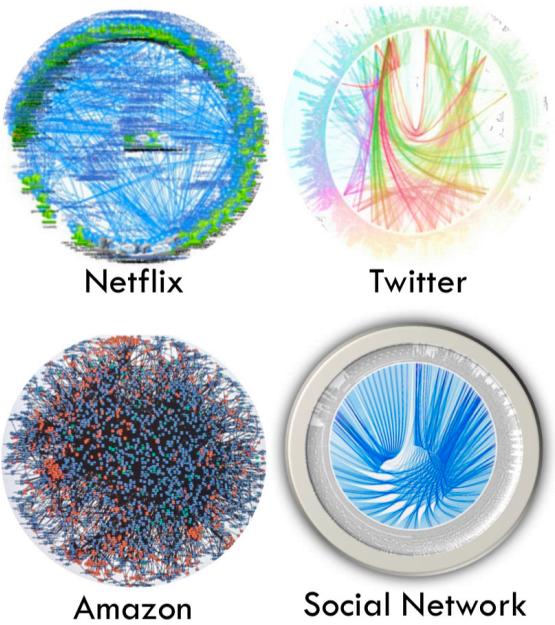
Backpropagation Algorithm

- Reti a Backpropagation Algorithm:
 1. Inizializzazione pesi random
 2. Fase feed forward
 3. Confronto labels (supervised)
 4. Valutazione dell'errore
 5. Aggiornamento dei pesi
 6. Repeat
- L'algoritmo è interrotto quando l'errore è più piccolo della soglia desiderata
- Backpropagation = propaga indietro l'errore per aggiustare i pesi
- Vado avanti così finche il la rete non classifica bene e poi test set

Classificazione di Iris

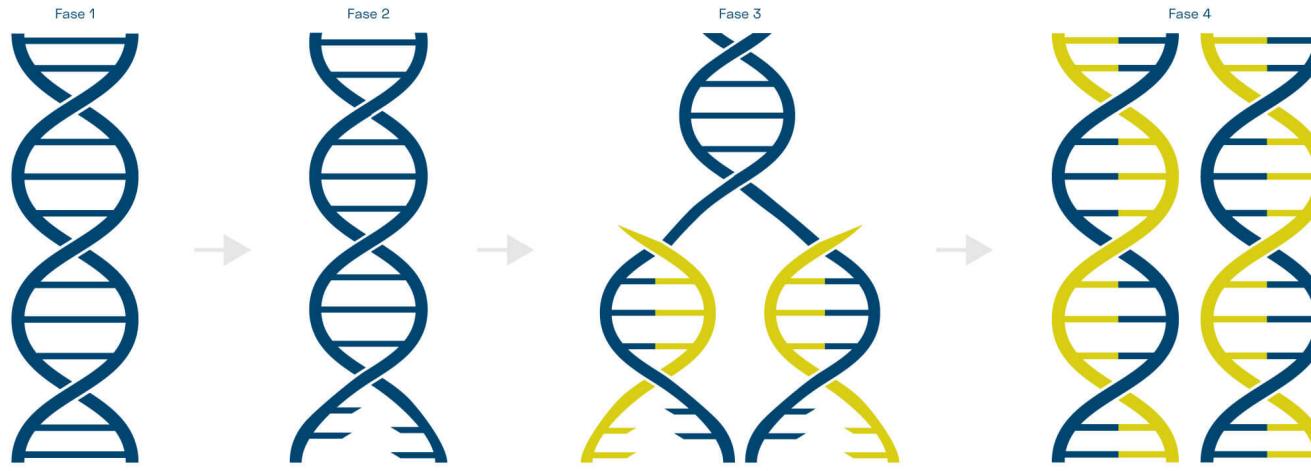
- Tensorflow e Keras: occupano un sacco di spazio sul PC
- Upload del notebook su Colab
- Esercizio: tunare il modello e studiare l'andamento dell'accuracy
- Notebook

Complex systems



What is Life?

- “*For otherwise the number of co-operating particles would be too small, the 'law' too inaccurate. The particularly exigent demand is the square root. For though a million is a reasonably large number, an accuracy of just 1 in 1,000 is not overwhelmingly good, If a thing claims the dignity of being a 'Law of Nature'"*



DNA
Processo di duplicazione semiconservativo



Ordine di Scala

- Concetti dei gas perfetti alla **larga scala**
- Particelle alla **microscala**
- Cosa succede quando passo da una all'altra?



Comportamento Emergente



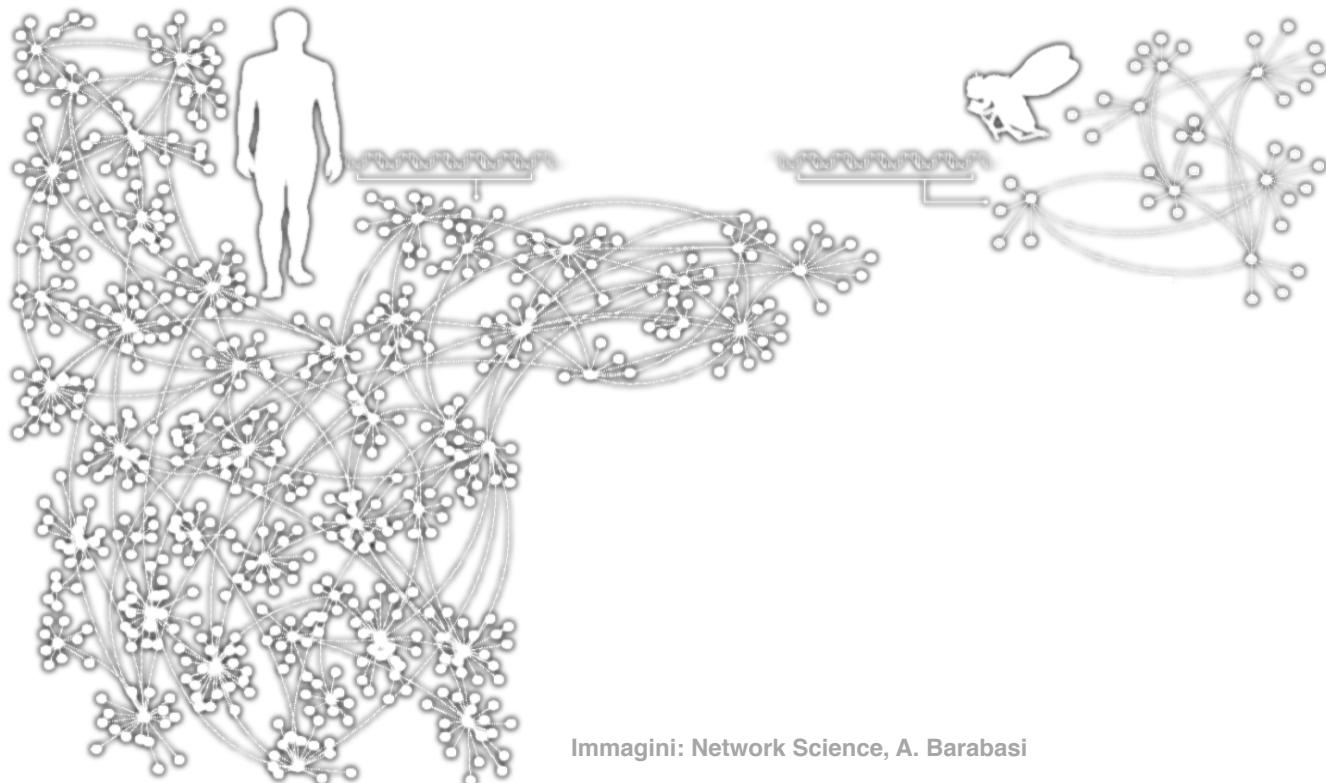
Alessandro Cossard

Interazioni non lineari



Reti

- Dietro ogni sistema complesso c'è una **rete** che ne descrive il comportamento

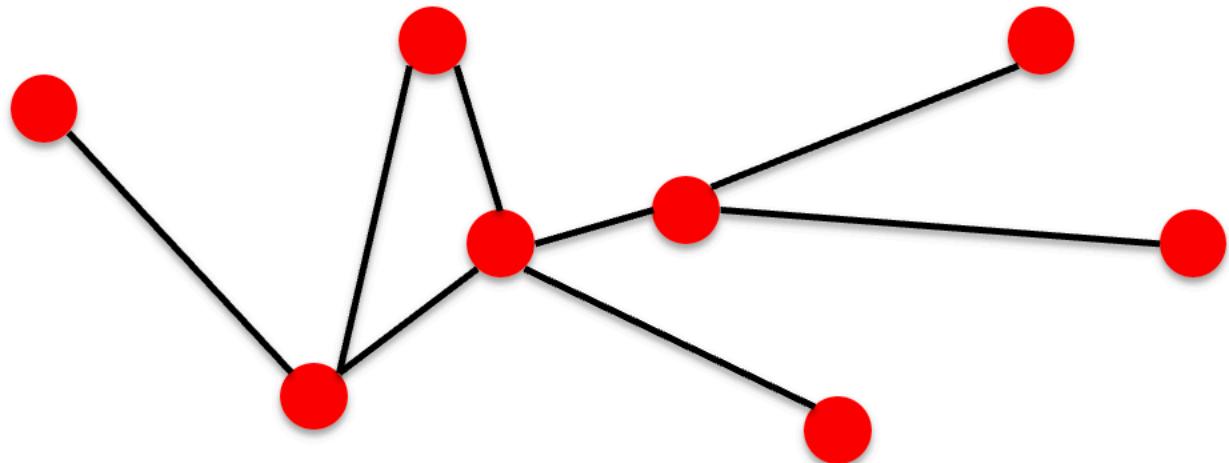


Immagini: Network Science, A. Barabasi



Reti

- Definizioni di base



- **components:** nodes, vertices

N

- **interactions:** links, edges

L

- **system:** network, graph

(N,L)

Network e Grafi

Le reti spesso si riferiscono a sistemi reali refers to real systems.

- www e internet
- social network
- Reti di protein e metaboliche

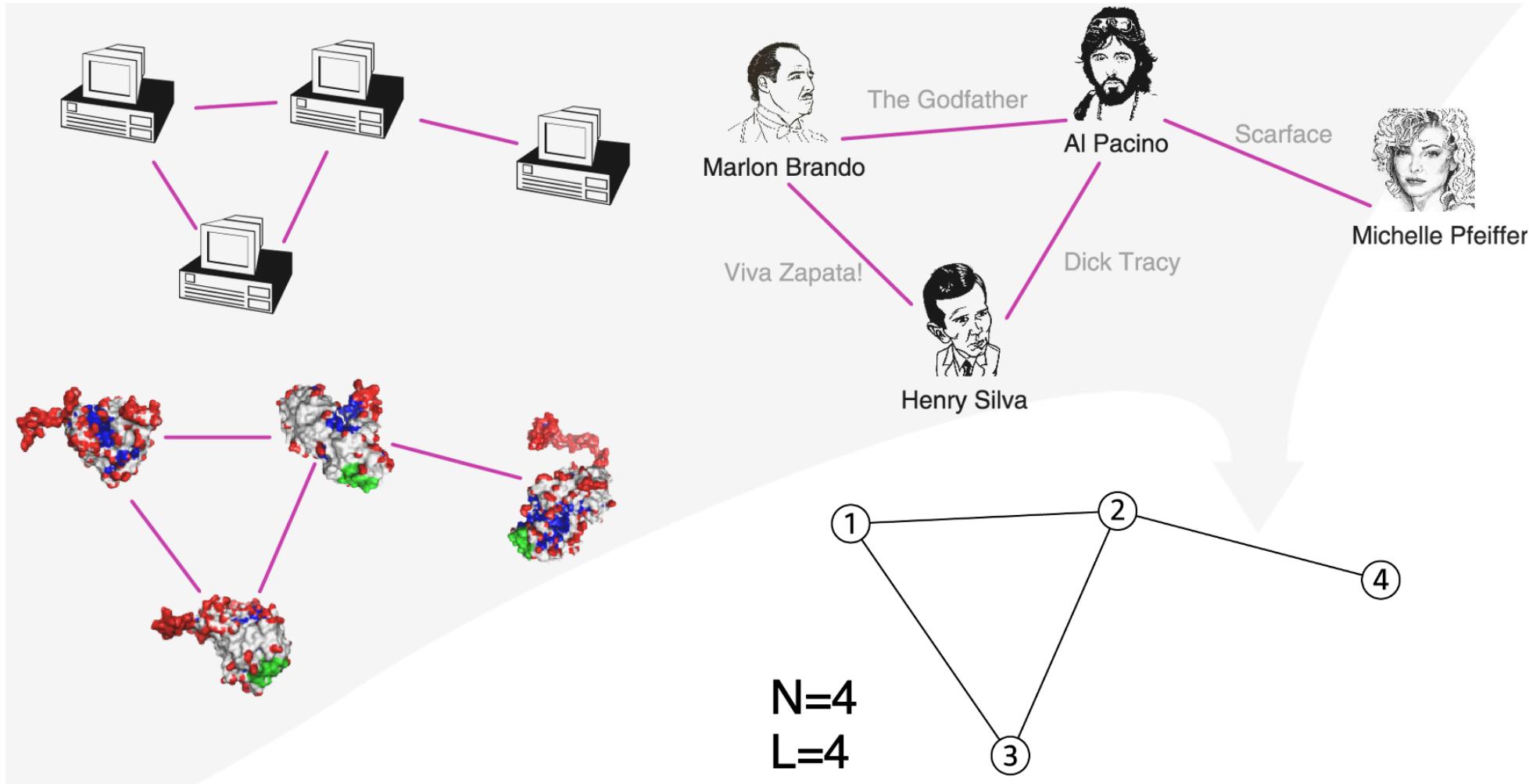
Termini: (Network, node, link)

I grafi sono la rappresentazione matematica di una rete.

- web graph

Termini: (Graph, vertex, edge)

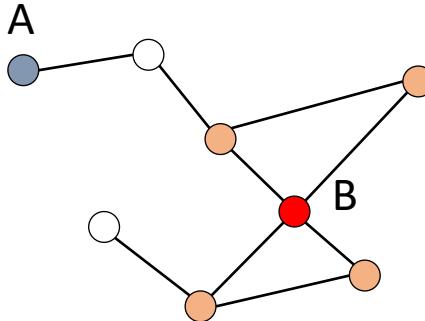
Uno stesso linguaggio



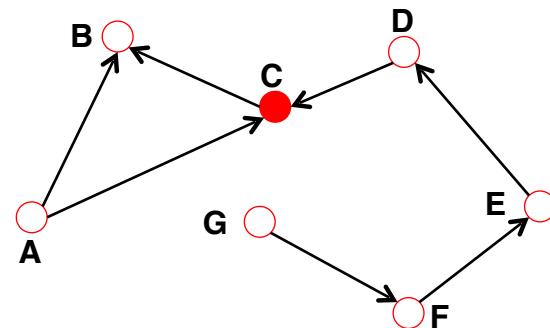
Definizioni di base

Grado di un nodo: il numero di links connessi a tale nodo.

Indiretta



Diretta

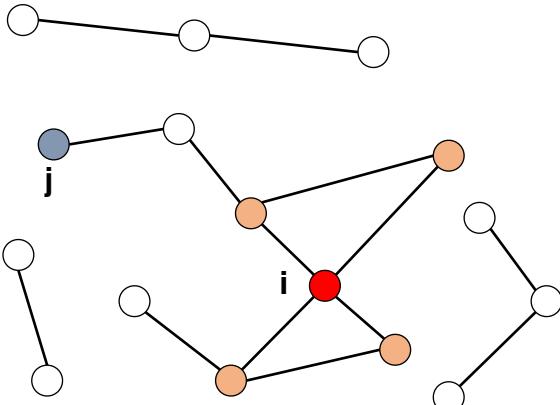


Nelle reti dirette definisco l'**in-degree** e l' **out-degree**. Il grado (totale) sarà la somma di in e out degrees.



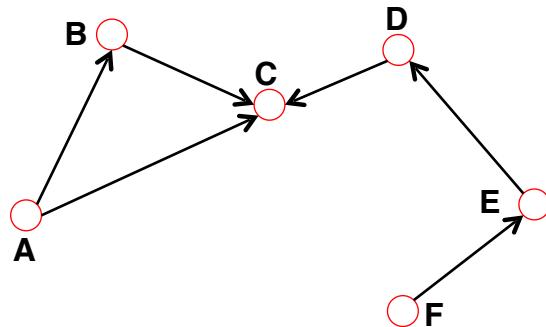
Definizioni di base

Indiretta



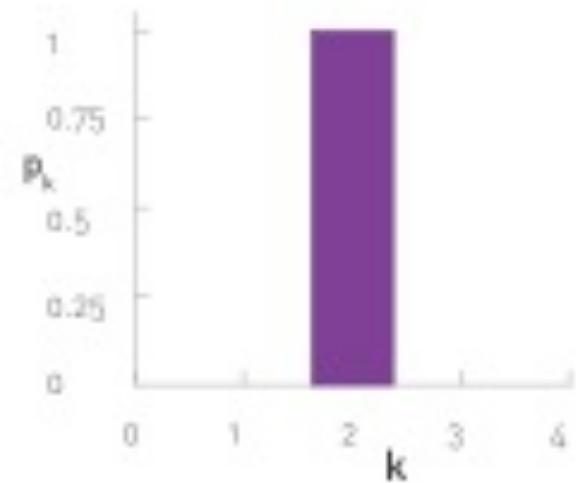
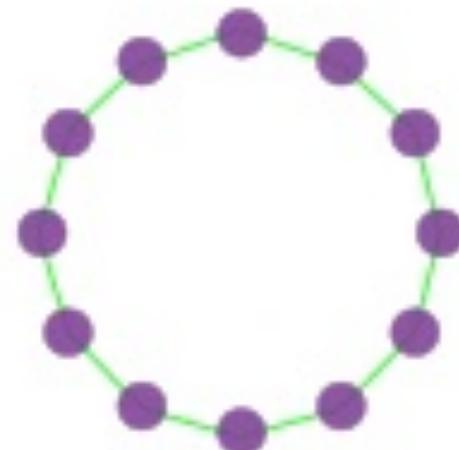
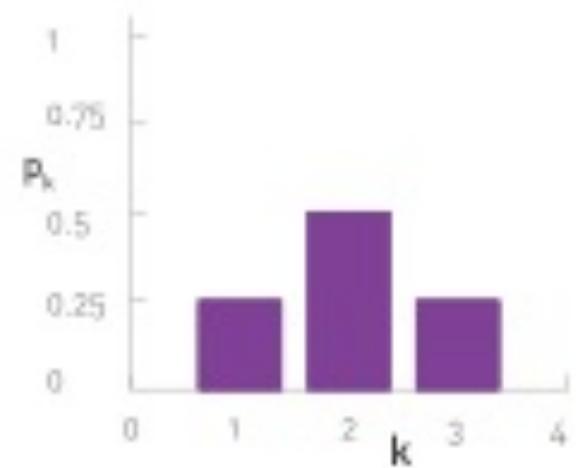
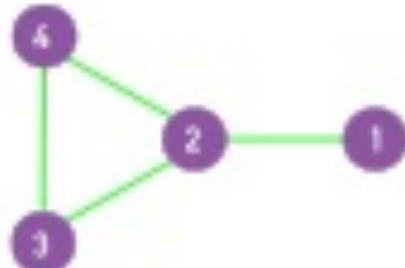
N – the number of nodes in the graph

Diretta



Distribuzione di Grado

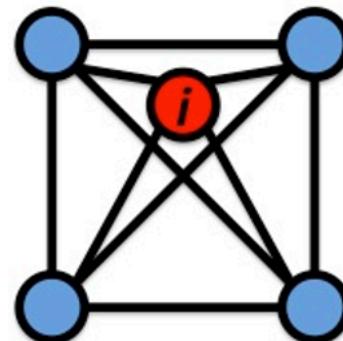
- **Degree Distribution:** in ascissa il grado, in ordinata il numero di nodi con quel grado
- Normalizzando per il numero di nodi ottengo una distribuzione di probabilità



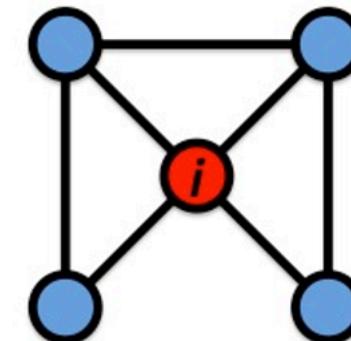
Coefficiente di Cluster

- Fornisce un'indicazione di quanto sono legati i **vicini** del nodo

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

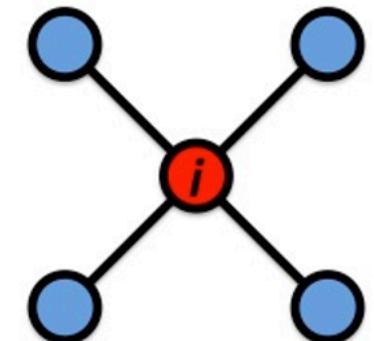


$$C_i = 1$$



$$C_i = 1/2$$

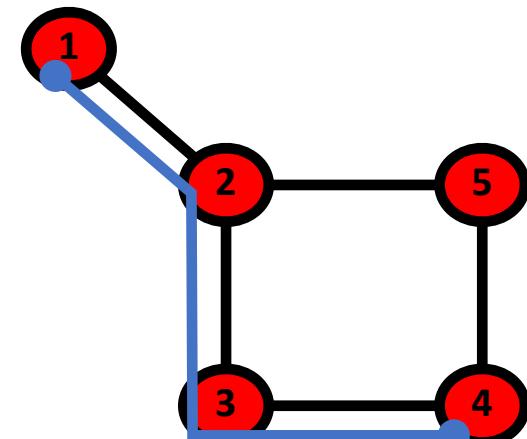
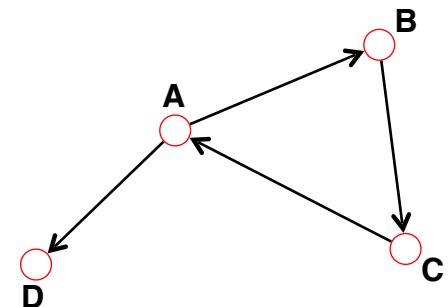
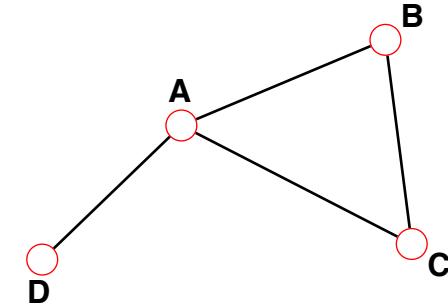
$$(2*3) / (4*3)$$



$$C_i = 0$$

Distanze nelle reti

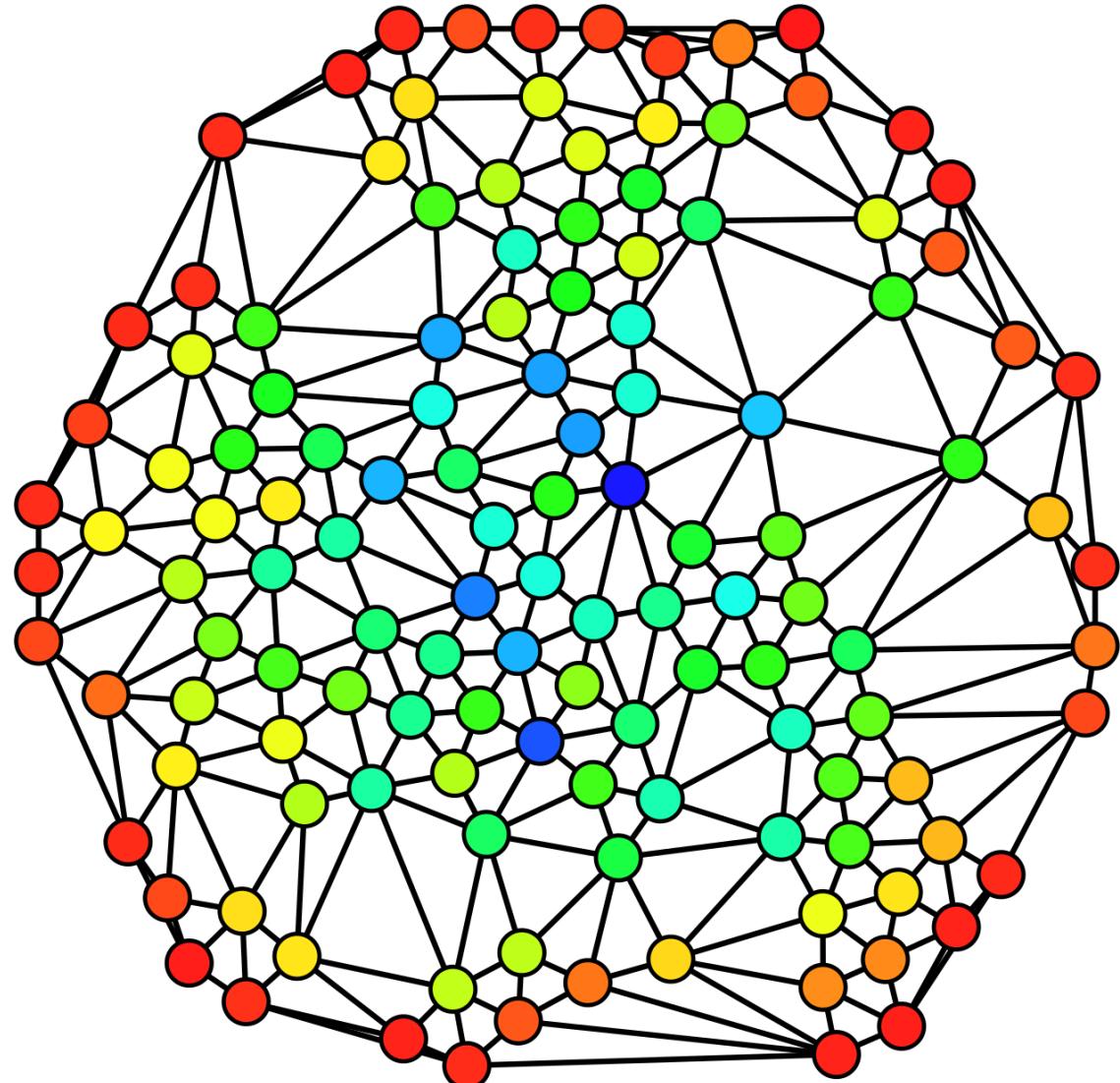
- La **distanza** tra due nodi è definita come il numero di links lungo il cammino più breve che li unisce. Se due nodi sono scollegati, la distanza è infinita.
- Nei grafi **diretti**, il cammino segue il verso della freccia. Quindi la distanza tra due nodi A e B può essere diversa dalla distanza tra B e A
- Il **diametro** è la distanza massima tra due nodi qualsiasi del grafo



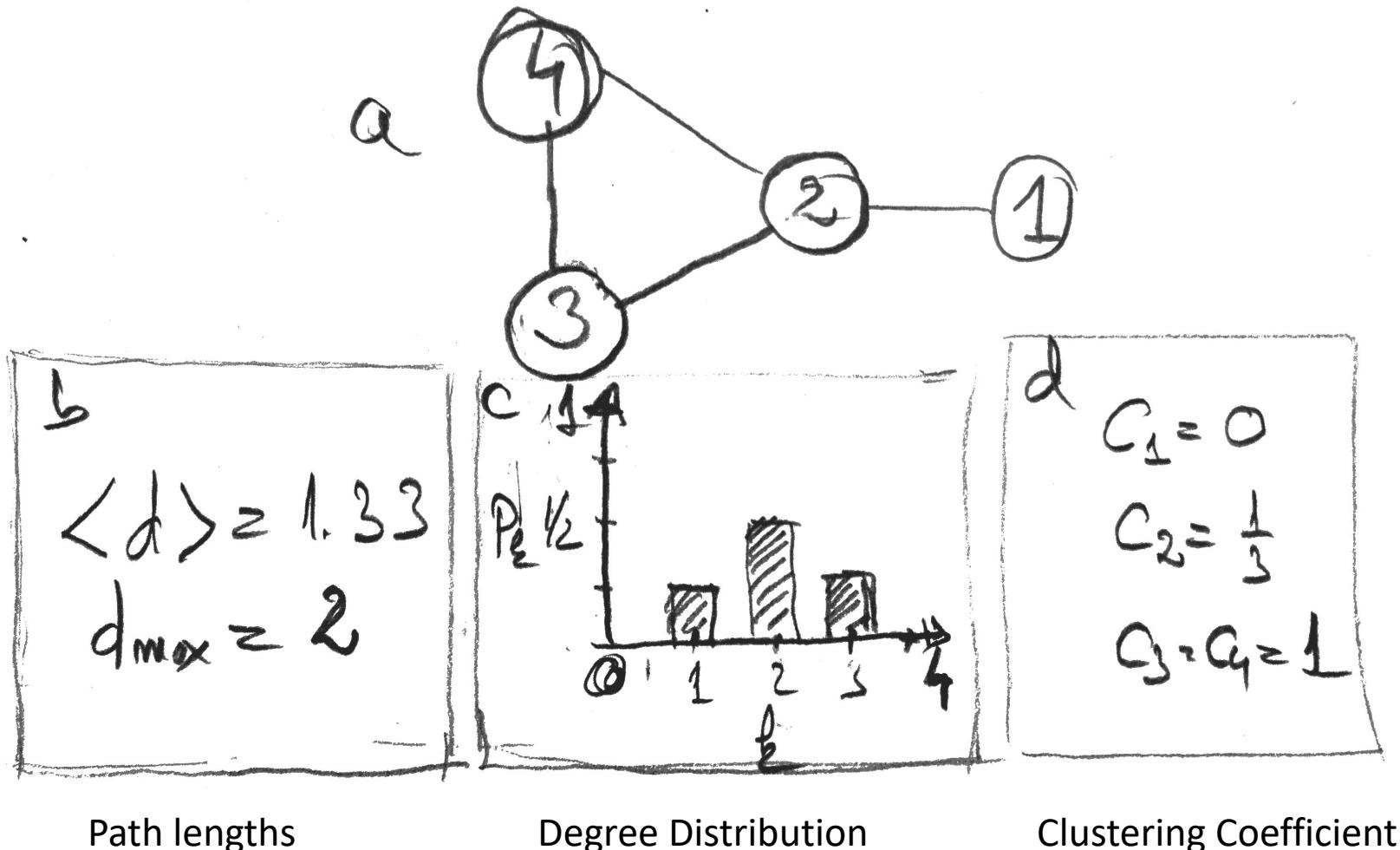
Betweenness centrality

- Un modo per indicare quanto un nodo è **centrale** nella rete

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$



Tre quantità fondamentali



Path lengths

Degree Distribution

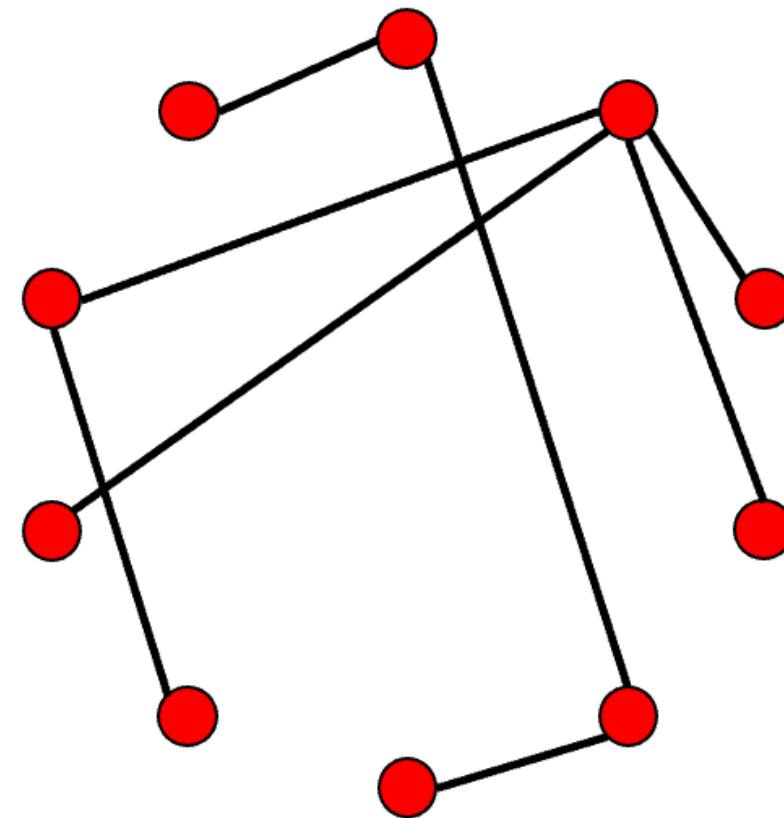
Clustering Coefficient

Random Network

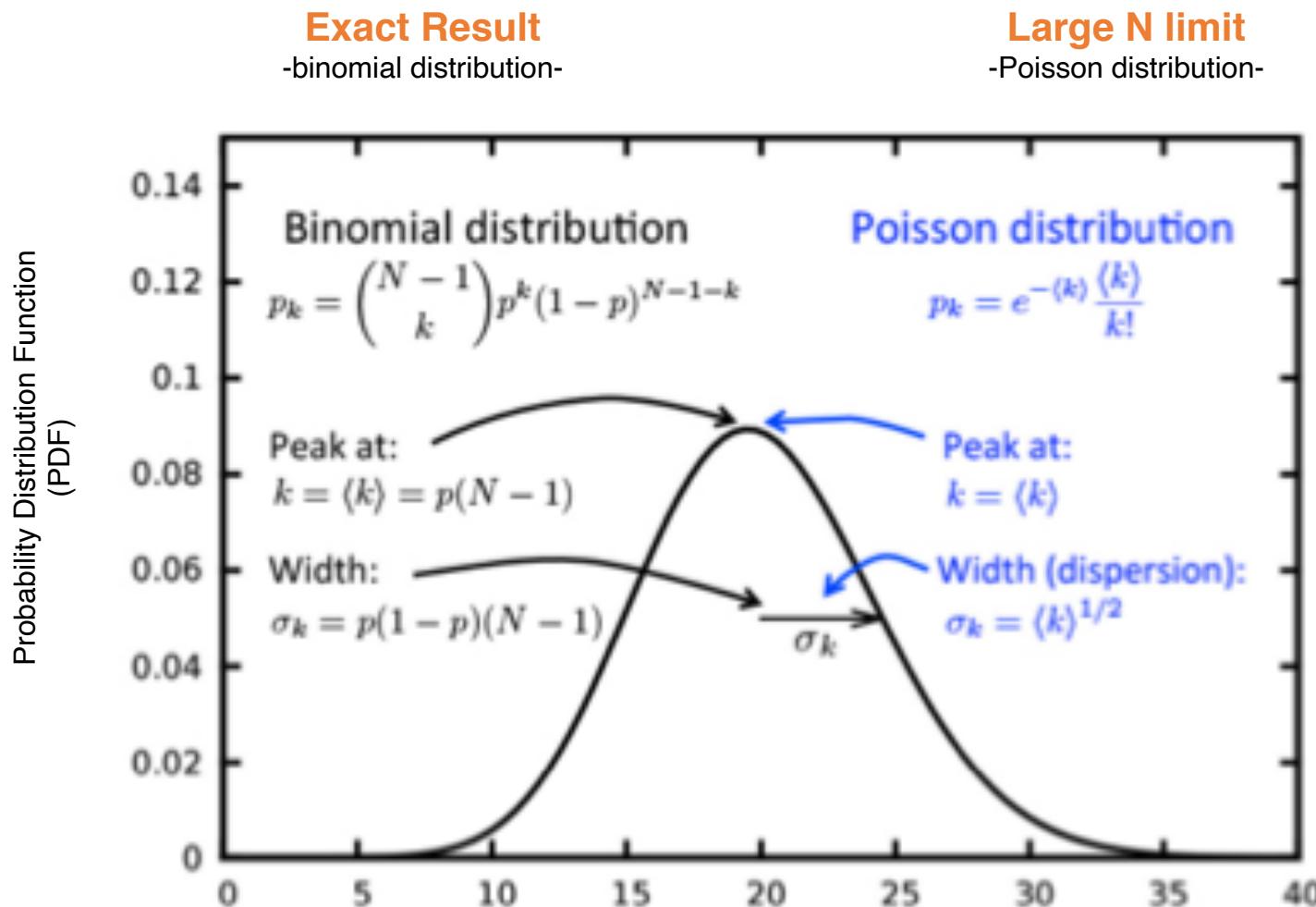
- Una Rete Random $G(N,P)$ è un grafo di N nodi in cui ogni coppia di nodi è connessa con probabilità p

$p=1/6$ $N=10$

$\langle k \rangle \sim 1.5$

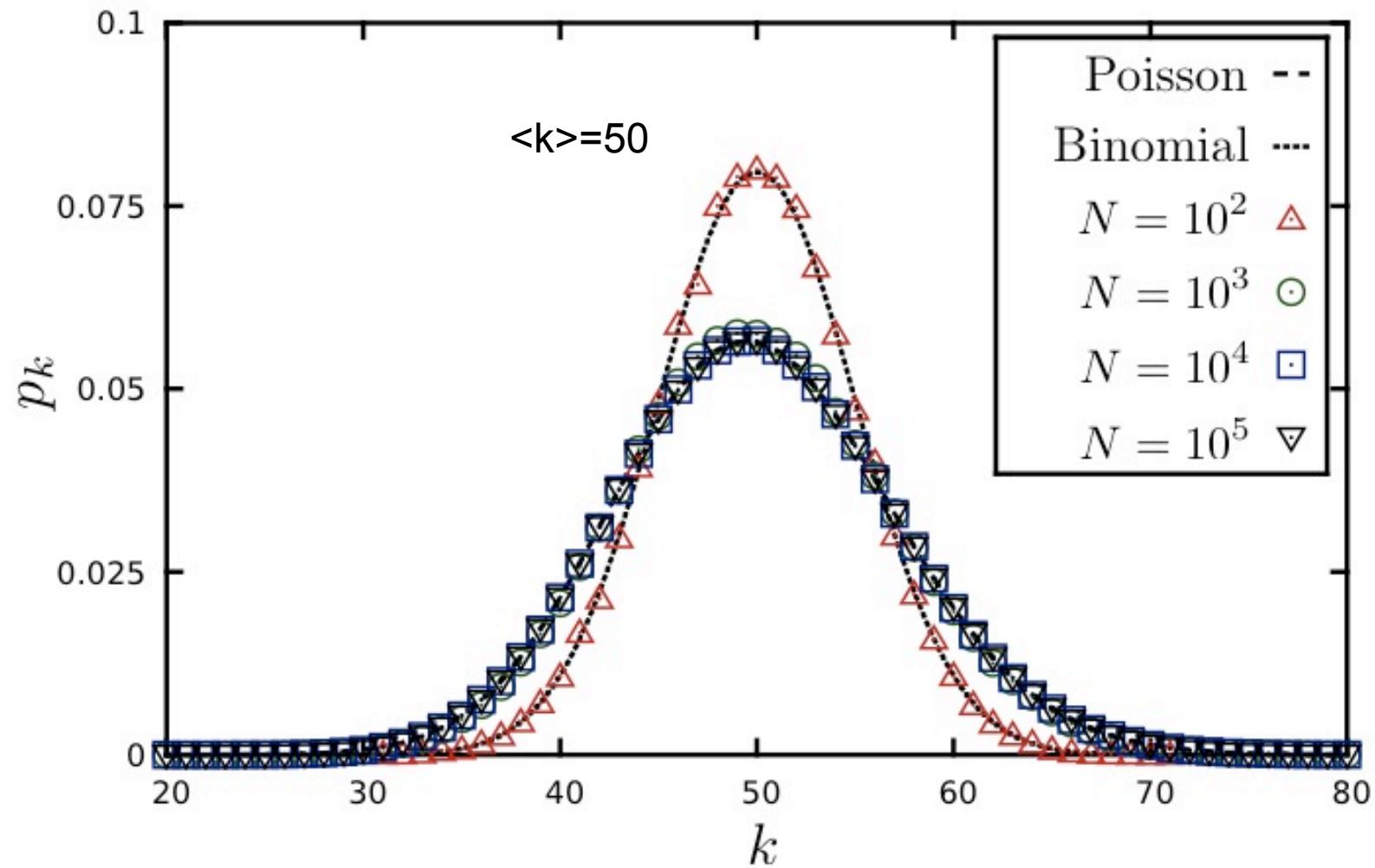


Degree Distribution

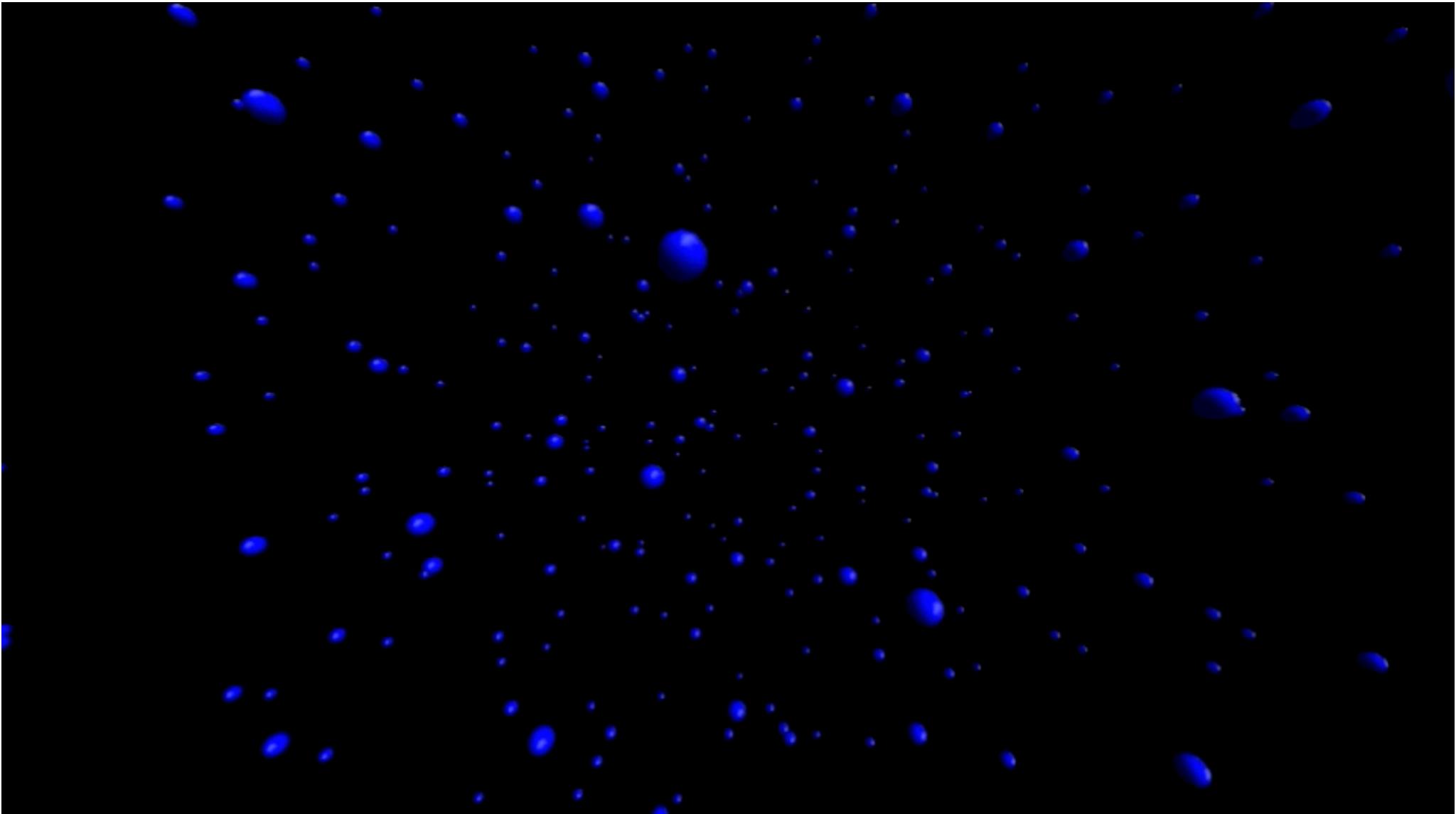


Degree Distribution

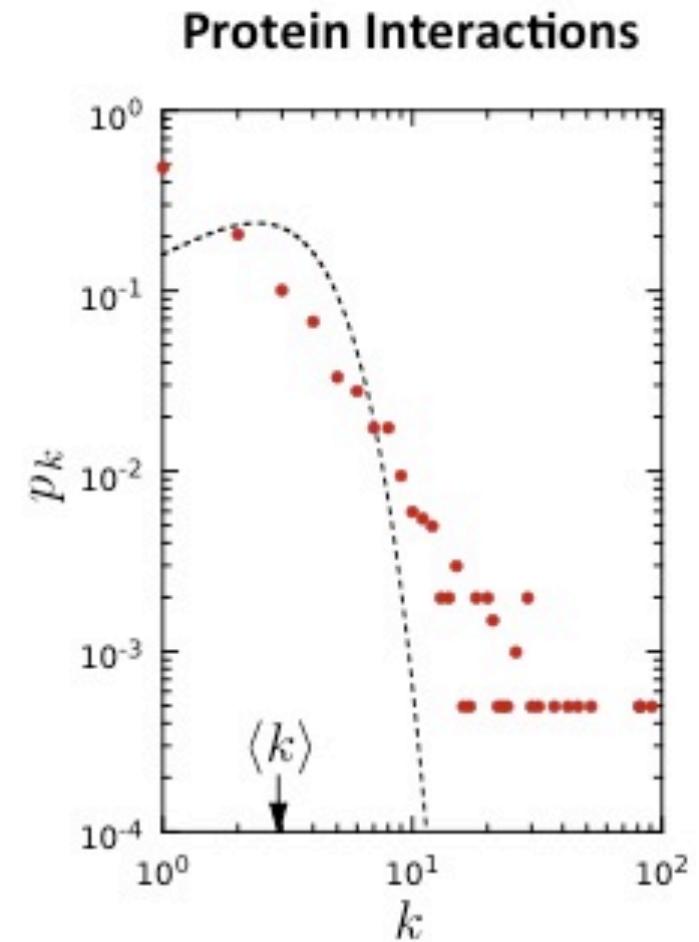
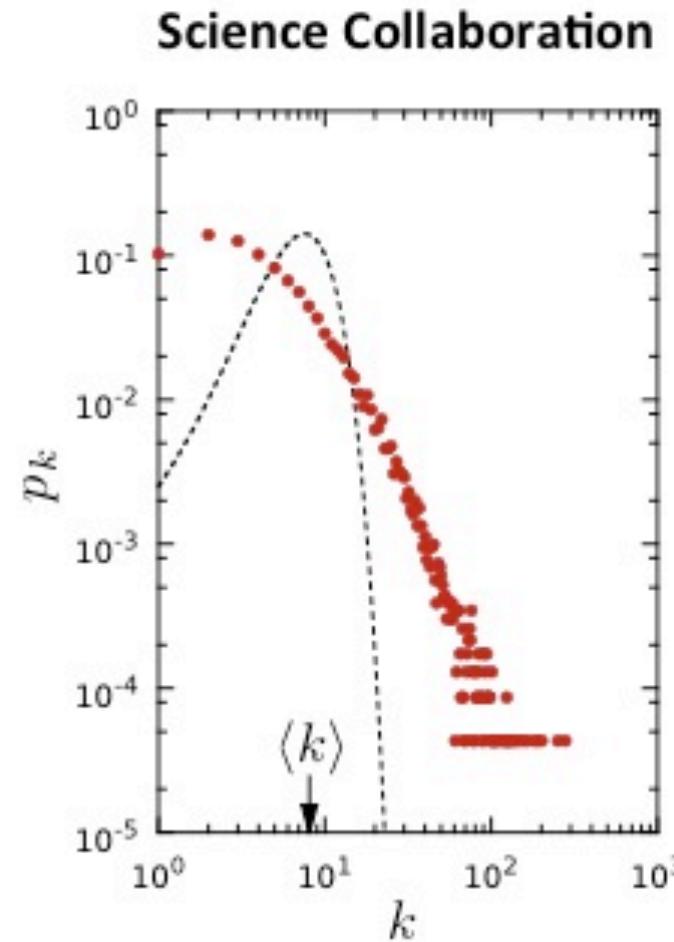
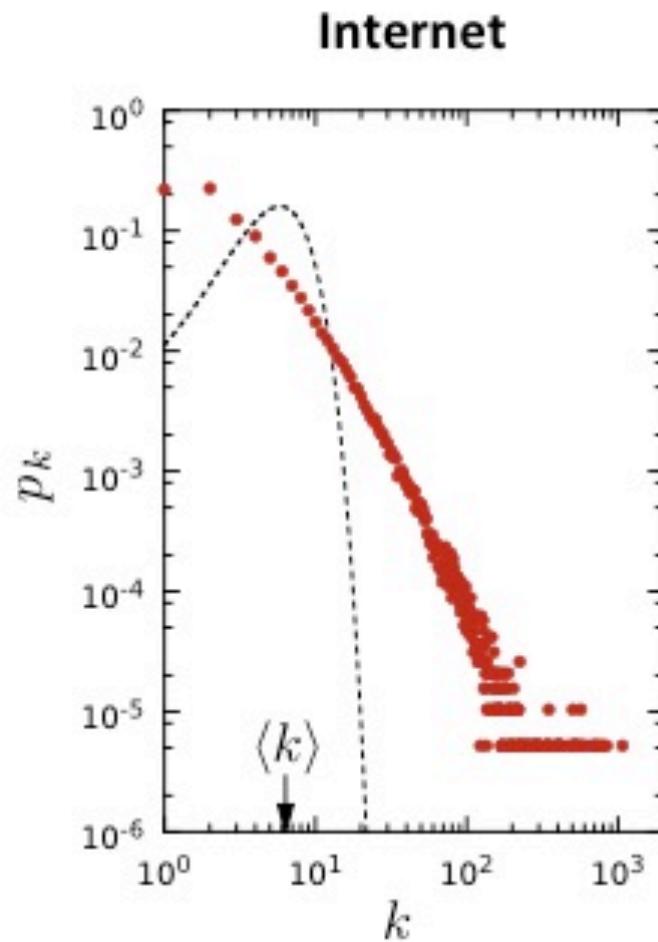
- tutti i nodi hanno più o meno lo stesso numero di nodi:
irrealistico!



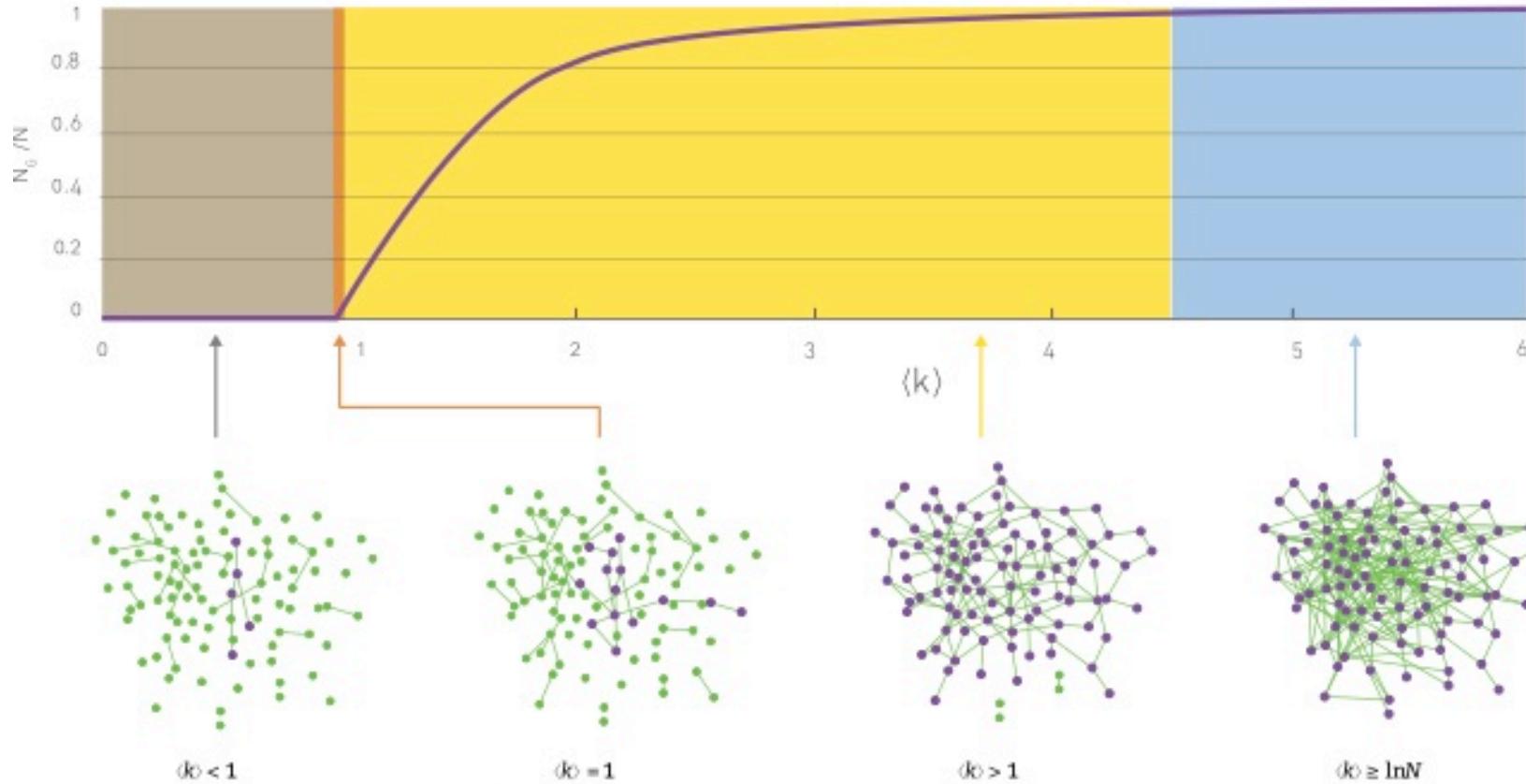
Evoluzione di una Rete Random



Le reti reali NON sono Random



Le reti reali sono Supercritiche



- (b) Subcritical Regime**
- * No giant component
 - * Cluster size distribution: $p_s \sim s^{-3/2} e^{-\alpha s}$
 - * Size of the largest cluster: $N_G \sim \ln N$
 - * The clusters are trees

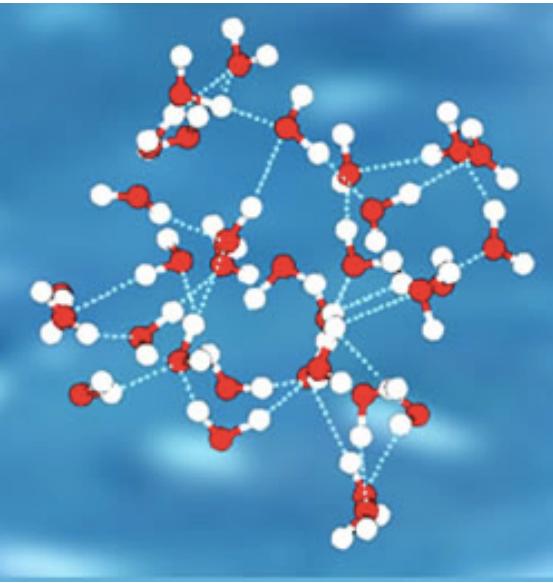
- (c) Critical Point**
- * No giant component
 - * Cluster size distribution: $p_s \sim s^{-5/2}$
 - * Size of the largest cluster: $N_G \sim N^{1/2}$
 - * The clusters may contain loops

- (d) Supercritical Regime**
- * Single giant component
 - * Cluster size distribution: $p_s \sim s^{-3/2} e^{-\alpha s}$
 - * Size of the giant component: $N_G \sim (p - p_c)N$
 - * The small clusters are trees
 - * Giant component has loops

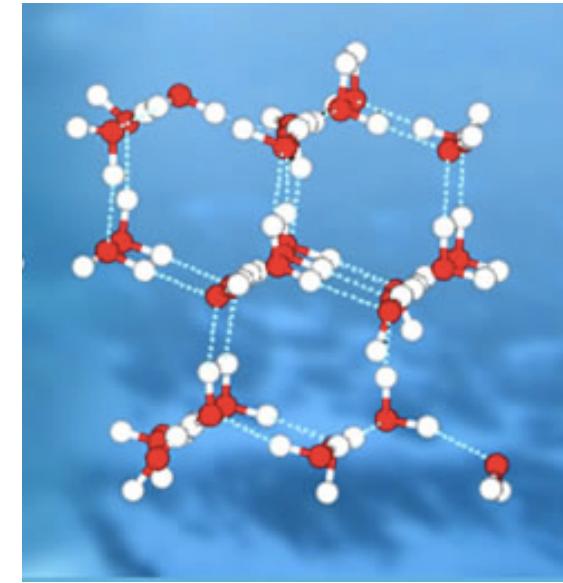
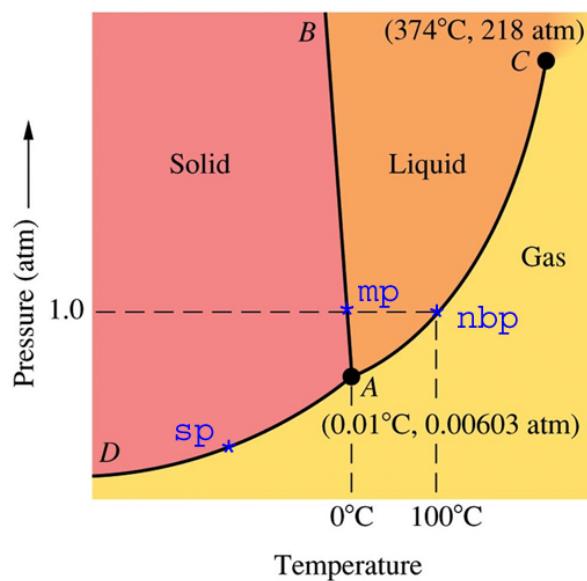
- (e) Connected Regime**
- * Single giant component
 - * No isolated nodes or clusters
 - * Size of the giant component: $N_G = N$
 - * Giant component has loops

Network	<i>N</i>	<i>L</i>	$\langle k \rangle$	$\ln N$
Internet	192,244	609,066	6.34	12.17
Power Grid	4,941	6,594	2.67	8.51
Science Collaboration	23,133	186,936	8.08	10.04
Actor Network	212,250	3,054,278	28.78	12.27
Yeast Protein Interactions	2,018	2,930	2.90	7.61

Esempio fisico



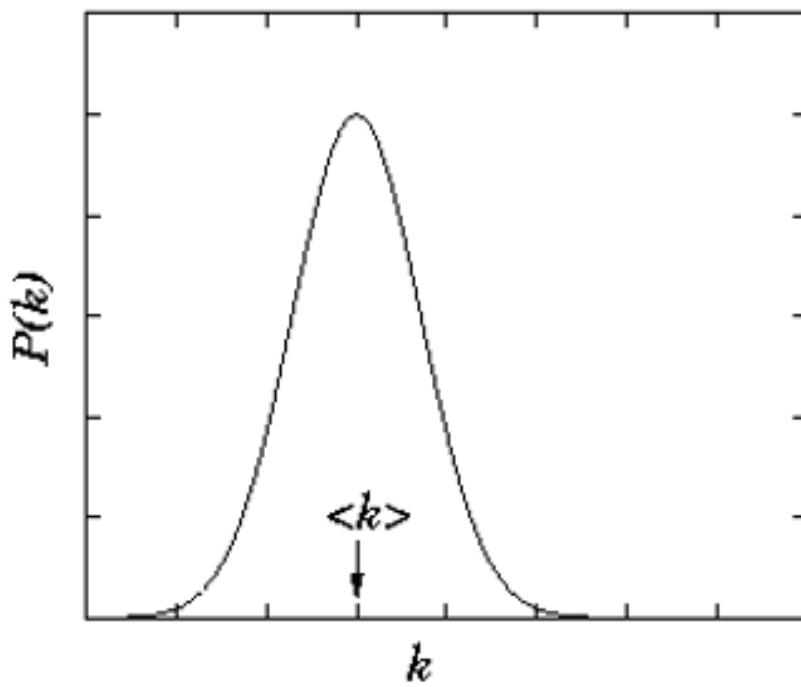
Water



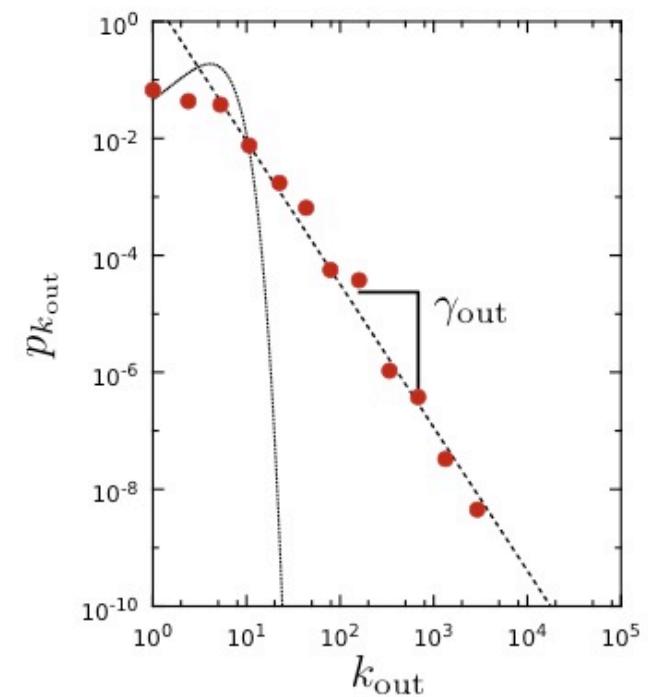
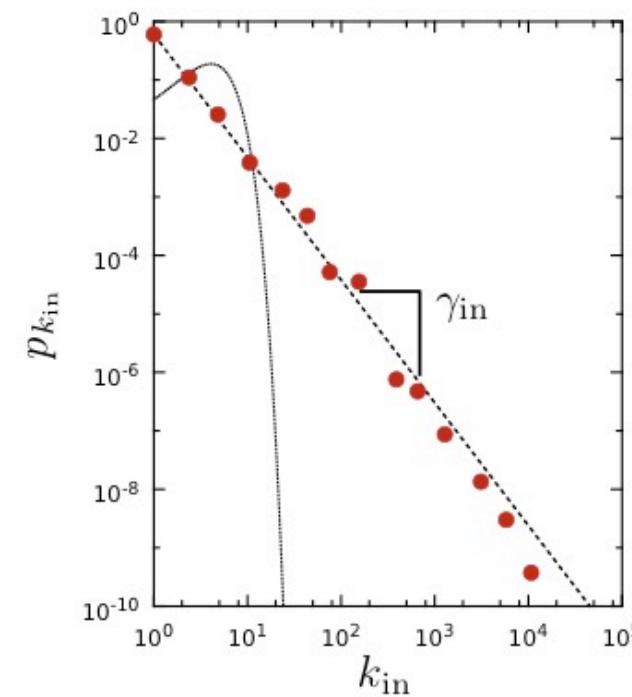
Ice

Come descrivo le reti reali?

Expected

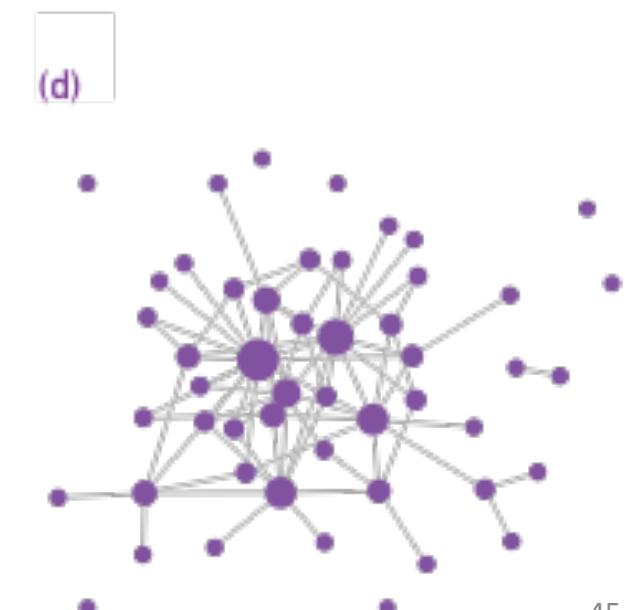
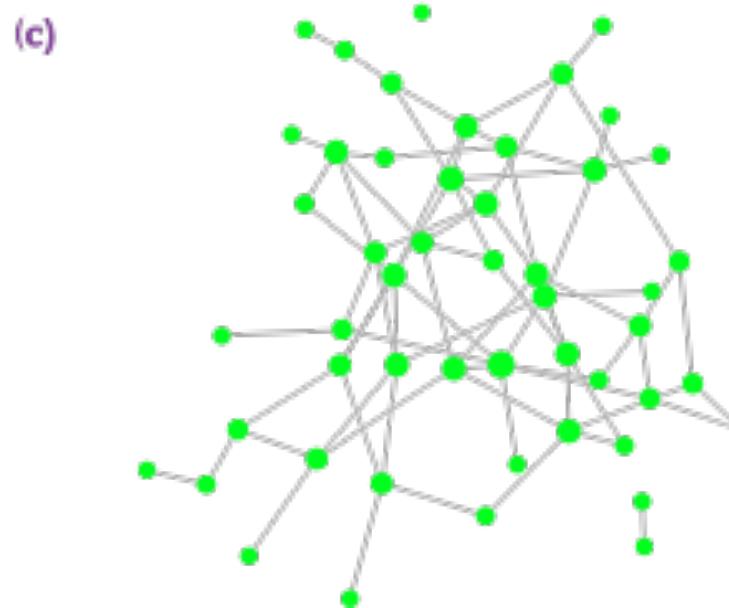
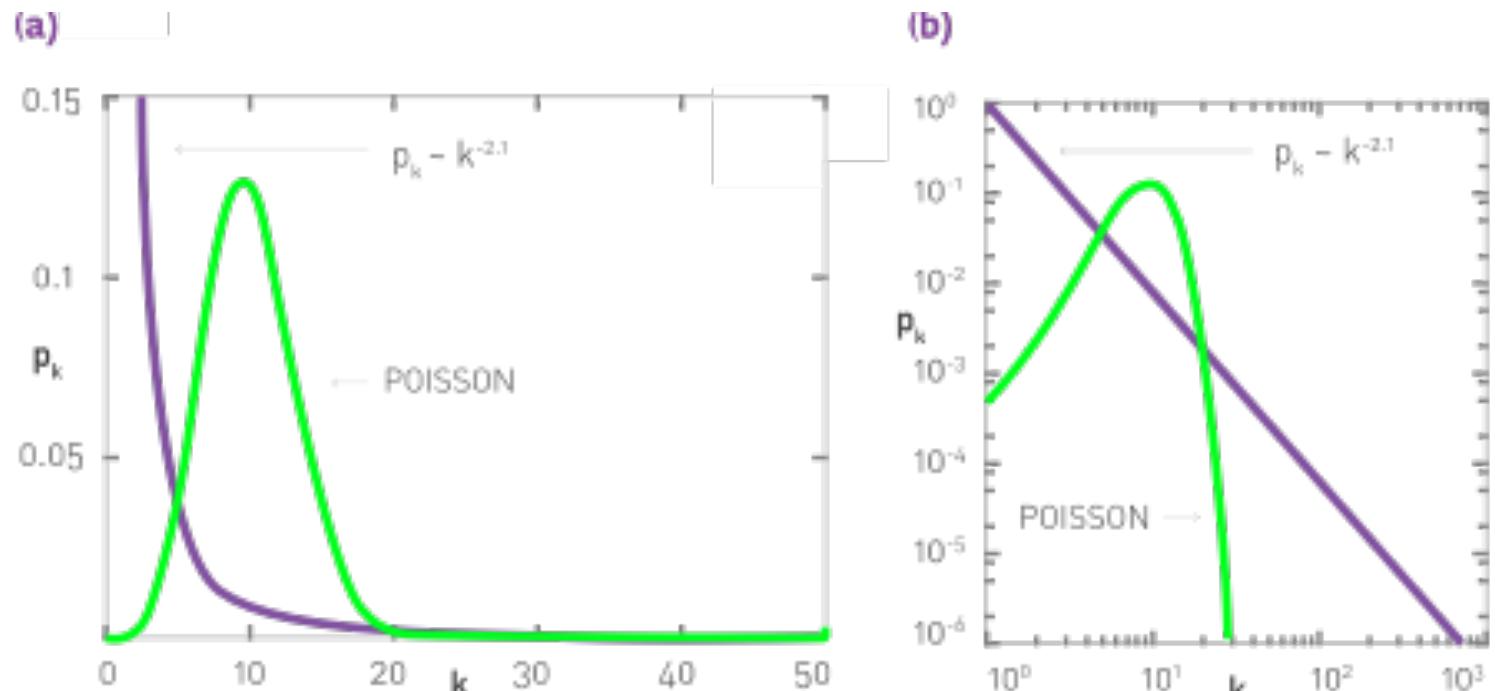


Reality

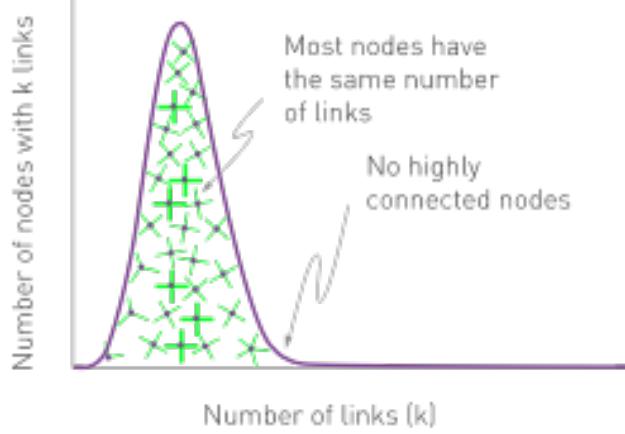


HUBS

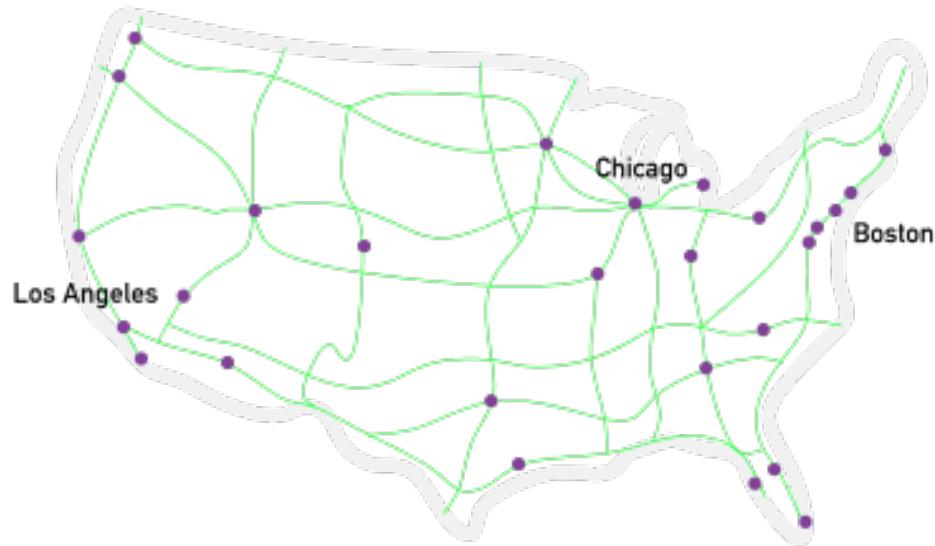
- Un **hub** è un nodo con grado MOLTO Maggiore rispetto al grafo medio della rete



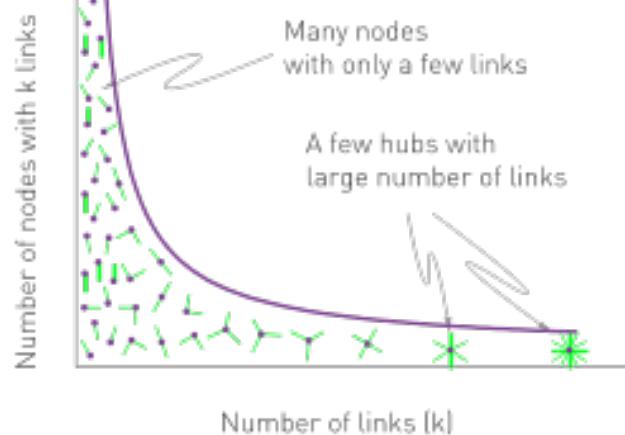
(a) POISSON



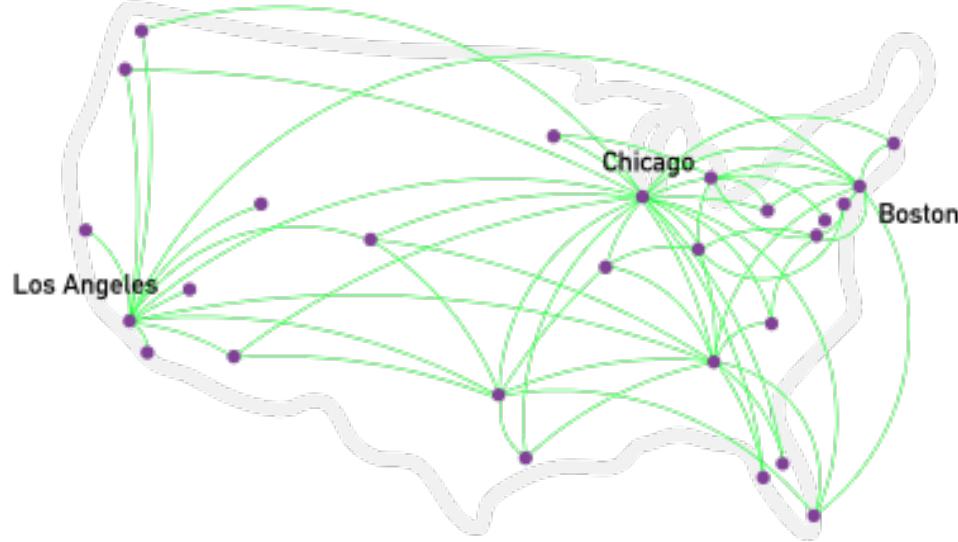
(b)



(c) POWER LAW

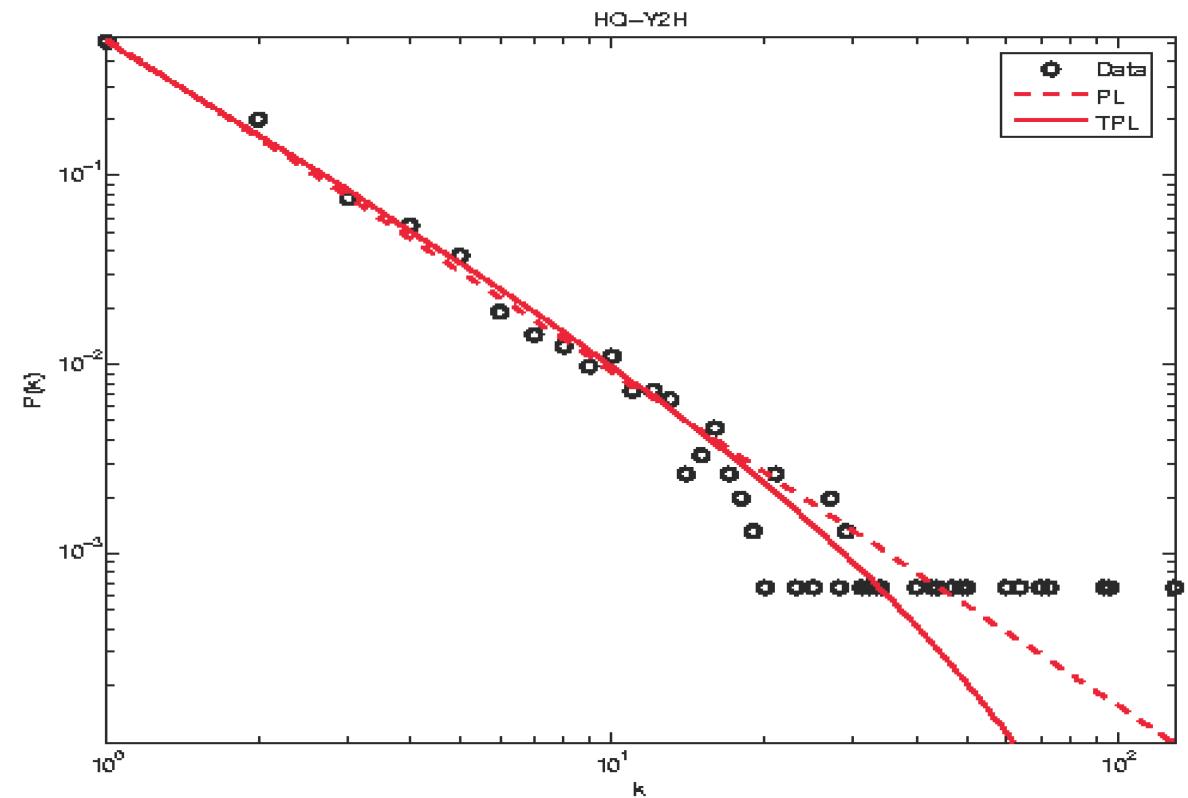


(d)



Reti Reali

- Le reti reali sono le reti che presentano gli hubs e sono dette **Scale Free**
- La degree distribution di una rete Scale Free è una **legge di potenza**
- In scala logaritmica, la DD diventa una retta in cui l'esponente è il coefficiente angolare
- Convivono ordini di grandezza molto diversi tra loro



Perché si chiamano scale free?

- Si dimostra che l'indeterminazione sul grado medio è infinita

$$P(k) = Ck^{-\gamma} \quad k = [k_{\min}, \infty)$$
$$\int_{k_{\min}}^{\infty} P(k) dk = 1 \quad C = \frac{1}{\int_{k_{\min}}^{\infty} k^{-\gamma} dk} = (\gamma - 1)k_{\min}^{\gamma-1}$$
$$P(k) = (\gamma - 1)k_{\min}^{\gamma-1}k^{-\gamma}$$

$$\langle k^m \rangle = \int_{k_{\min}}^{\infty} k^m P(k) dk \quad \langle k^m \rangle = (\gamma - 1)k_{\min}^{\gamma-1} \int_{k_{\min}}^{\infty} k^{m-\gamma} dk = \frac{(\gamma - 1)}{(m - \gamma + 1)} k_{\min}^{\gamma-1} \left[k^{m-\gamma+1} \right]_{k_{\min}}^{\infty}$$

Se $m - \gamma + 1 > 0$, l'integrale converge

Se scelgo un nodo a caso

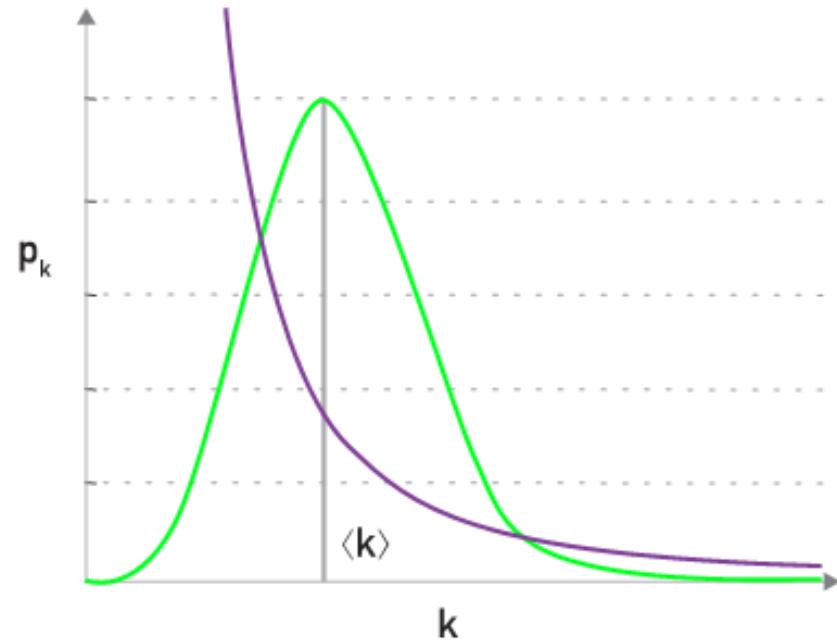
- Random Network:

$$k = \langle k \rangle \pm \langle k \rangle^{\frac{1}{2}}$$

- Scale free network

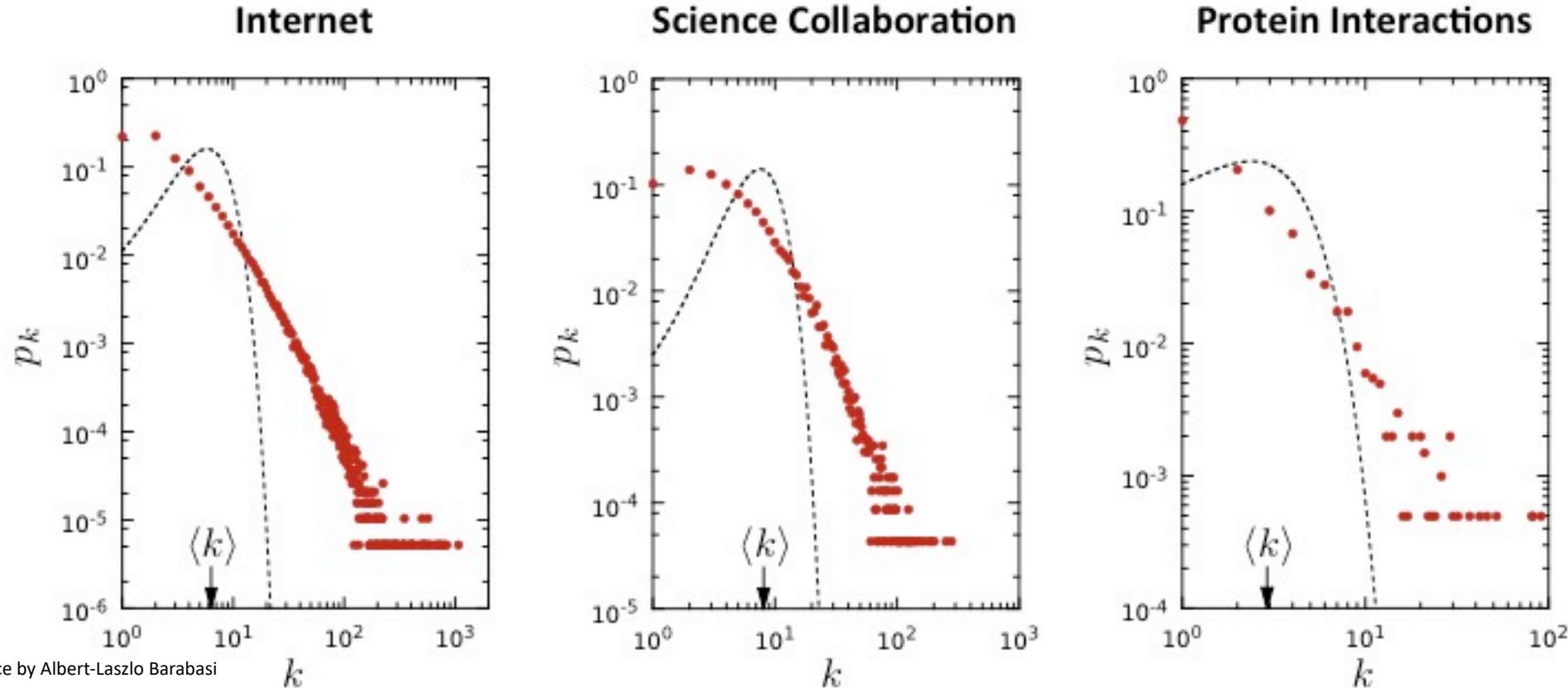
$$k = \langle k \rangle \pm \infty$$

- Non ho un parametro di scala



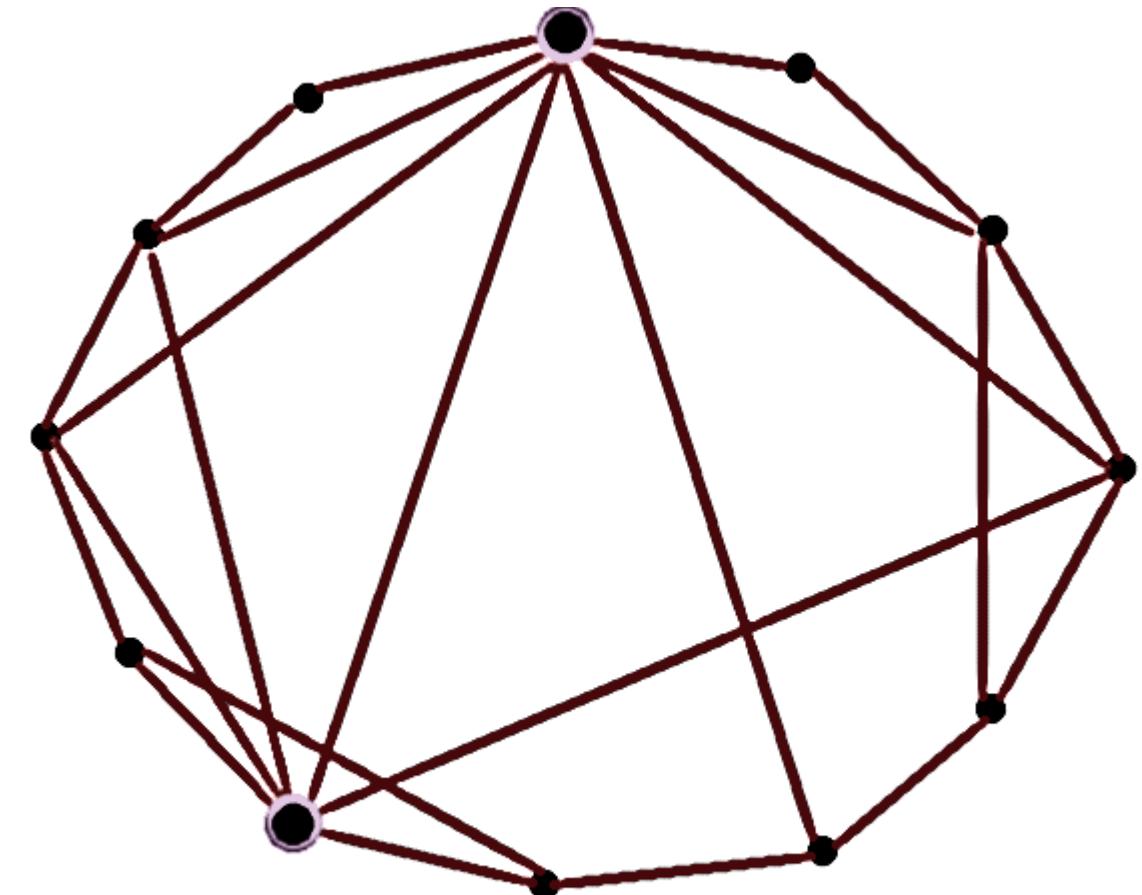
Universalità

- Le reti non dipendono dai dettagli del sistema
- Sono descrivibili da una retta, l'equazione più semplice che esista



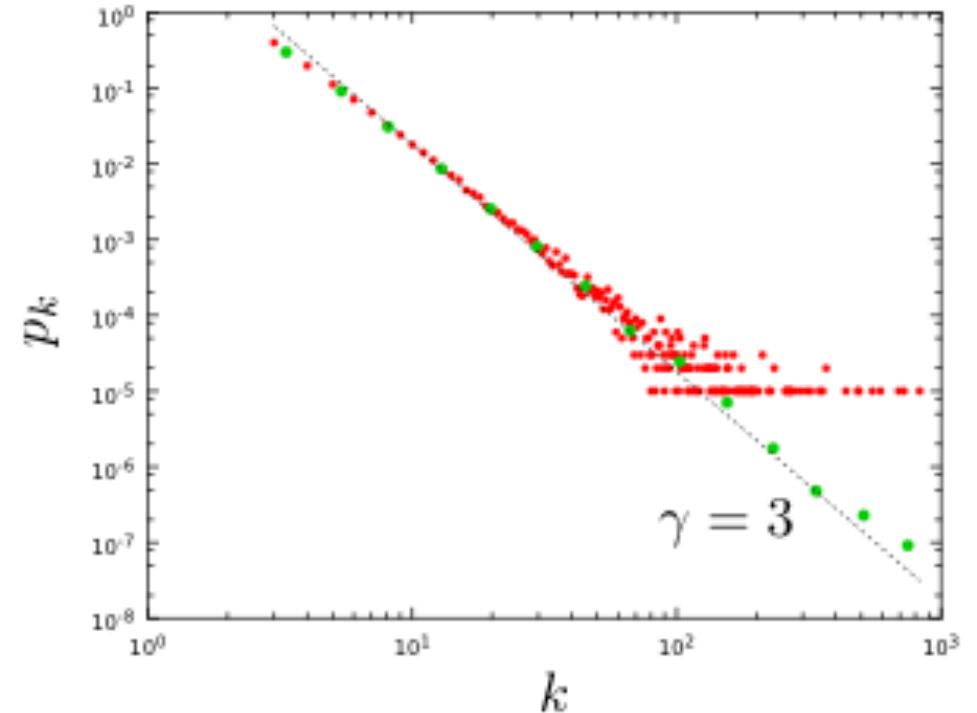
Ultra Small World

- Teoria dei **6 gradi** di separazione
- **Small World** $d = \frac{\log(N)}{\log < k >}$
- Ultra Small World $d = \log(\log(N))$
- Più la rete è densa e più le distanze si riducono grazie agli hubs

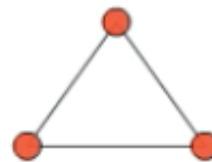


Modello di Barabasi - Albert

- **Crescita**: a ogni istante di tempo si aggiunge un nodo
- **Attaccamento Preferenziale**: la probabilità a cui il nodo si attacca non è uguale per tutti i nodi.
- In questo modello l'AP è rispetto al **grado**

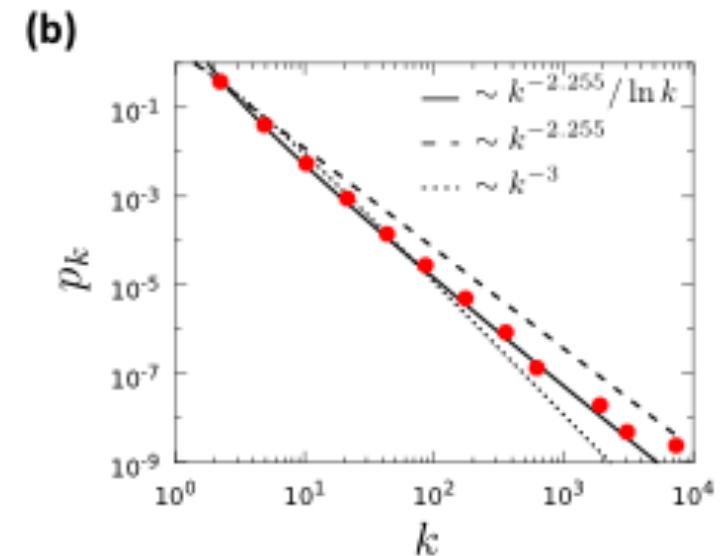
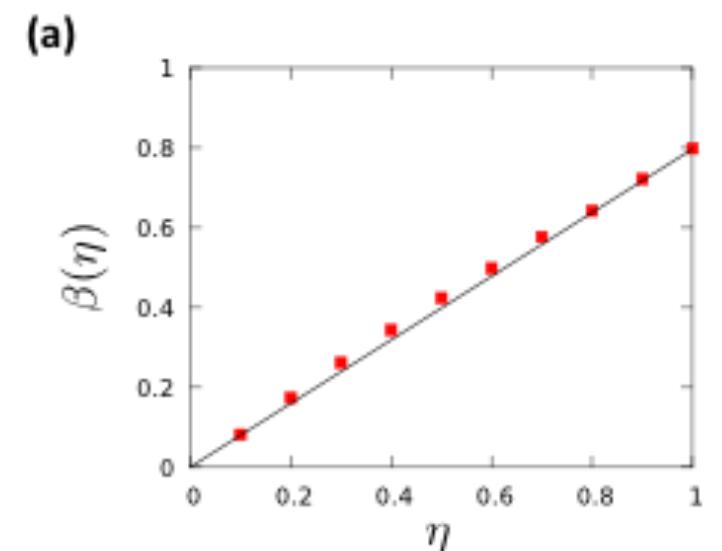


Modello di Barabasi - Albert



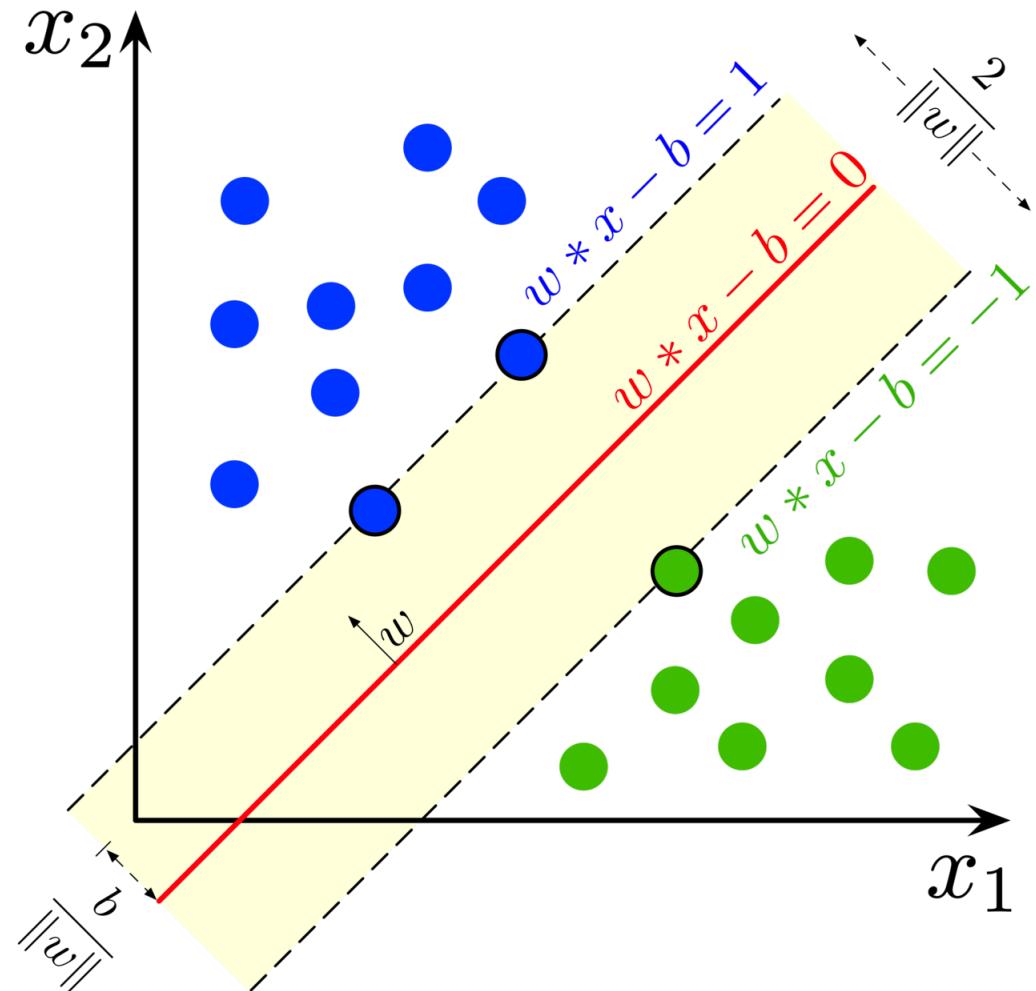
Modello di Bianconi - Barabasi

- Ho sempre la crescita
- Per il PA introduco la **fitness**, ossia l'abilità ad attirare nodi
- Altrimenti non posso uscire dal first mover advantage



Network Science AND ML

- Estraggo le **features** dalla rete
- Utilizzo Support Vector Machine



Grazie!

alessandro.cossard@gmail.com

