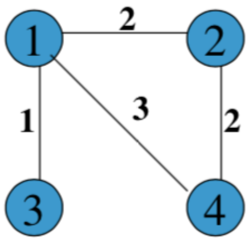
CLUSTERING

ALGORITMO MARKOV CLUSTERING

Markov Clustering

Parte dei contenuti della presente lezione sono tratti dai testi elencati di seguito.



Van Dongen, S. (2000)

Graph Clustering by Flow Simulation.

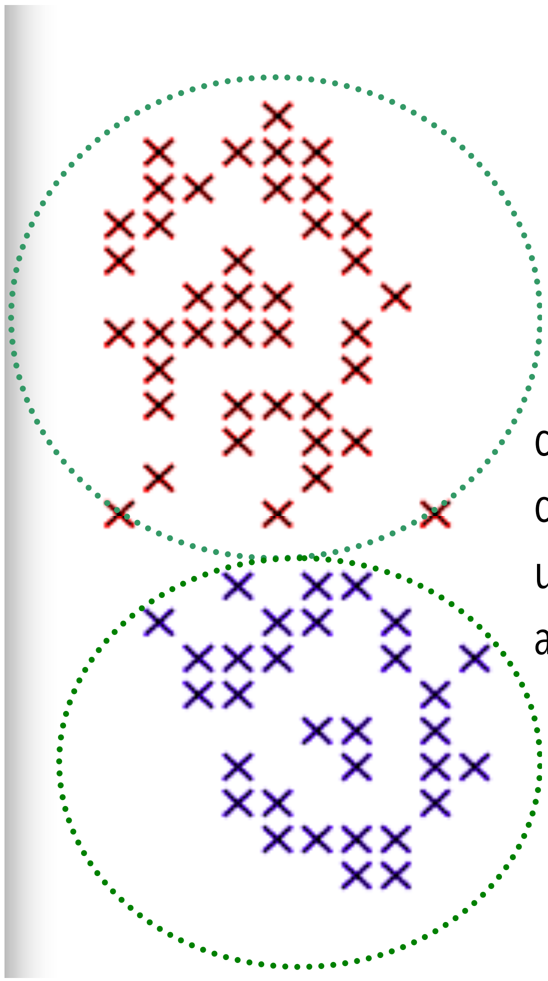
PhD Thesis, University of Utrecht, The Netherlands.

Il software MCL (Markov Clustering) è scaricabile all'indirizzo http://www.micans.org/mcl/

Markov Clustering

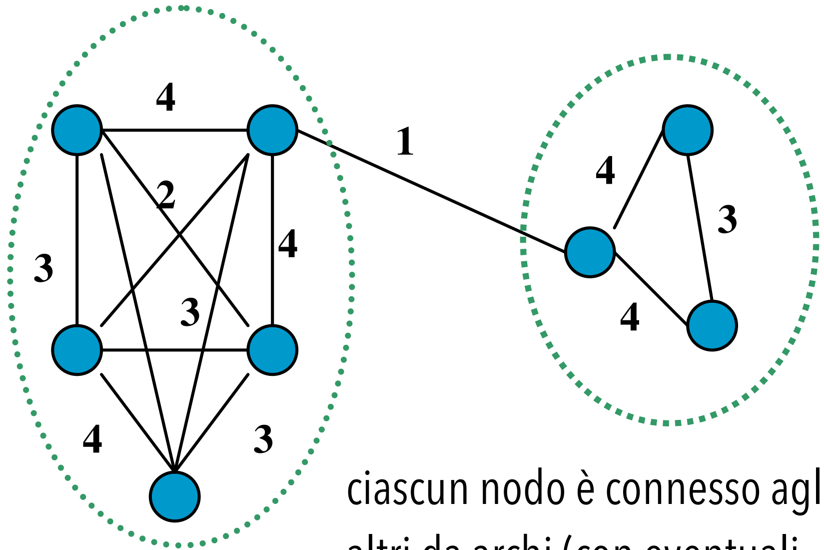
- È un algoritmo di clustering basato su grafi, utilizzato in bioinformatica.

Vector clustering



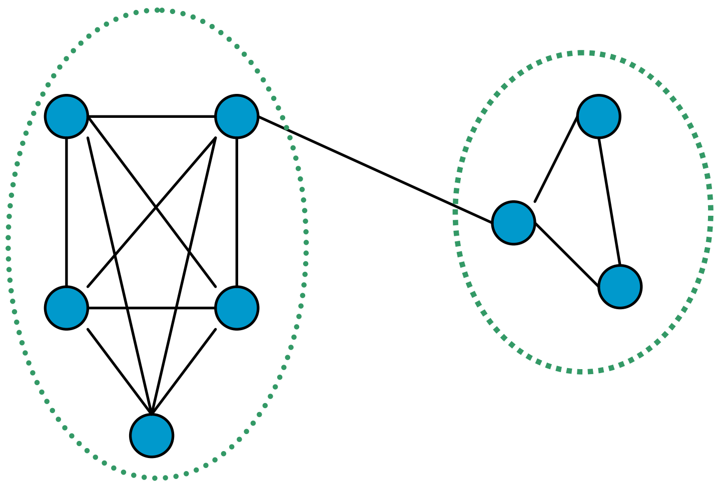
ciascun punto ha coordinate (x, y) ed una classe (colore) di appartenenza

Graph clustering



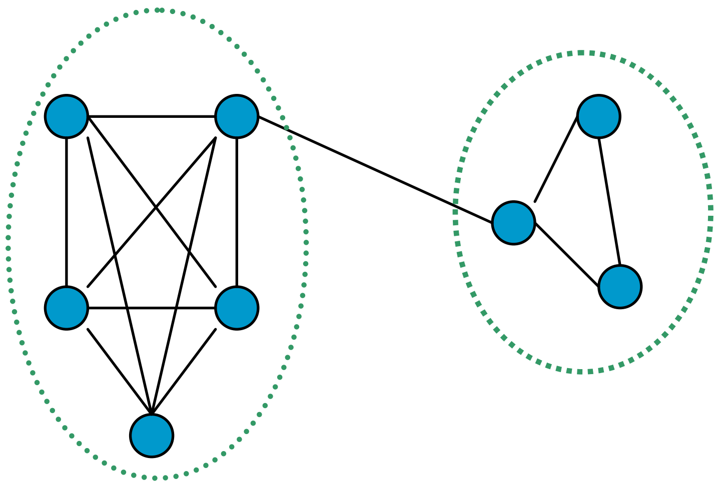
ciascun nodo è connesso agli altri da archi (con eventuali pesi)

Random Walks



- Considerando un grafo, vi saranno molti link all'interno di un cluster e pochi tra i cluster.
- Questo vuol dire che, se si parte da un nodo e si segue un percorso casuale sino ad un altro nodo connesso, è più probabile restare all'interno di un cluster che attraversarlo per raggiungere l'altro nodo.
- Questo è il concetto su cui è basato l'algoritmo MCL.

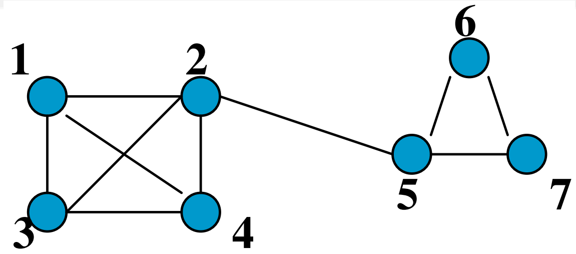
Random Walks



- Seguendo percorsi casuali (*random walks*) nel grafo, è possibile scoprire dove i flussi (percorsi) tendono a convergere, e quindi dove sono i cluster.
- I Random Walks su u grafo sono calcolati per mezzo delle “Markov Chains” (catene di Markov).

Random Walks

Vediamo un <u>esempio</u> di funzionamento.



- Ad un primo step, un *walker* casuale che parta dal nodo 1 ha una probabilità del 33% di andare verso i nodi 2, 3 e 4 e dello 0% di andare verso i nodi 5, 6 o 7.
- Partendo invece dal nodo 2, ha il 25% di probabilità di raggiungere i nodi 1, 3, 4, 5 e 0% verso i nodi 6 e 7.
- La corrispondente matrice di transizione (persorsi sulle colonne) sarà quindi:

$$\begin{array}{cc} & \begin{array}{ccccccc} \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \mathit{6} & \mathit{7} \end{array} \\ \begin{array}{c} \mathit{1} \\ \mathit{2} \\ \mathit{3} \\ \mathit{4} \\ \mathit{5} \\ \mathit{6} \\ \mathit{7} \end{array} & \begin{pmatrix} 0 & .25 & .33 & .33 & 0 & 0 & 0 \\ .33 & 0 & .33 & .33 & .33 & 0 & 0 \\ .33 & .25 & 0 & .33 & 0 & 0 & 0 \\ .33 & .25 & .33 & 0 & 0 & 0 & 0 \\ 0 & .25 & 0 & 0 & 0 & .5 & .5 \\ 0 & 0 & 0 & 0 & .33 & 0 & .5 \\ 0 & 0 & 0 & 0 & .33 & .5 & 0 \end{pmatrix} \end{array}$$

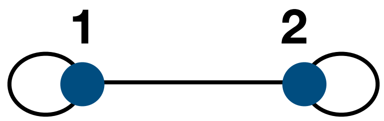
- ciasuna colonna ha somma 1
- può essere quindi vista come una *matrice di probabilità*

Markov Clustering

Markov Chain

Un esempio più semplice.

$$\begin{array}{c|c} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{pmatrix} .6 & .2 \\ .4 & .8 \end{pmatrix} \end{array}$$



- *le transizioni si concretizzano nel prodotto (ripetuto) delle matrici di probabilità*

- Valutiamo i passi ai tempi: $t_0 \rightarrow t_1 \rightarrow t_2$

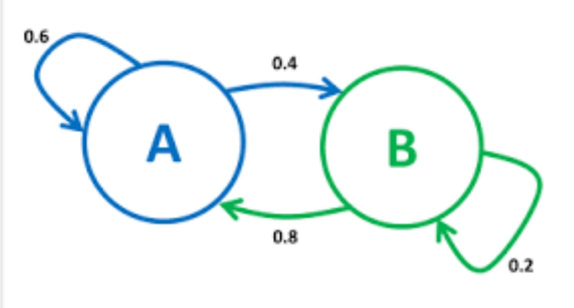
$$.6 \times .6 + .2 \times .4 = .44$$

$$\ldots$$

$$.4 \times .2 + .8 \times .8 = .72$$

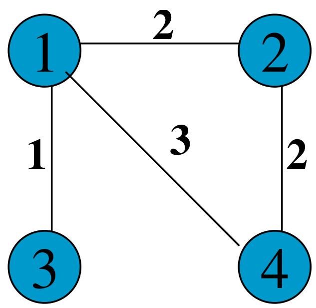
$$\begin{pmatrix} .6 & .2 \\ .4 & .8 \end{pmatrix} \cdot \begin{pmatrix} .6 & .2 \\ .4 & .8 \end{pmatrix} = \begin{pmatrix} .44 & .28 \\ .56 & .72 \end{pmatrix} \longrightarrow \begin{pmatrix} .35 & .32 \\ .65 & .68 \end{pmatrix} \longrightarrow \begin{pmatrix} .34 & .33 \\ .67 & .67 \end{pmatrix}$$

Markov Chain



- Una sequenza di variabili $X_1, X_2, X_3, \ldots$ (nel nostro caso le matrici di probabilità) nelle quali, dato lo stato presente, gli stati passato e futuro sono indipendenti.
- Una "Catena di Markov" (processo markoviano) quindi non ha memoria.
- Le probabilità per il passo temporale successivo dipendono solo dalle probabilità correnti.
- Un Random Walk è un esempio di Catena di Markov, utilizzando le matrici di transizione di probabilità.

Weighted Graphs (grafi pesati)



- Per trasformare un grafo pesato in una matrice di (transizione di) probabilità occorre normalizzare le colonne:

$$\begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1/2 & 1 & 3/5 \\ 1/3 & 0 & 0 & 2/5 \\ 1/6 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

- *una colonna si normalizza dividendo ciascun elemento per la somma dei suoi elementi;*
- *alla fine della trasformazione la matrice non è più simmetrica!*

Markov Clustering

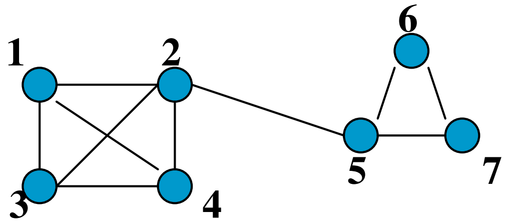
Self Loops

- Piccoli percorsi semplici con cicli (loop) possono complicare le cose.
  - Vi è un forte effetto dato dal fatto che potenze dispari di espansione ottengono la loro massa da percorsi semplici di lunghezza dispari, così come per quelli di lunghezza pari.
  - Ciò rende le probabilità di transizione dipendenti dalla *parità* delle lunghezze dei percorsi semplici.
- L'aggiunta di archi self-loop su ciascun nodo risolve il problema.
  - Un self-loop aggiunge un piccolo percorso di lunghezza 1 in modo che la massa non compaia solo nelle potenze dispari (o pari) della matrice.

$$
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0
\end{pmatrix}
\longrightarrow
\begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1
\end{pmatrix}
$$

Struttura a cluster di una catena di Markov

<u>Esempio</u>.



$$\begin{pmatrix} 0 & .25 & .33 & .33 & 0 & 0 & 0 \\ .33 & 0 & .33 & .33 & .33 & 0 & 0 \\ .33 & .25 & 0 & .33 & 0 & 0 & 0 \\ .33 & .25 & .33 & 0 & 0 & 0 & 0 \\ 0 & .25 & 0 & 0 & 0 & .5 & .5 \\ 0 & 0 & 0 & 0 & .33 & 0 & .5 \\ 0 & 0 & 0 & 0 & .33 & .5 & 0 \end{pmatrix} \xrightarrow{\text{alla fine}} \begin{pmatrix} .15 & .15 & .15 & .15 & .15 & .15 & .15 \\ .2 & .2 & .2 & .2 & .2 & .2 & .2 \\ .15 & .15 & .15 & .15 & .15 & .15 & .15 \\ .15 & .15 & .15 & .15 & .15 & .15 & .15 \\ .15 & .15 & .15 & .15 & .15 & .15 & .15 \\ .1 & .1 & .1 & .1 & .1 & .1 & .1 \\ .1 & .1 & .1 & .1 & .1 & .1 & .1 \end{pmatrix}$$

- Si osservi che, nei passi iniziali, prima che il flusso si mescoli, la struttura dei cluster è già evidente nella matrice.
- Questa non è una coincidenza e l'algoritmo MCL utilizza questa caratteristica, modificando il processo di random walk, per enfatizzare ulteriormente la separazione tra i cluster nella matrice.

L’algoritmo MCL

- Il flusso è più facile attraverso regioni dense che attraverso confini sparsi; comunque, alla lunga questo effetto svanisce.
- Durante le potenze iniziali della Catena di Markov, i pesi degli archi saranno **maggiori** nei link *interni* ai cluster, e minori nei link *tra* i cluster.
- Questo vuol dire che vi è una corrispondenza tra la distribuzione dei pesi sulle colonne e i clustering.

L'algoritmo MCL

- MCL aumenta deliberatamente tale effetto:
  - interrompendo prima la catena;
  - quindi modificando le transizioni attraverso le colonne.
- Per ogni nodo, i valori di transizione vengono modificati in modo che:
  - i vicini "forti" sono ulteriormente rafforzati;
  - i vicini "più deboli" vengono retrocessi.
- Questa modifica può essere effettuata elevando una singola colonna ad una potenza non negativa, e quindi ri-normalizzandola.
- Questa operazione è chiamata "Inflazione" (*Inflation*).
- L'elevamento a potenza della matrice è chiamato "Espansione" (*Expansion*).

MCL Inflation

- Esempio di inflazione di ordine 2 (elevamento al quadrato):

$$\begin{array}{cc} \text{Colonna } i & \text{Colonna } j \\ \begin{pmatrix} 0 \\ 1/2 \\ 0 \\ 1/6 \\ 1/3 \end{pmatrix} & \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix} \end{array}$$

$$\Big\downarrow$$

- *Eleva una colonna al quadrato, quindi la normalizza*

$$\begin{pmatrix} 0 \\ 9/14 \\ 0 \\ 1/14 \\ 4/14 \end{pmatrix} \quad \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix}$$

MCL Inflation

Definizione. *Data una matrice $M \in \mathbb{R}^{k \times l}$, $M \geq 0$, ed un numero reale non negativo r, $\Gamma_r M$ è la matrice che si ottiene elevando ciascuna colonna di M alla potenza r; Γ_r è chiamato l'operatore di* ***inflazione*** *con coefficiente potenza r.*

Formalmente, l'azione di $\Gamma_r : \mathbb{R}^{k \times l} \to \mathbb{R}^{k \times l}$ è così definita:

$$(\Gamma_r M)_{pq} = \frac{(M_{pq})^r}{\sum_{i=1}^{k} (M_{iq})^r}$$

q indica il vertice (colonna) attratto dal vertice p

Se r viene omesso, il coefficiente di potenza è uguale a 2.

- L'operatore di inflazione è responsabile sia del rafforzamento che dell'indebolimento dei flussi (aumenta la forza dei flussi già forti, diminuisce la forza dei flussi già deboli).
- Il parametro di inflazione *r* controlla la rapidità di tale processo.
- Questo alla fine influenza la *granularità* dei cluster ottenuti.

Algoritmo MCL

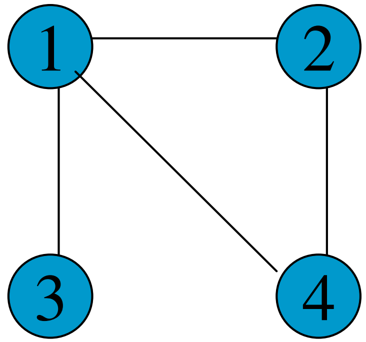
- Nell’algoritmo MCL si alternano ripetutamente i due seguenti processi:
  - **Espansione** (elevare a potenza la matrice di transizione);
  - **Inflazione**.
- L’operatore di espansione consente al flusso di connettere regioni differenti del grafo.
- L’operatore di inflazione è responsabile dell’aumento (intra-cluster) e diminuzione (inter-cluster) del flusso.

Algoritmo MCL

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

Algoritmo MCL

$e = 2$
$r = 2$

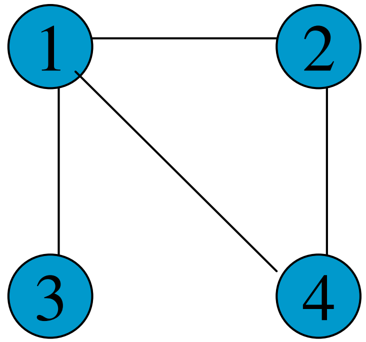


1. **Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.**
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

Algoritmo MCL

$e = 2$

$r = 2$



$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

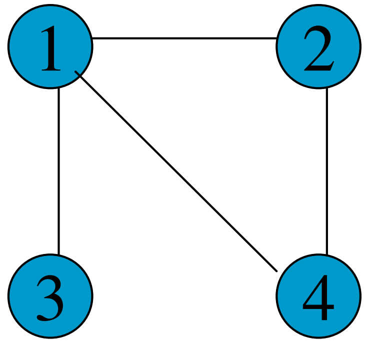
1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. **Creare la matrice *M* associata al grafo.**
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

Algoritmo MCL

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. **Aggiungere (eventualmente) i self-loop ai nodi.**
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

$e = 2$

$r = 2$



$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

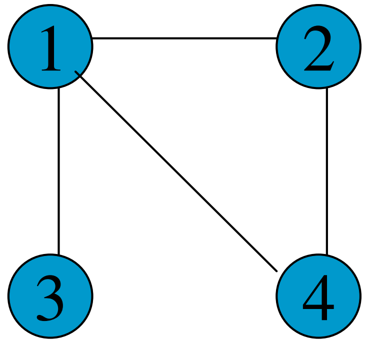
Markov Clustering

Algoritmo MCL

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. **Normalizzare la matrice *M*.**
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

$\boldsymbol{e = 2}$

$\boldsymbol{r = 2}$



$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

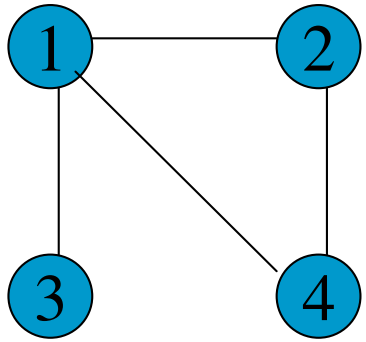
$$\begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 \\ 1/4 & 0 & 1/2 & 0 \\ 1/4 & 1/3 & 0 & 1/3 \end{pmatrix}$$

Algoritmo MCL

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. **Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.**
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [*M*(*i*+1) = *M*(*i*)].
8. Interpretare la matrice risultante per scoprire i cluster.

$e = 2$

$r = 2$



$$\begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 \\ 1/4 & 0 & 1/2 & 0 \\ 1/4 & 1/3 & 0 & 1/3 \end{pmatrix} \cdot \begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/3 \\ 1/4 & 1/3 & 0 & 1/3 \\ 1/4 & 0 & 1/2 & 0 \\ 1/4 & 1/3 & 0 & 1/3 \end{pmatrix}$$

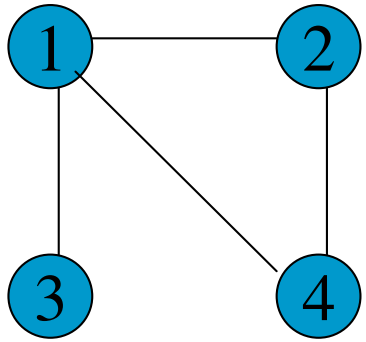
$$= \begin{pmatrix} .35 & .31 & .38 & .31 \\ .23 & .31 & .13 & .31 \\ .19 & .08 & .38 & .08 \\ .23 & .31 & .13 & .31 \end{pmatrix}$$

Algoritmo MCL

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. **Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.**
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. Interpretare la matrice risultante per scoprire i cluster.

$\boldsymbol{e = 2}$

$\boldsymbol{r = 2}$



$$\begin{pmatrix} .35 & .31 & .38 & .31 \\ .23 & .31 & .13 & .31 \\ .19 & .08 & .38 & .08 \\ .23 & .31 & .13 & .31 \end{pmatrix}$$

$$\begin{pmatrix} .13 & .09 & .14 & .09 \\ .05 & .09 & .02 & .09 \\ .04 & .01 & .14 & .01 \\ .05 & .09 & .02 & .09 \end{pmatrix}$$

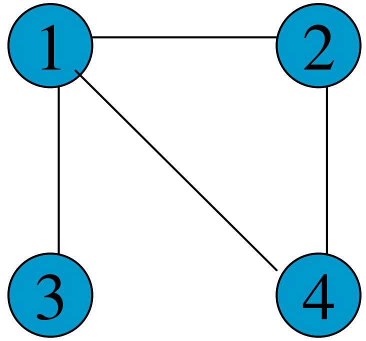
$$\begin{pmatrix} .47 & .33 & .45 & .33 \\ .20 & .33 & .05 & .33 \\ .13 & .02 & .45 & .02 \\ .20 & .33 & .05 & .33 \end{pmatrix}$$

Markov Clustering

Algoritmo MCL

$e = 2$

$r = 2$



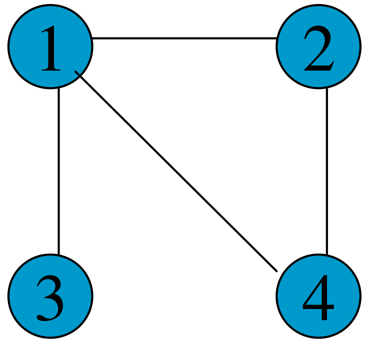
1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. **Ripetere i passi 5 e 6 fino alla convergenza [*M*(*i*+1) = *M*(*i*)].**
8. Interpretare la matrice risultante per scoprire i cluster.

$$\begin{pmatrix} .70 & .33 & .49 & .33 \\ .12 & .33 & .01 & .33 \\ .05 & .02 & .49 & \text{--} \\ .12 & .33 & .01 & .33 \end{pmatrix}$$

$$\begin{pmatrix} .94 & .33 & .50 & .33 \\ .03 & .33 & \text{--} & .33 \\ .01 & \text{--} & .50 & \text{--} \\ .13 & .33 & \text{--} & .33 \end{pmatrix}$$

$$\begin{pmatrix} 1 & .33 & .50 & .33 \\ \text{--} & .33 & \text{--} & .33 \\ \text{--} & \text{--} & .50 & \text{--} \\ \text{--} & .33 & \text{--} & .33 \end{pmatrix}$$

Algoritmo MCL



$e = 2$
$r = 2$

1. Input: grafo *g* non orientato (bidirezionale), parametro di espansione *e*, parametro di inflazione *r*.
2. Creare la matrice *M* associata al grafo.
3. Aggiungere (eventualmente) i self-loop ai nodi.
4. Normalizzare la matrice *M*.
5. Espandere elevando la matrice *M* alla potenza *e*, ottenendo *M'*.
6. Inflazionare la matrice *M'* ottenuta al passo precedente applicando il parametro *r*.
7. Ripetere i passi 5 e 6 fino alla convergenza [$M(i+1) = M(i)$].
8. **Interpretare la matrice risultante per scoprire i cluster.**

più avanti…

Convergenza dell'algoritmo MCL

- Non è dimostrato che l'algoritmo converga [$M(i+1) = M(i)$]; nella tesi di dottorato l'autore ne mostra la convergenza solo in via sperimentale…
- In pratica, l'algoritmo converge *quasi sempre* verso una matrice "doppiamente idempotente" (stato stazionario, valori uguali nelle colonne):

$$\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}$$

$$M_{mcl}^{\infty}$$

Convergenza dell'algoritmo MCL

- È dimostrato che quando la matrice sta per diventare "doppiamente idempotente", l'algoritmo converge quadraticamente.
- Comunque, lo stato stazionario finale può talvolta essere ciclico e consistere in una sequenza di matrici che si ripetono identicamente.
  - *In certi casi, l'espansione e l'inflazione si comportano come l'una l'inversa dell'altra.*
  - *Ciò si verifica di solito in mancanza di self-loop nei grafi bipartiti a causa della lunghezza dispari dei cammini.*
  - *Per ovviare all'inconveniente è sufficiente aggiungere i self-loop e apportare una leggera modifica ai parametri.*

Convergenza dell’algoritmo MCL

$$
\underset{M}{\begin{pmatrix}
0.200 & 0.250 & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
0.200 & 0.250 & 0.250 & -- & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & 0.250 & 0.250 & 0.200 & 0.200 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & 0.250 & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & 0.250 & 0.250 & -- & 0.200 & -- & 0.250 & 0.200 & -- & -- & -- & -- \\
0.200 & -- & -- & -- & -- & 0.333 & -- & -- & -- & 0.250 & -- & -- \\
0.200 & -- & -- & -- & 0.200 & -- & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & 0.200 & -- & -- & 0.200 & 0.200 & -- & 0.200 & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
0.200 & -- & -- & -- & -- & 0.333 & 0.250 & -- & -- & 0.250 & -- & -- \\
-- & -- & -- & 0.200 & -- & -- & -- & 0.200 & 0.200 & -- & 0.200 & 0.333 \\
-- & -- & -- & -- & -- & -- & -- & -- & 0.200 & -- & 0.200 & 0.333
\end{pmatrix}}
$$

$$
\underset{\Gamma_2 M^2,}{\begin{pmatrix}
0.380 & 0.087 & 0.027 & -- & 0.077 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
0.047 & 0.347 & 0.210 & 0.017 & 0.150 & 0.019 & 0.066 & 0.012 & -- & 0.012 & -- & -- \\
0.014 & 0.210 & 0.347 & 0.056 & 0.150 & -- & 0.016 & 0.046 & 0.009 & -- & 0.009 & -- \\
-- & 0.027 & 0.087 & 0.302 & 0.062 & -- & -- & 0.184 & 0.143 & -- & 0.143 & 0.083 \\
0.058 & 0.210 & 0.210 & 0.056 & 0.406 & -- & 0.083 & 0.046 & 0.009 & 0.019 & 0.009 & -- \\
0.142 & 0.017 & -- & -- & -- & 0.295 & 0.083 & -- & -- & 0.184 & -- & -- \\
0.113 & 0.069 & 0.017 & -- & 0.062 & 0.097 & 0.333 & 0.012 & -- & 0.147 & -- & -- \\
-- & 0.017 & 0.069 & 0.175 & 0.049 & -- & 0.016 & 0.287 & 0.143 & -- & 0.143 & 0.083 \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
0.246 & 0.017 & -- & -- & 0.019 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\
-- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\
-- & -- & -- & 0.044 & -- & -- & -- & 0.046 & 0.120 & -- & 0.120 & 0.278
\end{pmatrix}}
$$

Convergenza dell’algoritmo MCL

$$\begin{pmatrix}
0.448 & 0.080 & 0.023 & -- & 0.068 & 0.426 & 0.359 & -- & -- & 0.432 & -- & -- \\
0.018 & 0.285 & 0.228 & 0.007 & 0.176 & 0.006 & 0.033 & 0.005 & -- & 0.007 & -- & -- \\
0.005 & 0.223 & 0.290 & 0.022 & 0.173 & -- & 0.010 & 0.017 & 0.003 & 0.001 & 0.003 & 0.001 \\
-- & 0.018 & 0.059 & 0.222 & 0.040 & -- & 0.001 & 0.187 & 0.139 & -- & 0.139 & 0.099 \\
0.027 & 0.312 & 0.314 & 0.028 & 0.439 & 0.005 & 0.054 & 0.022 & 0.003 & 0.010 & 0.003 & 0.001 \\
0.116 & 0.007 & 0.001 & -- & 0.004 & 0.157 & 0.085 & -- & -- & 0.131 & -- & -- \\
0.096 & 0.040 & 0.013 & -- & 0.037 & 0.083 & 0.197 & 0.001 & -- & 0.104 & -- & -- \\
-- & 0.012 & 0.042 & 0.172 & 0.029 & -- & 0.002 & 0.198 & 0.133 & -- & 0.133 & 0.096 \\
-- & 0.001 & 0.015 & 0.256 & 0.009 & -- & -- & 0.266 & 0.326 & -- & 0.326 & 0.346 \\
0.290 & 0.021 & 0.002 & -- & 0.017 & 0.323 & 0.260 & -- & -- & 0.316 & -- & -- \\
-- & 0.001 & 0.015 & 0.256 & 0.009 & -- & -- & 0.266 & 0.326 & -- & 0.326 & 0.346 \\
-- & -- & 0.001 & 0.037 & 0.001 & -- & -- & 0.039 & 0.069 & -- & 0.069 & 0.112
\end{pmatrix}$$

$\Gamma_2(\Gamma_2 M^2 \cdot \Gamma_2 M^2)$

$$\begin{pmatrix}
0.807 & 0.040 & 0.015 & -- & 0.034 & 0.807 & 0.807 & -- & -- & 0.807 & -- & -- \\
-- & 0.090 & 0.092 & -- & 0.088 & -- & -- & -- & -- & -- & -- & -- \\
-- & 0.085 & 0.088 & -- & 0.084 & -- & -- & -- & -- & -- & -- & -- \\
-- & 0.001 & 0.001 & 0.032 & 0.001 & -- & -- & 0.032 & 0.031 & -- & 0.031 & 0.031 \\
-- & 0.777 & 0.798 & -- & 0.786 & -- & 0.001 & -- & -- & -- & -- & -- \\
0.005 & -- & -- & -- & -- & 0.005 & 0.005 & -- & -- & 0.005 & -- & -- \\
0.003 & 0.001 & -- & -- & 0.001 & 0.003 & 0.003 & -- & -- & 0.003 & -- & -- \\
-- & -- & 0.001 & 0.024 & -- & -- & -- & 0.024 & 0.024 & -- & 0.024 & 0.024 \\
-- & -- & 0.002 & 0.472 & 0.001 & -- & -- & 0.472 & 0.472 & -- & 0.472 & 0.472 \\
0.185 & 0.005 & 0.001 & -- & 0.004 & 0.185 & 0.184 & -- & -- & 0.185 & -- & -- \\
-- & -- & 0.002 & 0.472 & 0.001 & -- & -- & 0.472 & 0.472 & -- & 0.472 & 0.472 \\
-- & -- & -- & 0.001 & -- & -- & -- & 0.001 & 0.001 & -- & 0.001 & --
\end{pmatrix}$$

$(\Gamma_2 \circ Squaring)$ iterated four times on M

Convergenza dell'algoritmo MCL

$$
\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}
$$

$$M_{mcl}^{\infty}$$

Interpretazione dei Cluster MCL

- Per interpretare i cluster, i vertici vengono divisi in due tipi:
  - *gli* ***attrattori*** *(che attraggono altri vertici);*
  - *gli attratti (vertici che vengono attratti dai primi).*
- Gli attrattori hanno almeno un valore di flusso positivo all'interno della loro corrispondente riga (della matrice stazionaria).
- Ciascun attrattore attira i vertici che hanno valori positivi all'interno della sua riga.

$$
\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}
$$

$$M_{mcl}^{\infty}$$

Interpretazione dei Cluster MCL

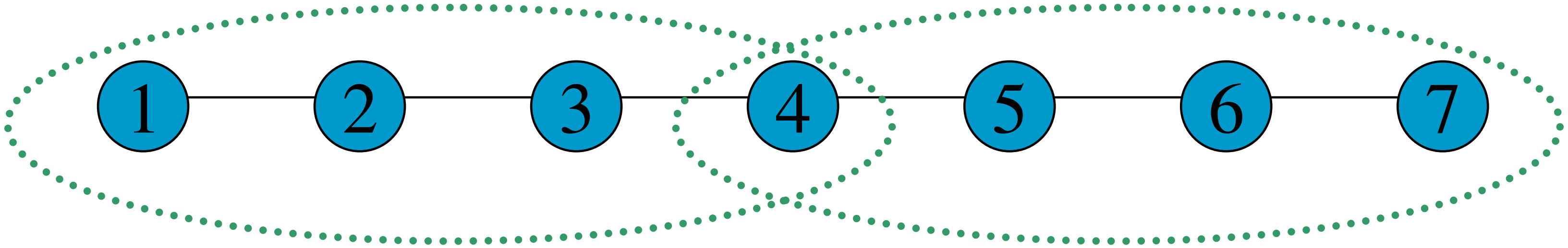
$$\begin{pmatrix}
1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\
-- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\
-- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & --
\end{pmatrix}$$

$$M_{mcl}^{\infty}$$

- Gli attrattori e gli elementi da essi attratti sono legati insieme nello stesso cluster.
- Nell'esempio sopra riportato, i cluster sono:
  - C_1 = {1, 6, 7, 10} (riga 1)
  - C_2 = {2, 3, 5} (riga 5)
  - C_3 = {4, 8, 9, 11, 12} (riga 9 = riga 11).

Interpretazione dei Cluster MCL

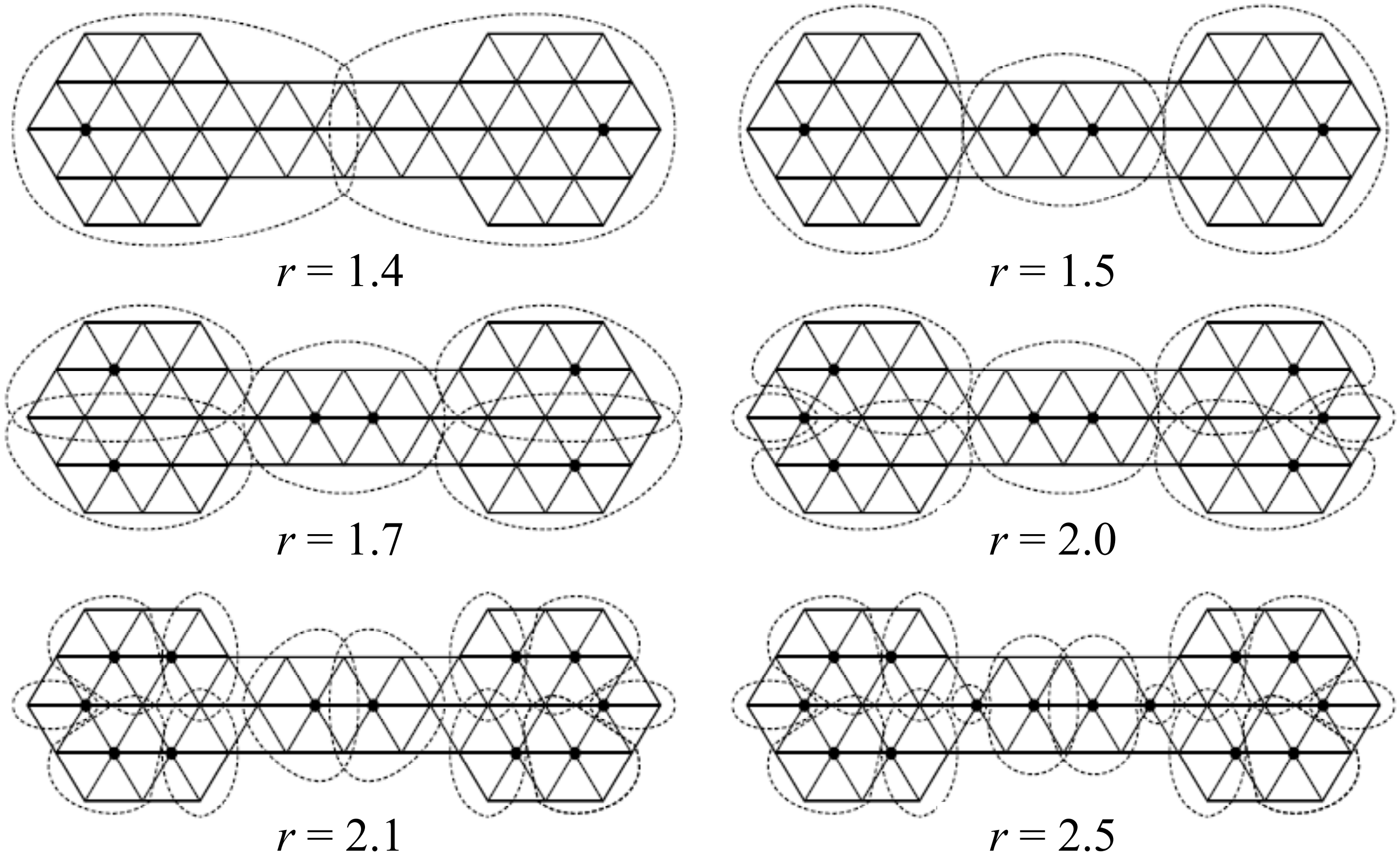
- In generale, cluster sovrapposti (in cui uno o più nodi appartengono a più di un cluster) risultano solo in alcuni particolari casi di grafi simmetrici:
  - solo quando un vertice (nodo del grafo) è attratto in maniera *esattamente* uguale da più di un cluster;
  - ciò accade solo quando entrambi i cluster sono isomorfi:



Markov Clustering

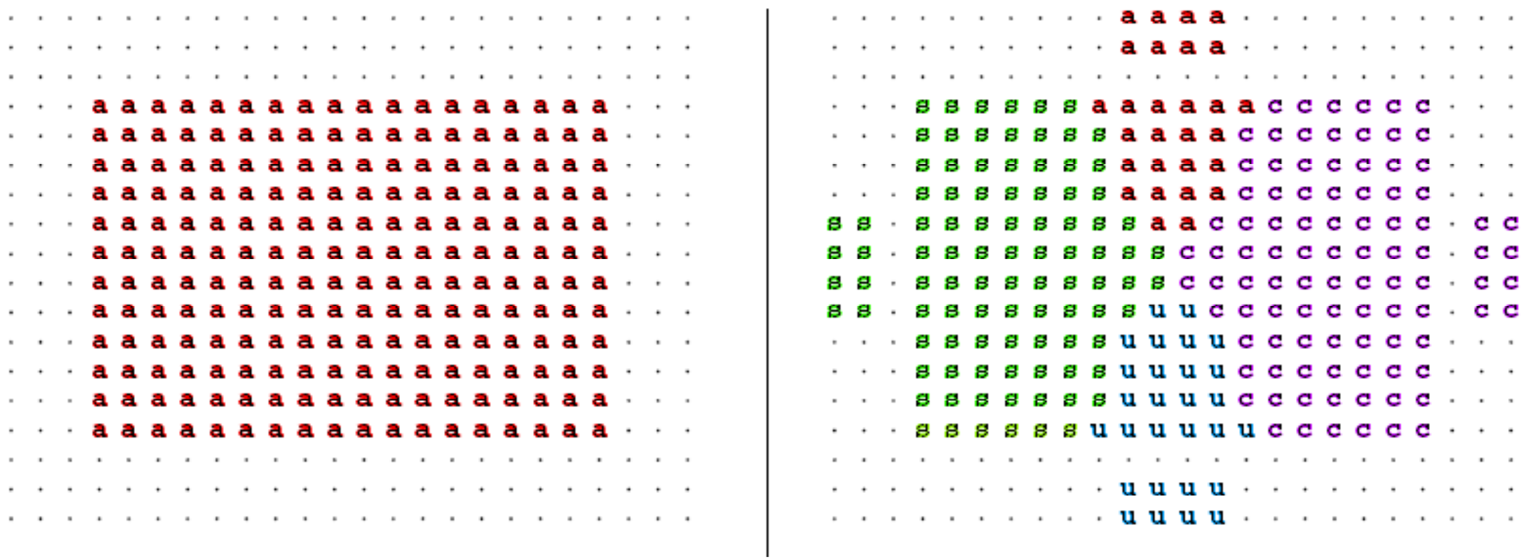
Cluster MCL

- Il parametro di inflazione r influenza la granularità dei cluster.
- Nell'esempio seguente il peso dei self-loop è 1.



Cluster MCL

- Per cluster di grande diametro, MCL ha problemi.
- Distribuire il flusso attraverso i cluster necessita di elevata espansione e bassa inflazione (altrimenti il cluster si scinde).
- Ciò comporta molte iterazioni e rende MCL sensibile a piccole perturbazioni nel grafo.
- L'aggiunta di cluster di piccolo diametro disturba il clustering, dato che il basso parametro di inflazione farà si che essi inflazionino sproporzionatamente le probabilità circostanti.



Analisi dell'algoritmo MCL

- Tempo di elaborazione proporzionale a N^3, dove N è il numero di vertici:
  - N^3 è il costo della moltiplicazione di due matrici di ordine N;
  - l'inflazione può essere eseguita in un tempo proporzionale a N^2;
  - non è dimostrato il numero di iterazioni necessarie per la convergenza dell'algoritmo, ma è stato dimostrato sperimentalmente essere $\sim 10 \div 100$ passi, per la maggior parte concernenti matrici sparse dopo i primi passi.
- La velocità di elaborazione può essere migliorata attraverso la rimozione (*pruning*) di valori superflui:
  - esaminando la matrice si possono impostare a zero i valori abbastanza piccoli (si presume che lo sarebbero diventati ad un certo passo);
  - l'algoritmo opera bene quando il diametro dei cluster è piccolo (distribuzione non omogenea dei pesi).

Analisi dell'algoritmo MCL

- Scala bene all'aumentare delle dimensioni del grafo.
- Opera con grafi pesati e non pesati.
- Produce buoni risultati di clustering.
- Robusto rispetto alla presenza di rumore nei dati del grafo.
- Numero di cluster non specificato inizialmente, ma è possibile regolare la granularità dei cluster con i parametri e ed r.
- In generale non è in grado di rilevare cluster sovrapposti.
- Non adatto per cluster di grande diametro.