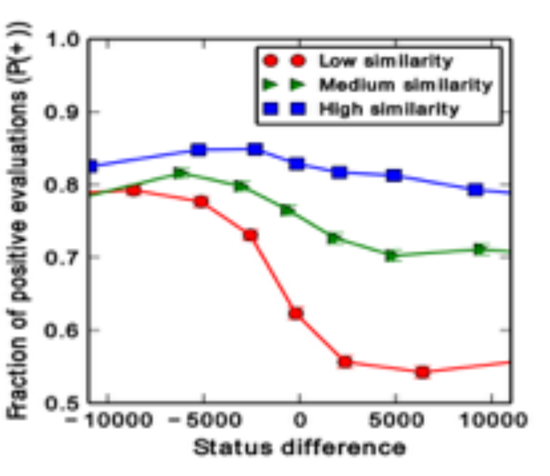
COMP4641 Lab10

User Evaluation, Recommender System, Homophily, K-Clique Community

User Evaluation

- 3 types: User to Item, User to User (Direct U-U), User to Item created by User (Indirect U-U).
- When A evaluates B:
    - 2 hypotheses about the probability B receives positive evaluation
        - depends primarily on B’s characteristics (Absolute Assessment)
        - depends on relationship between the characteristics of A and B (Relative Assessment)
    - 2 factors that affects the evaluation
        - relative status
        - mutual similarity



Recommender System

- rating matrix (implicit / explicit, real / integer) R: C x S . Sparsity, lots of missing ratings, how to predict those missing ratings?

- 3 approaches:

    - content-based

    - collaborative filtering

    - hybrid

Content-based recommendation: TF-IDF

- think of items as documents, features as words
- feature/word i in item/document j, f_{ij}: number of i in j, then TF score measures how important i is for j without prior knowledge about i

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$$

- IDF score measures how important i is in general, can be seemed as prior knowledge about i , n_i is number of items/documents which contains feature/word i, N is total number of items/documents

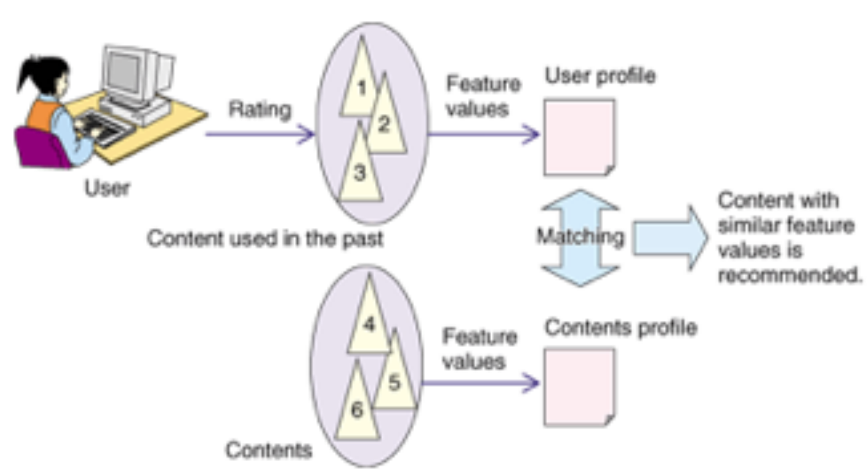
$$IDF_i = \log \frac{N}{n_i}$$

- TF-IDF score = TF * IDF , each item/document can be represented as a vector (item profile) which is made of each unique feature/word's TF-IDF score.

Content-based recommendation

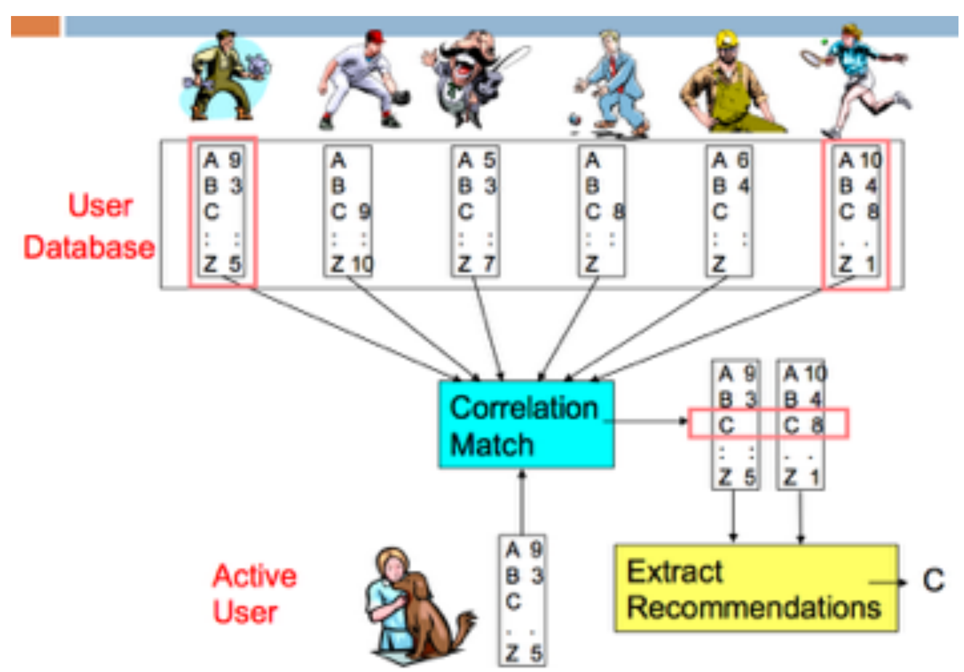
- each user is related to many item profiles, from which we can build a user profile (weighted averaging item profiles .e.g)

- estimate the fondness of a user on an item using cosine similarity between user profile and item profile



Collaborative Filtering

1. compute similarity between all users with the active user
2. select most similar users (neighbors) to serve as predictors (top n, or above threshold)
3. normalize ratings and compute a prediction from a weighted combination of the selected neighbors' ratings
4. recommend the item with top predicted ratings



Similarity Measure

- suppose $r_a = (r_a^1, r_a^2, \cdots, r_a^m)$ and $r_b = (r_b^1, r_b^2, \cdots, r_b^m)$ are the rating vectors for the m items rated by both user a and user b
- Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables. If the relationship between the variables is not linear, then the correlation coefficient does not adequately represent the strength of the relationship between the variables.
- plot m points $(r_a^1, r_b^1), \cdots, (r_a^m, r_b^m)$, calculate its Pearson correlation coefficient as the similarity between user a and user b. If m is too small, say <= 50, consider multiplied by some significance weight (m/50 .e.g).

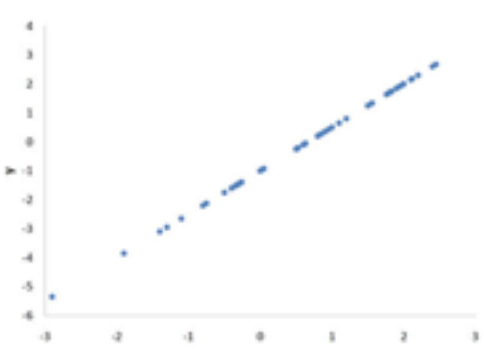


Figure 1. A perfect positive linear relationship, r = 1.

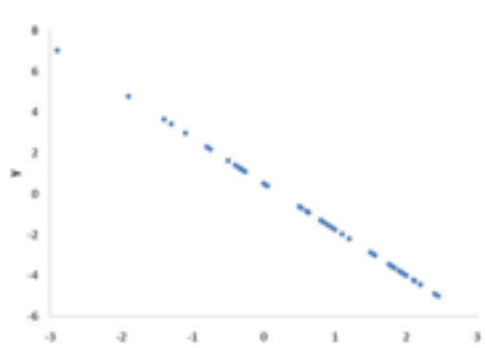


Figure 2. A perfect negative linear relationship, r = -1.

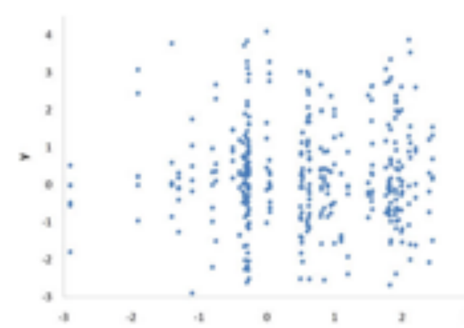


Figure 3. A scatter plot for which r = 0. Notice that there is no relationship between X and Y.

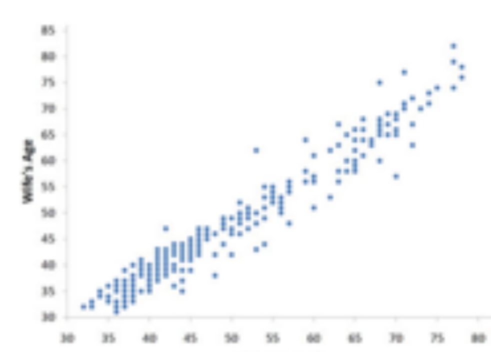


Figure 4. Scatter plot of spousal ages, r = 0.97.

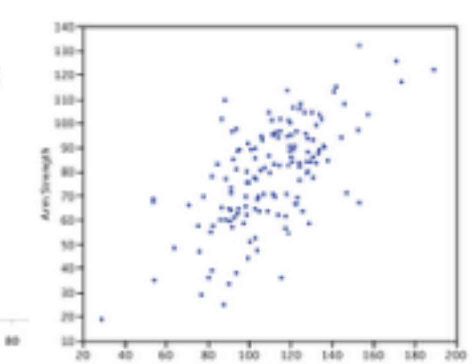


Figure 5. Scatter plot of Grip Strength and Arm Strength, r = 0.63.

Prediction

- final rating $p_{a,i}$ of active user a on item i after choosing a 's n neighbors $u \in \{1, 2, \cdots, n\}$ based on the similarity measure $w_{a,u}$

$$p_{a,i} - \overline{r}_a = \frac{\sum_{u=1}^{n} w_{a,u}(r_{u,i} - \overline{r}_u)}{\sum_{u=1}^{n} w_{a,u}}$$

- Can be generalized to Item-Item Collaborative Filtering

Pros and Cons

	Content Based	Collaborative Filtering
Pros	• no need for data on other users (no cold start or sparsity problem) • able to recommend to users with unique tastes • able to recommend new and unpopular items (no first-rater problem) • good explanations of recommended items	• works for any kind of item, no feature selection needed
Cons	• requires content that can be encoded using meaningful features • users' tastes (profiles) must be represented as a learnable function of these content features • unable to exploit quality judgements (ratings) of other users	• cold start: there needs to be enough other users in the systems to construct similar neighbors • sparsity: # of items is huge, hard to find users rated the same items • first rater: cannot recommend items that have no ratings yet • popularity bias: cannot recommend items to someone with unique tastes, tend to recommend popular items

Homophily

- The phenomenon that people are linked to similar others is called homophily

- Triadic closure, focal closure / selection (basketball introduce A to B), membership closure / social influence (my friend play guitar, so i start playing guitar)

- Schelling's segregation model: Micro-motives $\neq$ Macro-behavior

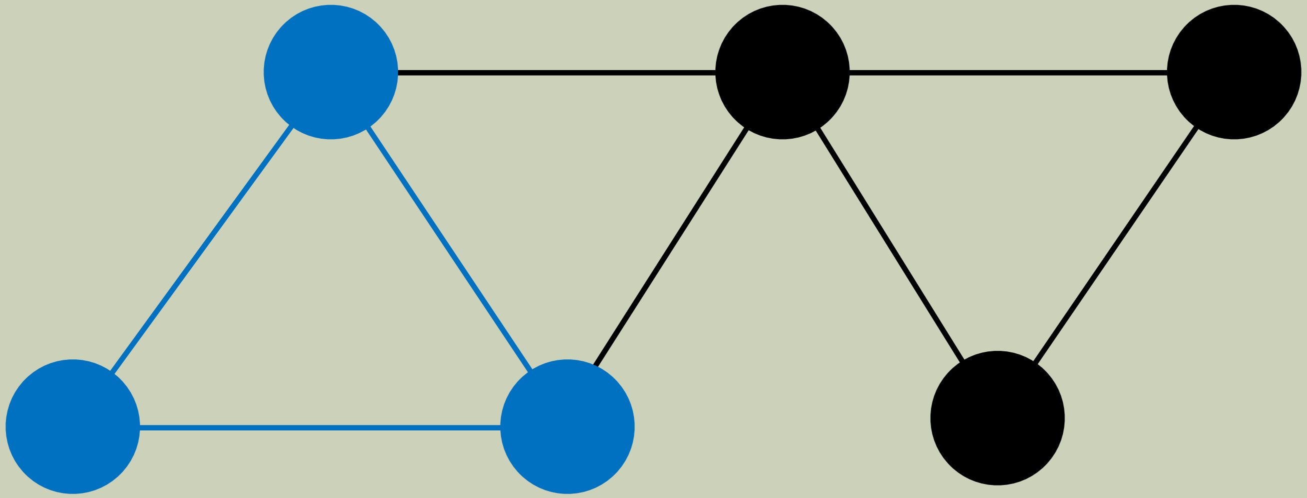
https://class.coursera.org/modelthinking-006/lecture/16

K-CLIQUE COMMUNITIES

- **Adjacent k-cliques**

Two k-cliques are adjacent when they share <u>k-1</u> nodes

k = 3



Clique 1

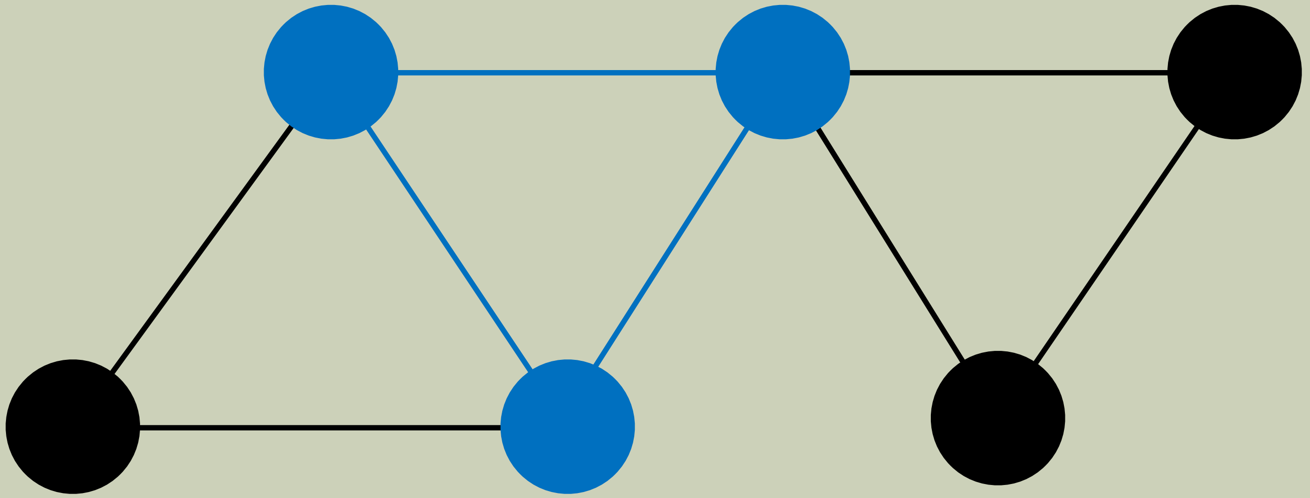
K-CLIQUE COMMUNITIES

- **Adjacent k-cliques**

Two k-cliques are adjacent when they share <u>k-1</u> nodes

Clique 2

k = 3



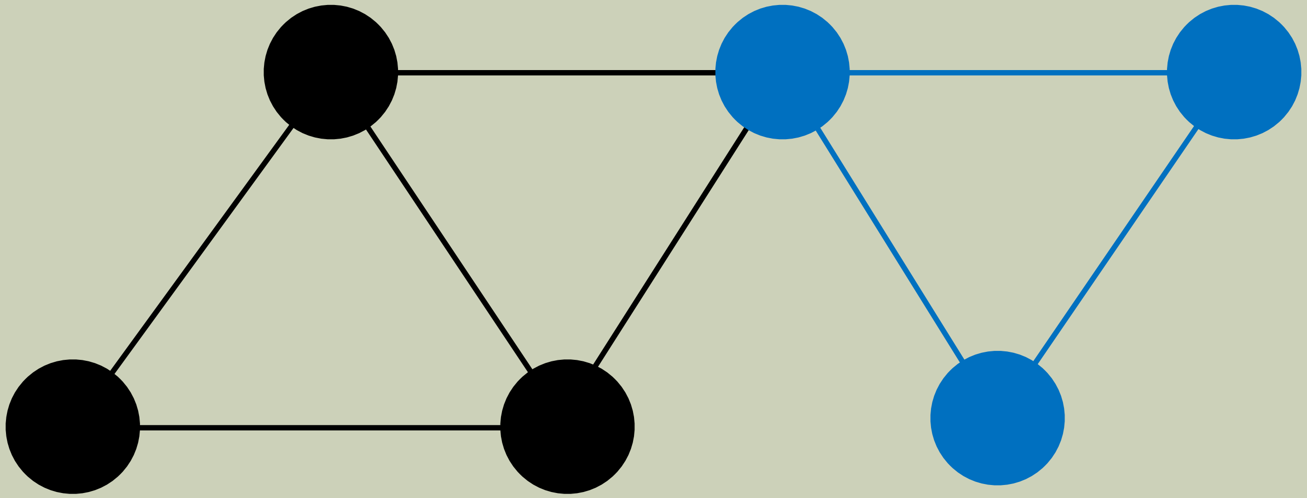
K-CLIQUE COMMUNITIES

- **Adjacent k-cliques**

Two k-cliques are adjacent when they share <u>k-1</u> nodes

k = 3

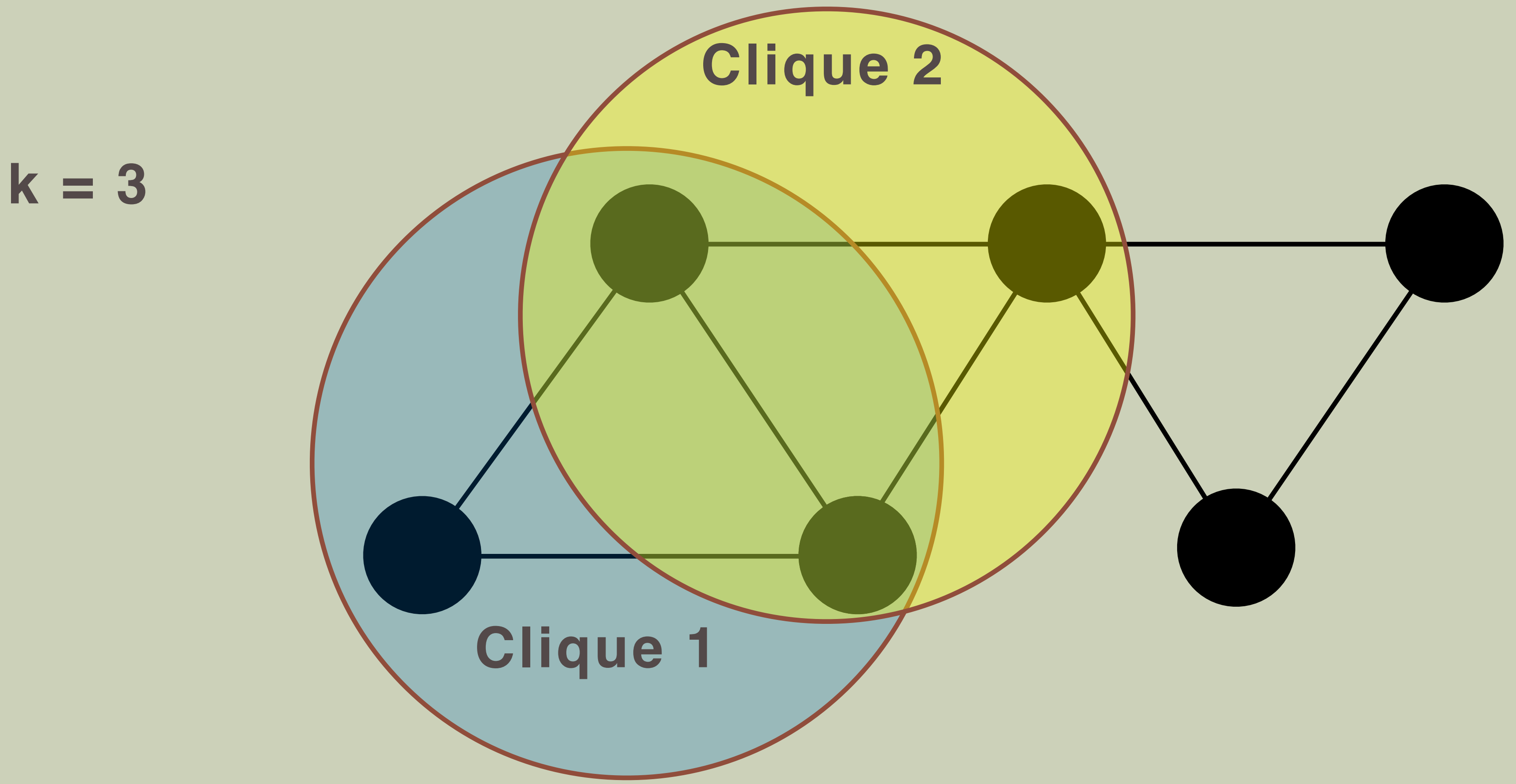
Clique 3



K-CLIQUE COMMUNITIES

- **k-clique community**

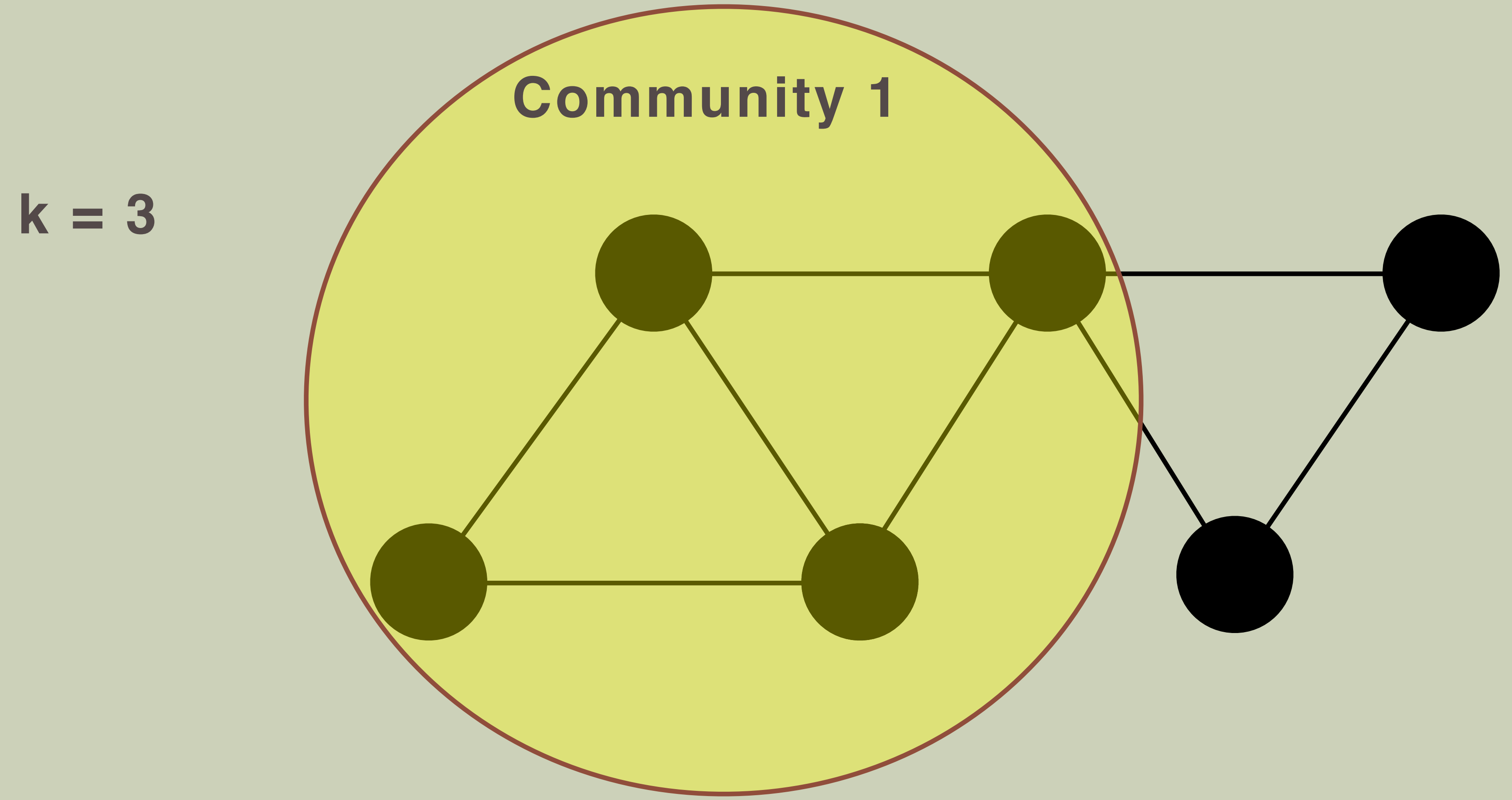
  Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques



K-CLIQUE COMMUNITIES

- **k-clique community**

  Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

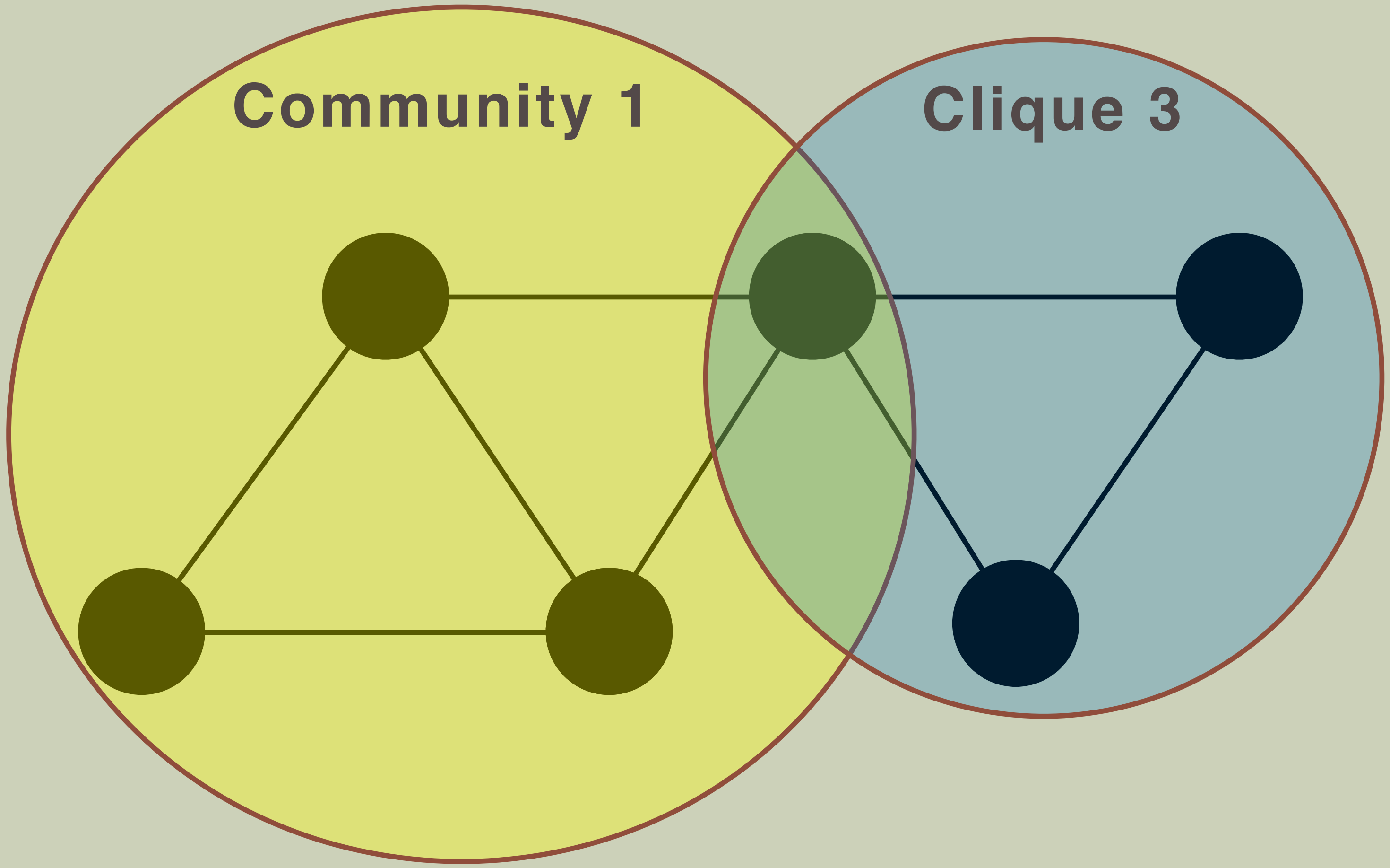


K-CLIQUE COMMUNITIES

- **k-clique community**

  Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

k = 3

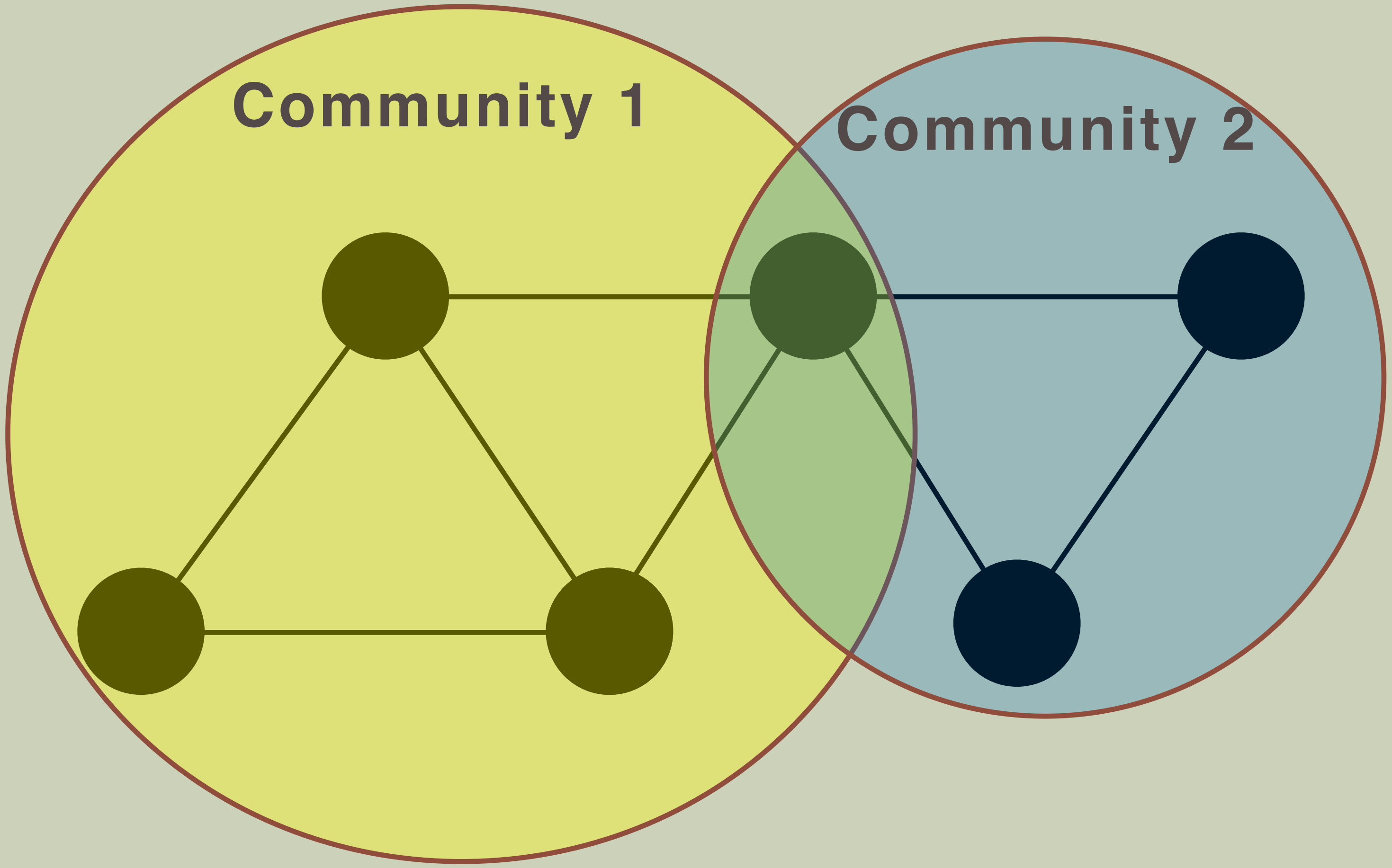


K-CLIQUE COMMUNITIES

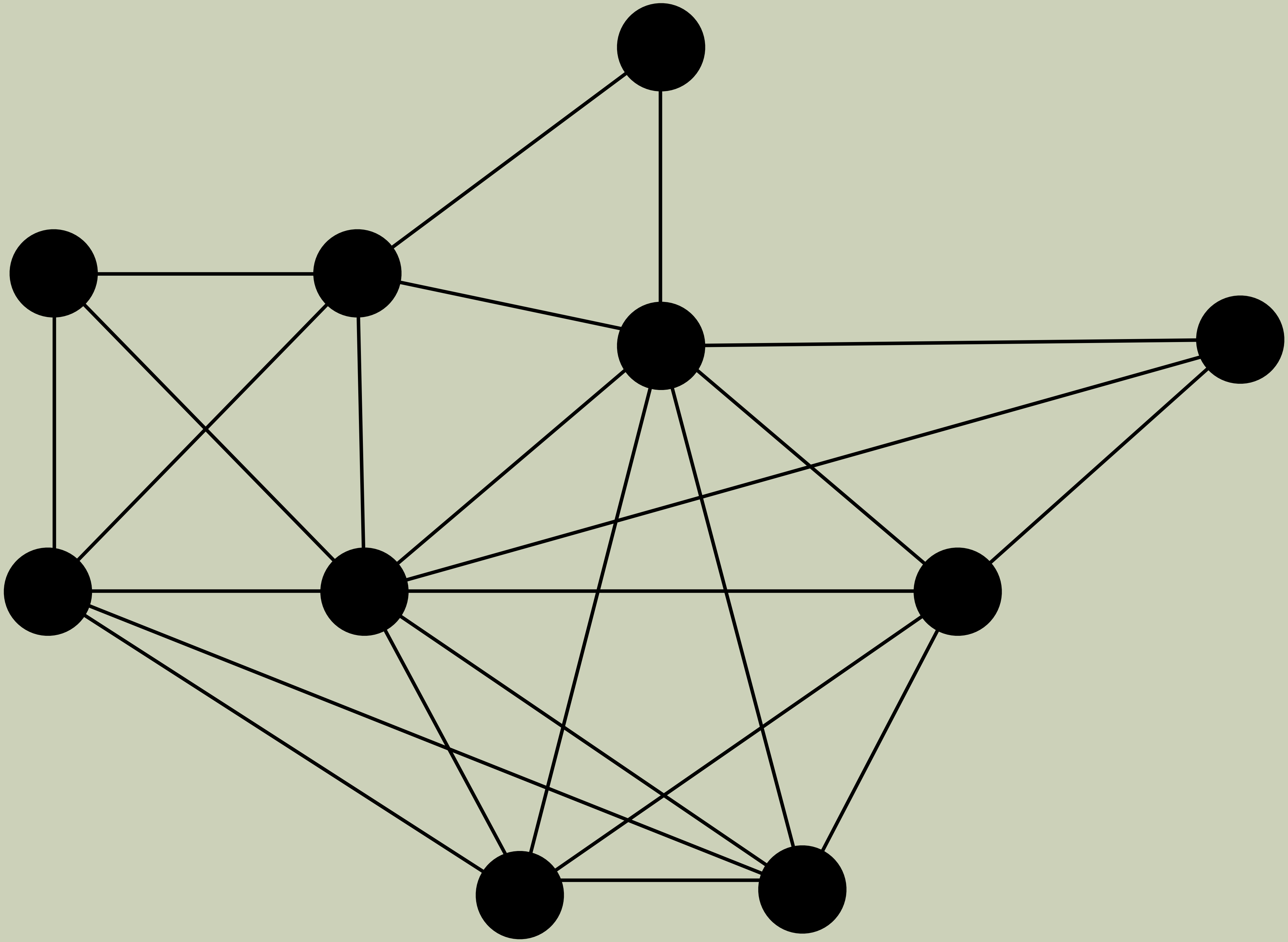
- **k-clique community**

  Union of all k-cliques that can be reached from each other through a series of adjacent k-cliques

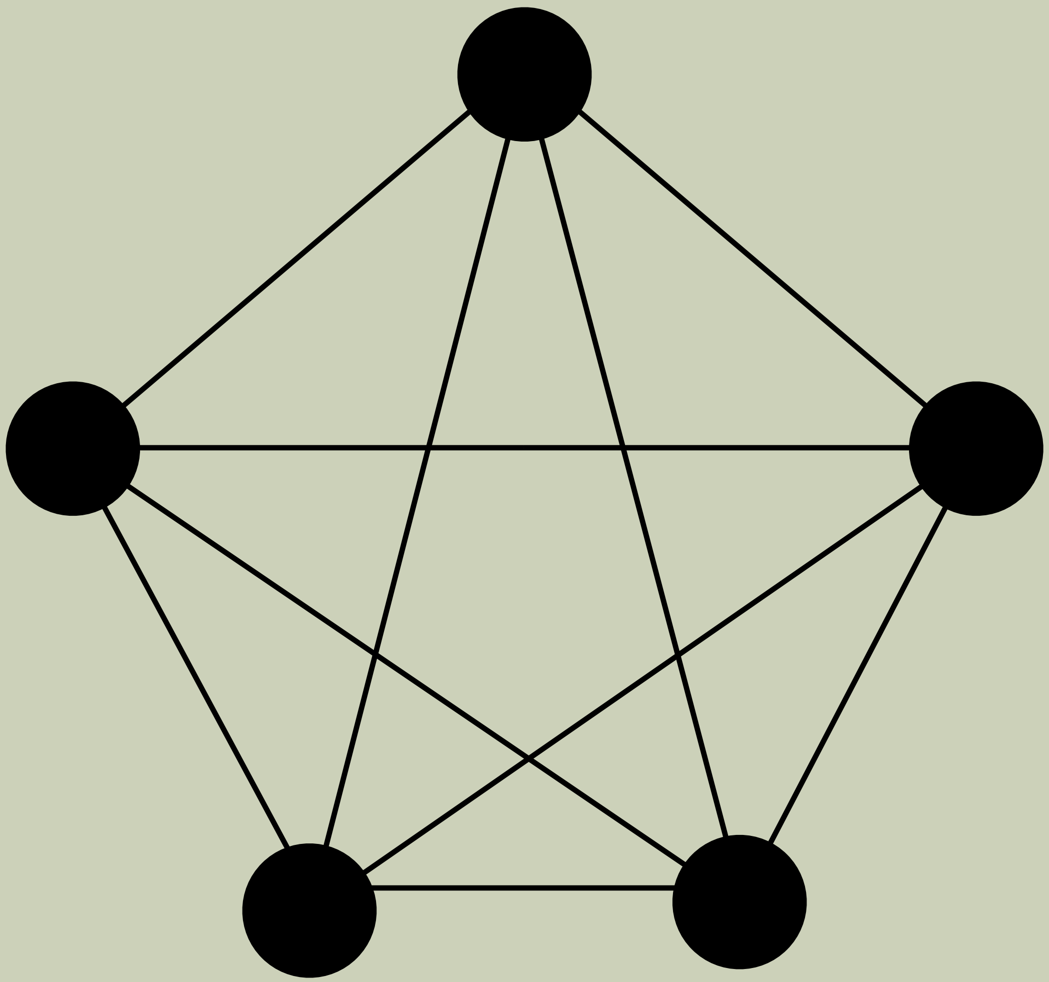
k = 3



CLIQUES TO K-CLIQUE COMMUNITIES

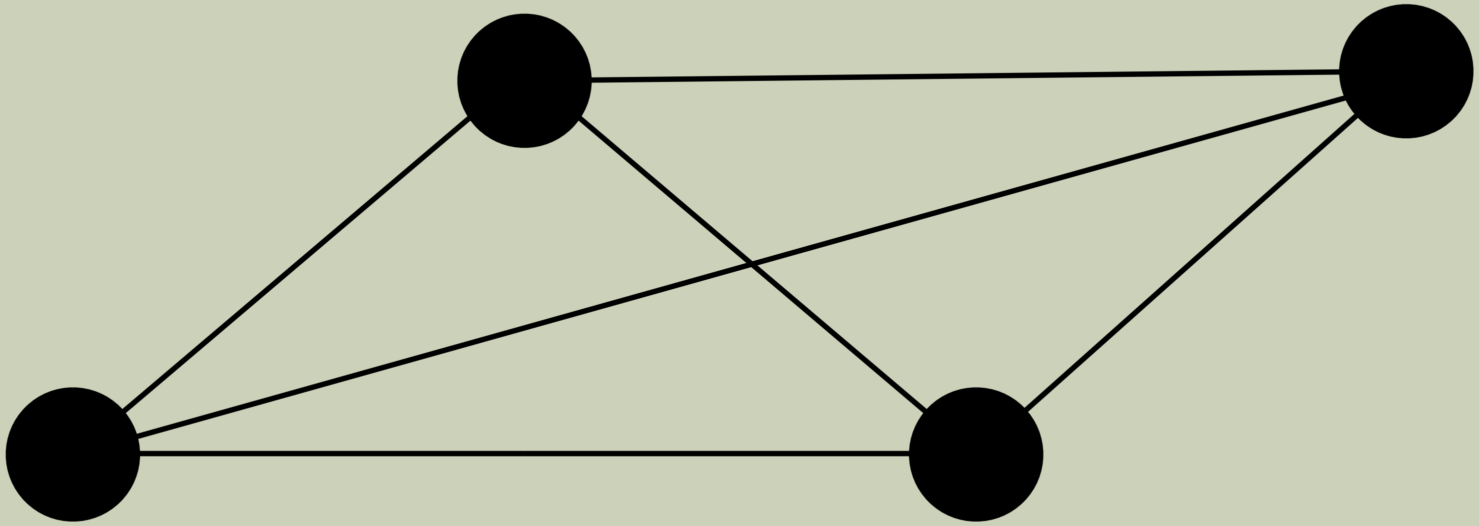


CLIQUES TO K-CLIQUE COMMUNITIES



Clique 1: 5-clique

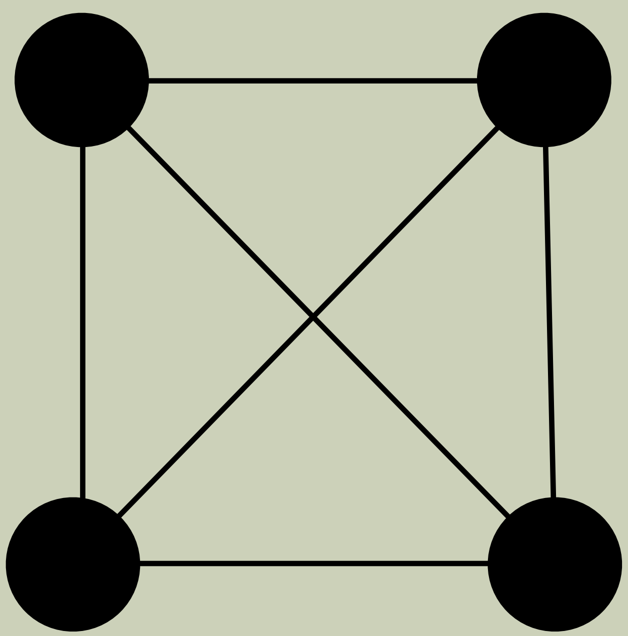
CLIQUES TO K-CLIQUE COMMUNITIES



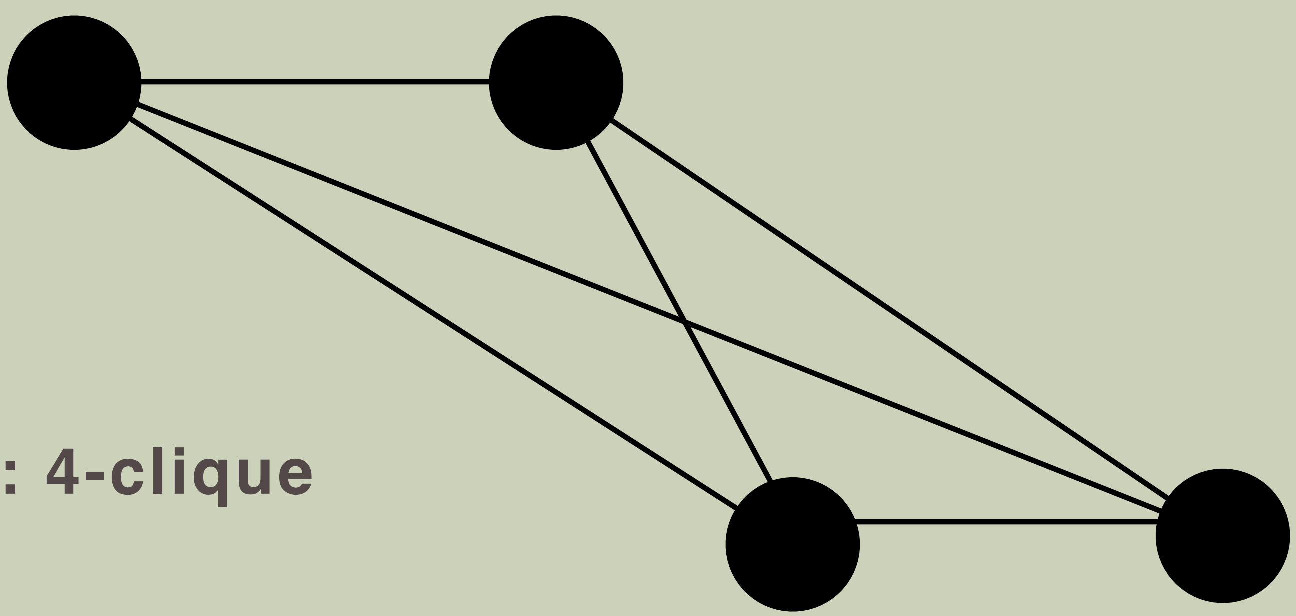
Clique 2: 4-clique

CLIQUES TO K-CLIQUE COMMUNITIES

Clique 3: 4-clique

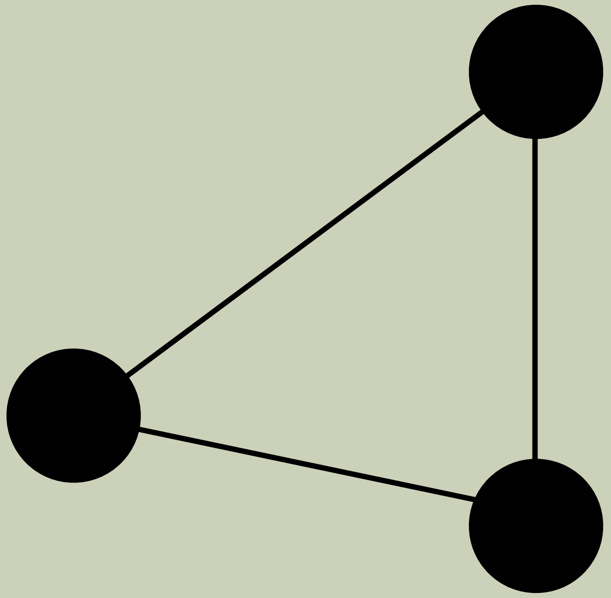


CLIQUES TO K-CLIQUE COMMUNITIES



Clique 4: 4-clique

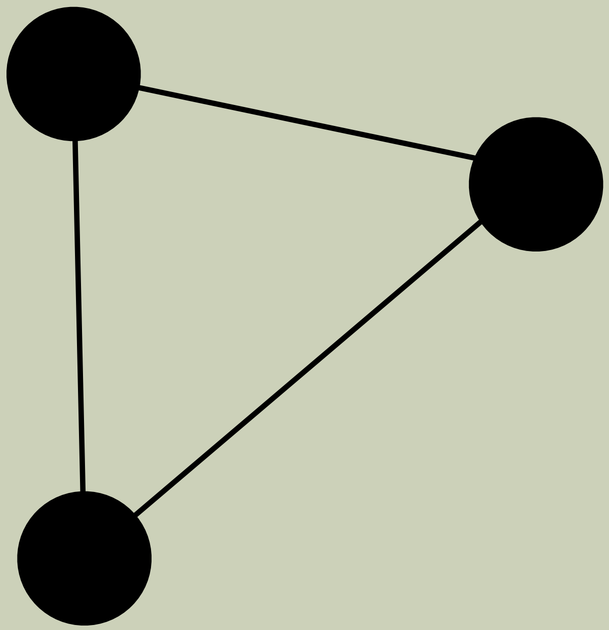
CLIQUES TO K-CLIQUE COMMUNITIES



Clique 5: 3-clique

CLIQUES TO K-CLIQUE COMMUNITIES

Clique 6: 3-clique



CLIQUES TO K-CLIQUE COMMUNITIES

	1	2	3	4	5	6
1	*5*					
2		4				
3			*4*			
4				4		
5					*3*	
6						3

CLIQUES TO K-CLIQUE COMMUNITIES

	1	2	3	4	5	6
1	*5*	*3*	*1*	*3*	*1*	*2*
2	3	4	1	1	1	2
3	*1*	*1*	*4*	*2*	*1*	*2*
4	3	1	2	4	0	1
5	*1*	*1*	*1*	*0*	*3*	*2*
6	2	2	2	1	2	3

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	3	2
6	2	2	2	1	2	3

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	0	2
6	2	2	2	1	2	0

Delete if less than k

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	5	3	1	3	1	2
2	3	4	1	1	1	2
3	1	1	4	2	1	2
4	3	1	2	4	0	1
5	1	1	1	0	0	2
6	2	2	2	1	2	0

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	5	3	0	3	0	0
2	3	4	0	0	0	0
3	0	0	4	0	0	0
4	3	0	0	4	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Delete if less than k-1

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	*5*	*3*	*0*	*3*	*0*	*0*
2	3	4	0	0	0	0
3	*0*	*0*	*4*	*0*	*0*	*0*
4	3	0	0	4	0	0
5	*0*	*0*	*0*	*0*	*0*	*0*
6	0	0	0	0	0	0

CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	0	0	0	0
3	0	0	1	0	0	0
4	1	0	0	1	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

Change all non-zeros to 1

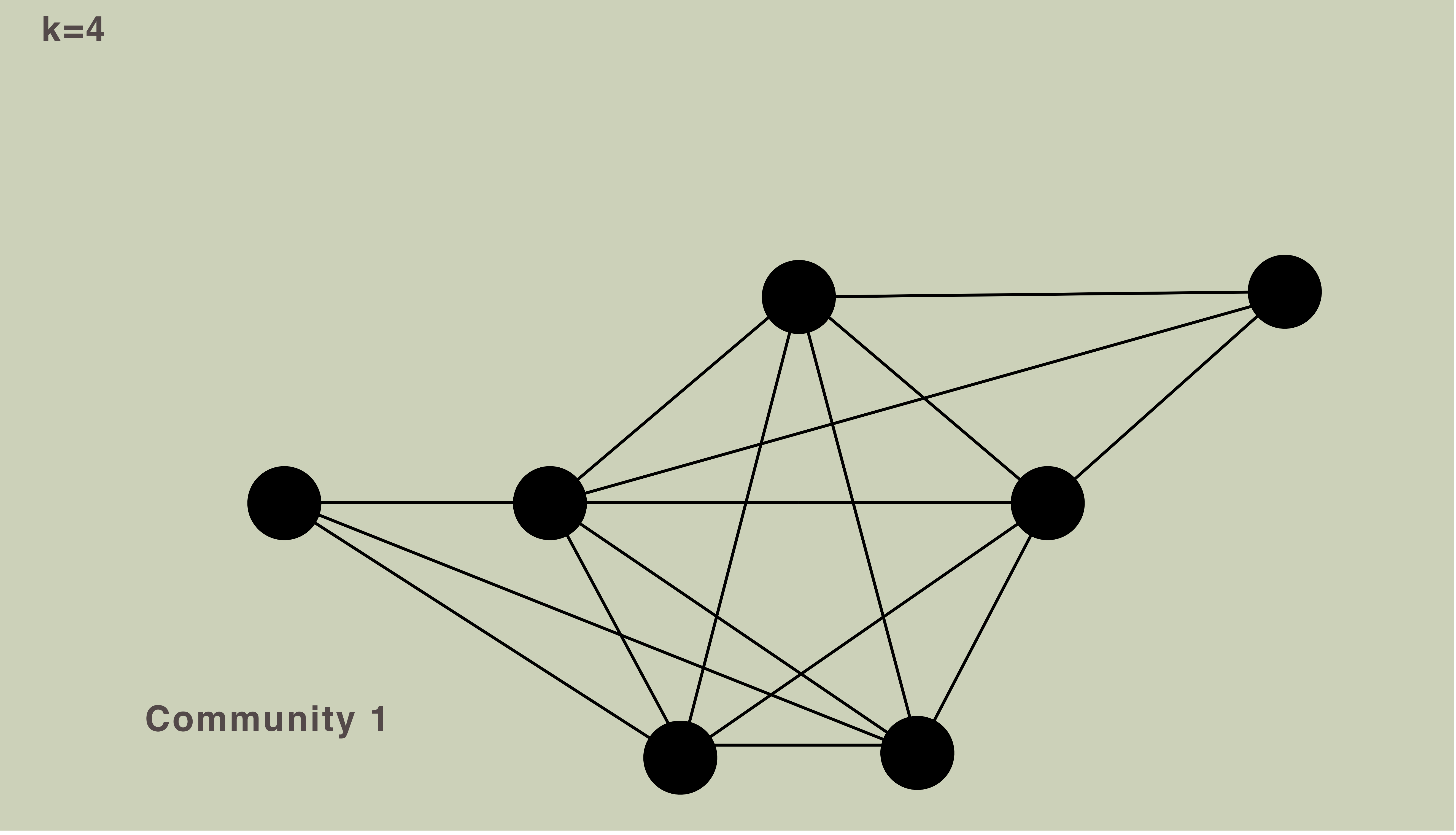
CLIQUES TO K-CLIQUE COMMUNITIES

k=4

	1	2	3	4	5	6
1	1	1	0	1	0	0
2	1	1	0	0	0	0
3	0	0	1	0	0	0
4	1	0	0	1	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

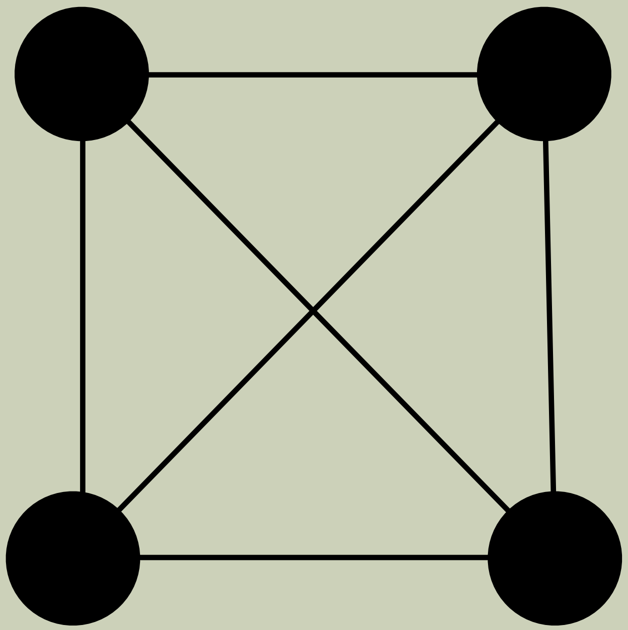
Clique-clique overlap matrix

CLIQUES TO K-CLIQUE COMMUNITIES



CLIQUES TO K-CLIQUE COMMUNITIES

k=4



Community 2