COMMUNITY DETECTION WITH TOPIC MODELLING

Presented by Mingyuan Zhang

OVERVIEW

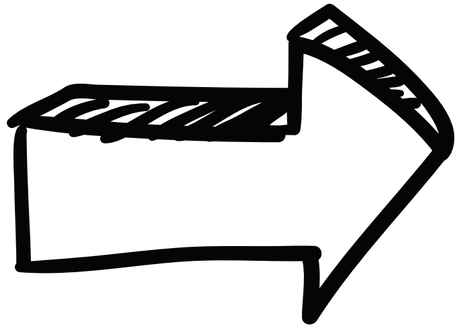
- Problem Statement
- Methodology
  - Community Detection
  - Topic Modelling
- Data Description
- Result

PROBLEM STATEMENT

Community detection

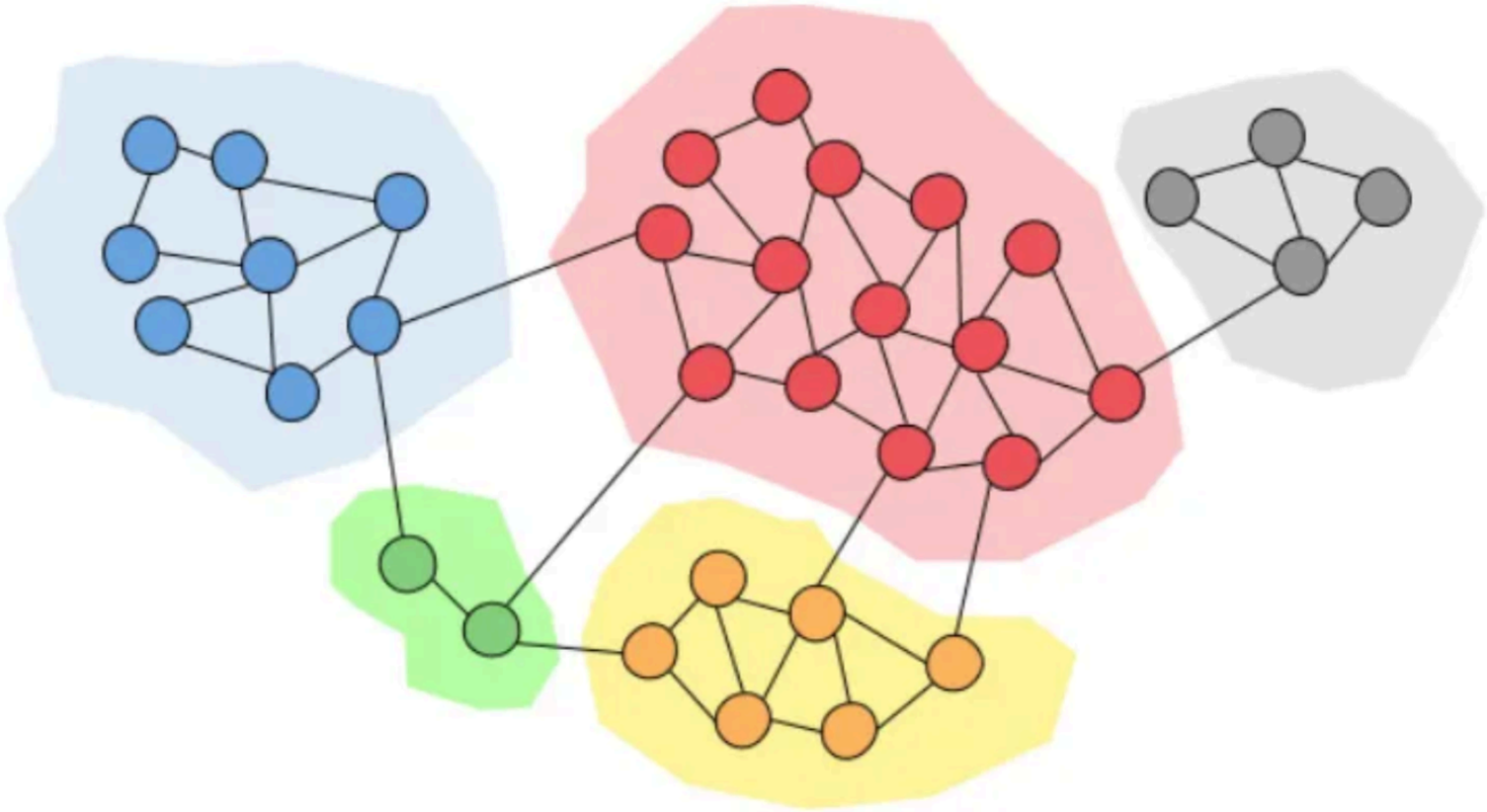
+

Topic modelling

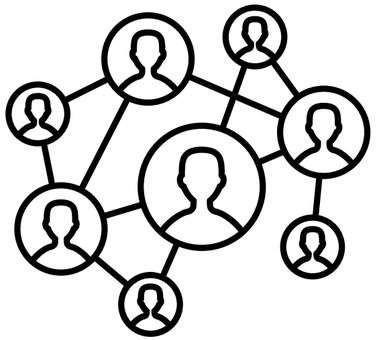


- **Better interpretation of the communities detected**
- **Discovering semantic similarities among communities**

Community detection

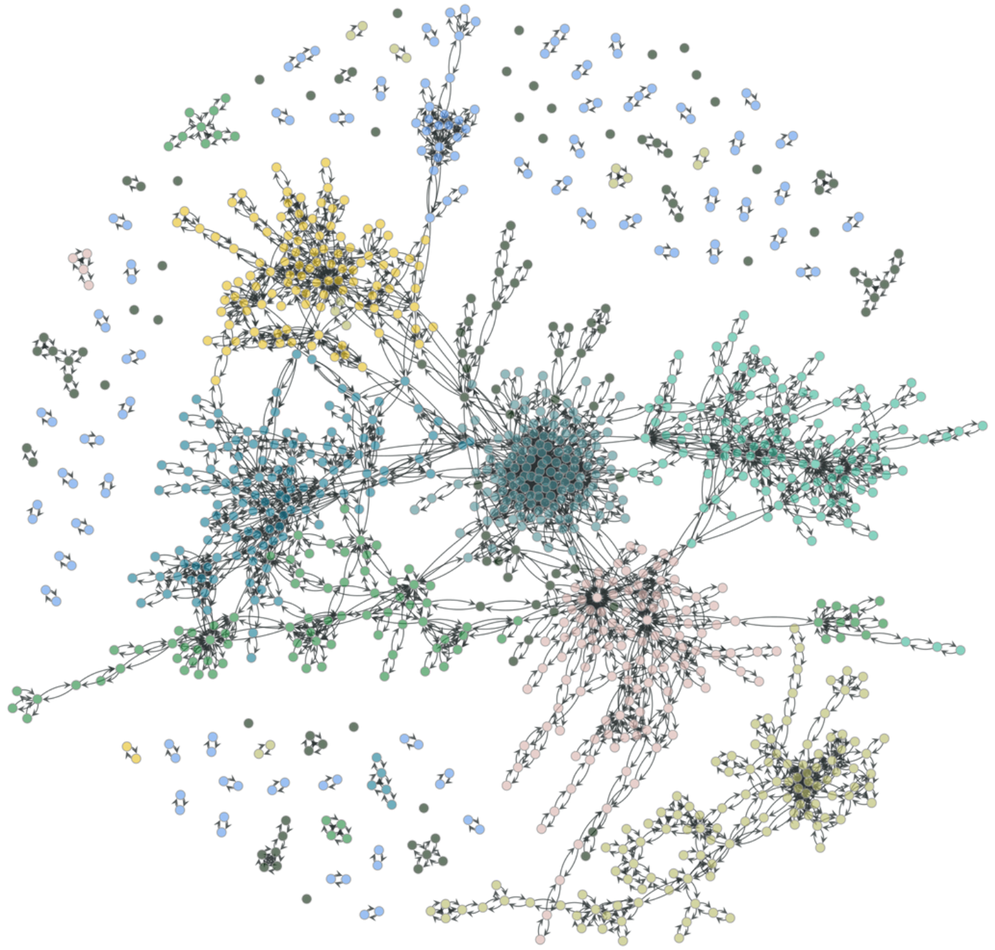


Comunity detection



- Often Applied to graphs:
  - nodes = entities,
  - edges = relationships
- Communities = groups of nodes with dense internal connections, sparse external ones
- Covariates = extra node features (e.g., text, metadata)
- Can improve downstream tasks such as recommendation

Stochastic Block Model



Stochastic Block model

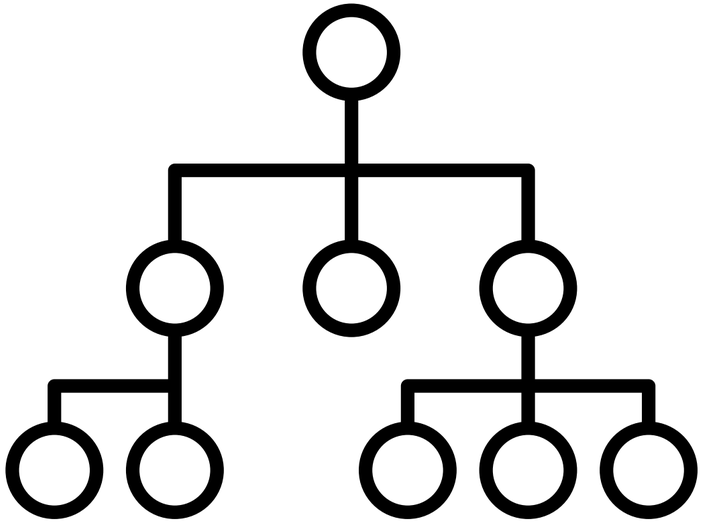
- A generative model for graph networks
- Assumes that nodes are divided into blocks or groups
- The likelihood of a connection between two nodes depends only on their group memberships.
- Python package: Graph-tools is used for this project.

Topic modelling

TOPIC MODELLING

- Clustering algorithm but for text
- Helps summarize unstructured text data
- Common approaches:
  - LDA
  - hLDA
  - NMF

Hierarchical Latent Dirichlet Allocation(hLDA)



hLDA

- A Bayesian non-parametric Inference technique
- Uncover multi-level topic structures, similar to real-word topics
- Sample output:

🔬 Sciences
- 🧪 Chemistry
  - Organic
  - Inorganic
- 🌌 Physics
  - Astronomy
  - Quantum Physics

🧿 Quantum Physics
[quantum, particle, entanglement, uncertainty, spin, ...]

Prior: Nested Chinese Restarurant Process

Chinese Restaruant Process(CRP)

- Non-parametric clustering
- Imagine a customer joining a chinese restaurant, he has 2 options:
  - Join an existing table
  - Sit at a new table

$$p(\text{occupied table } i) = \frac{m_i}{\gamma + m - 1}$$

$$p(\text{new table}) = \frac{\gamma}{\gamma + m - 1}$$

- γ contols how likely the costomer joins a new table

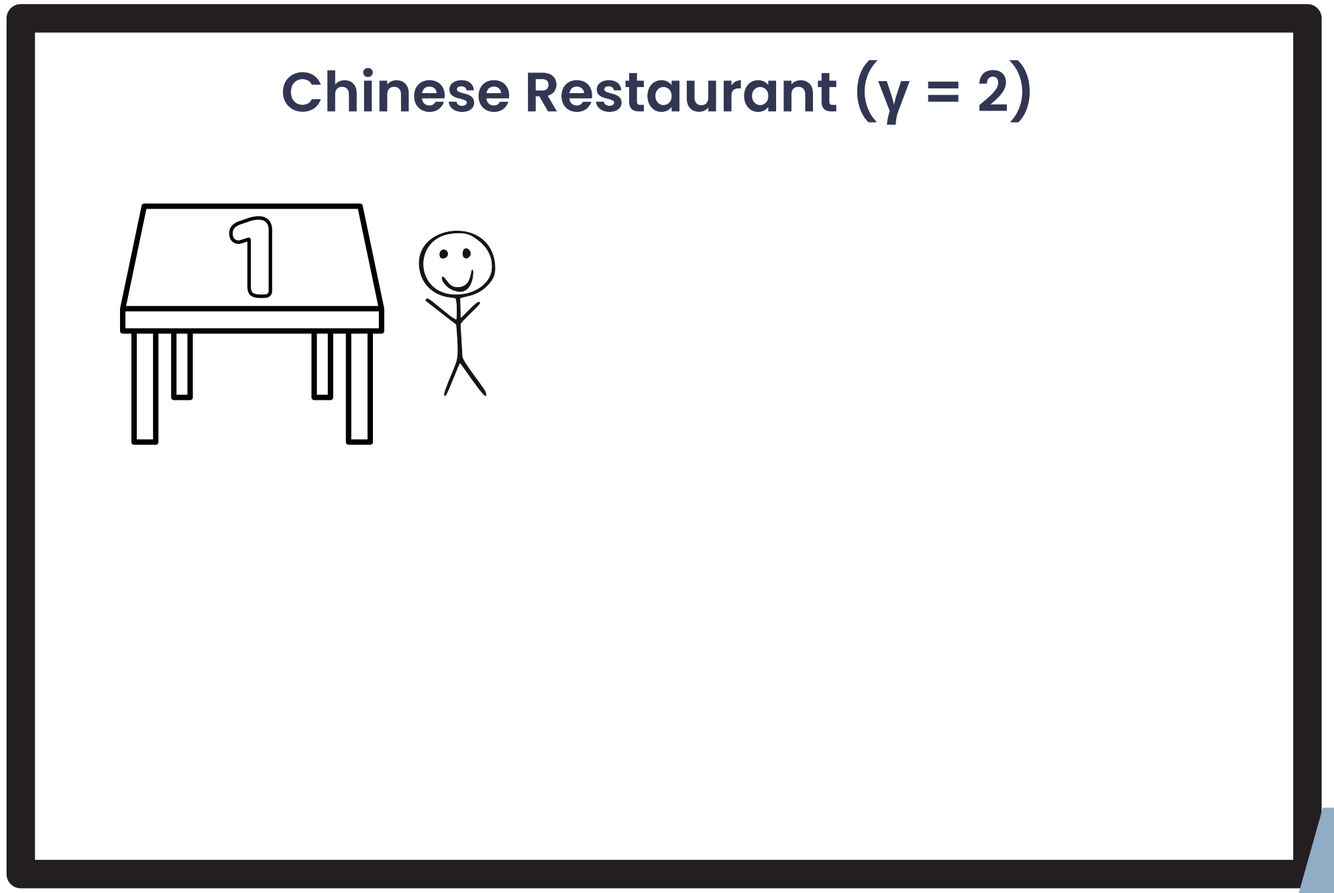
Chinese Restaruant Process(CRP) - Prior

First customer

P(new table) =γ/ (γ+m−1)
=2/ (2+1−1)
= 1

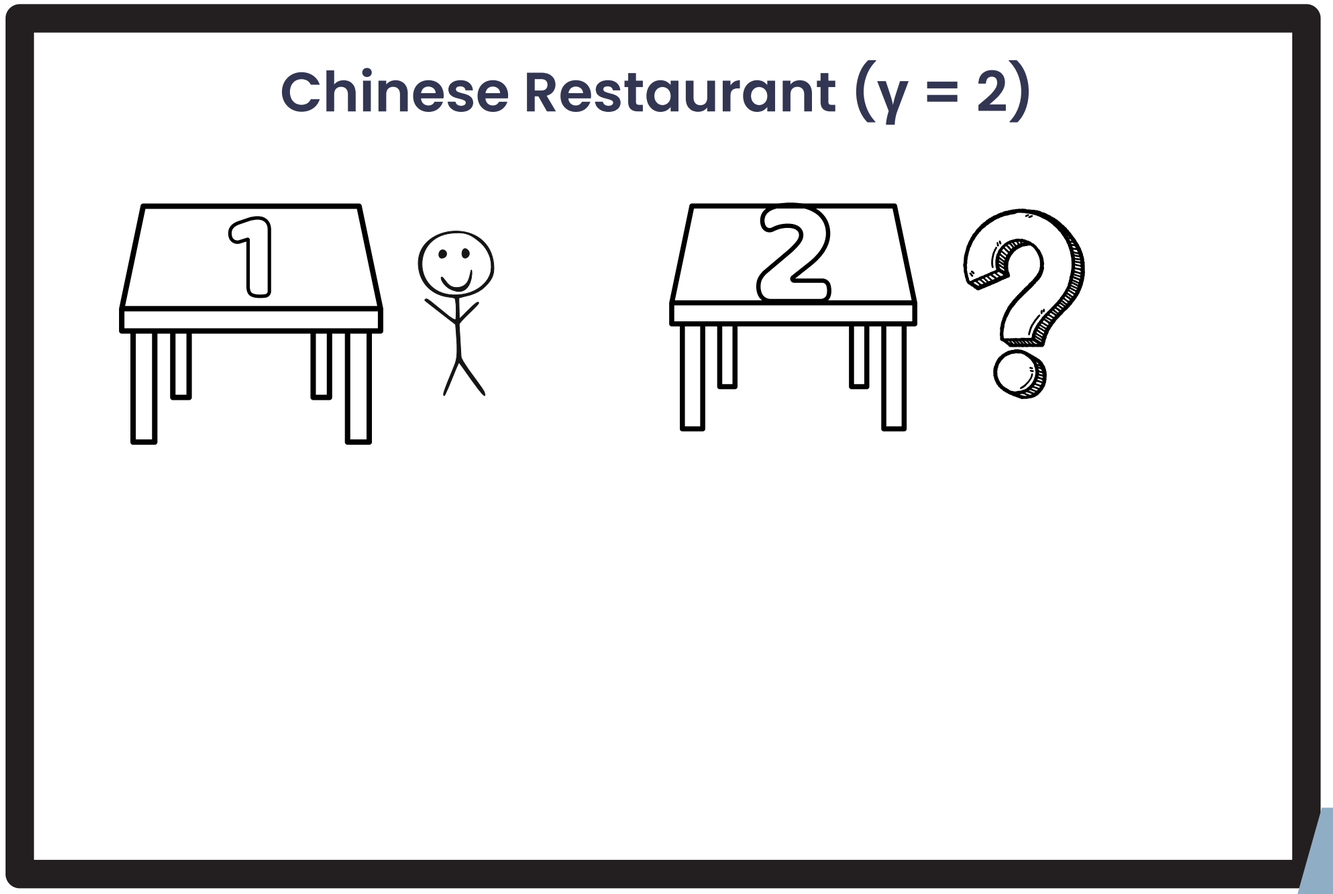
Chinese Restaurant (γ = 2)

Chinese Restaruant Process(CRP) - Prior



Chinese Restaruant Process(CRP) - Prior

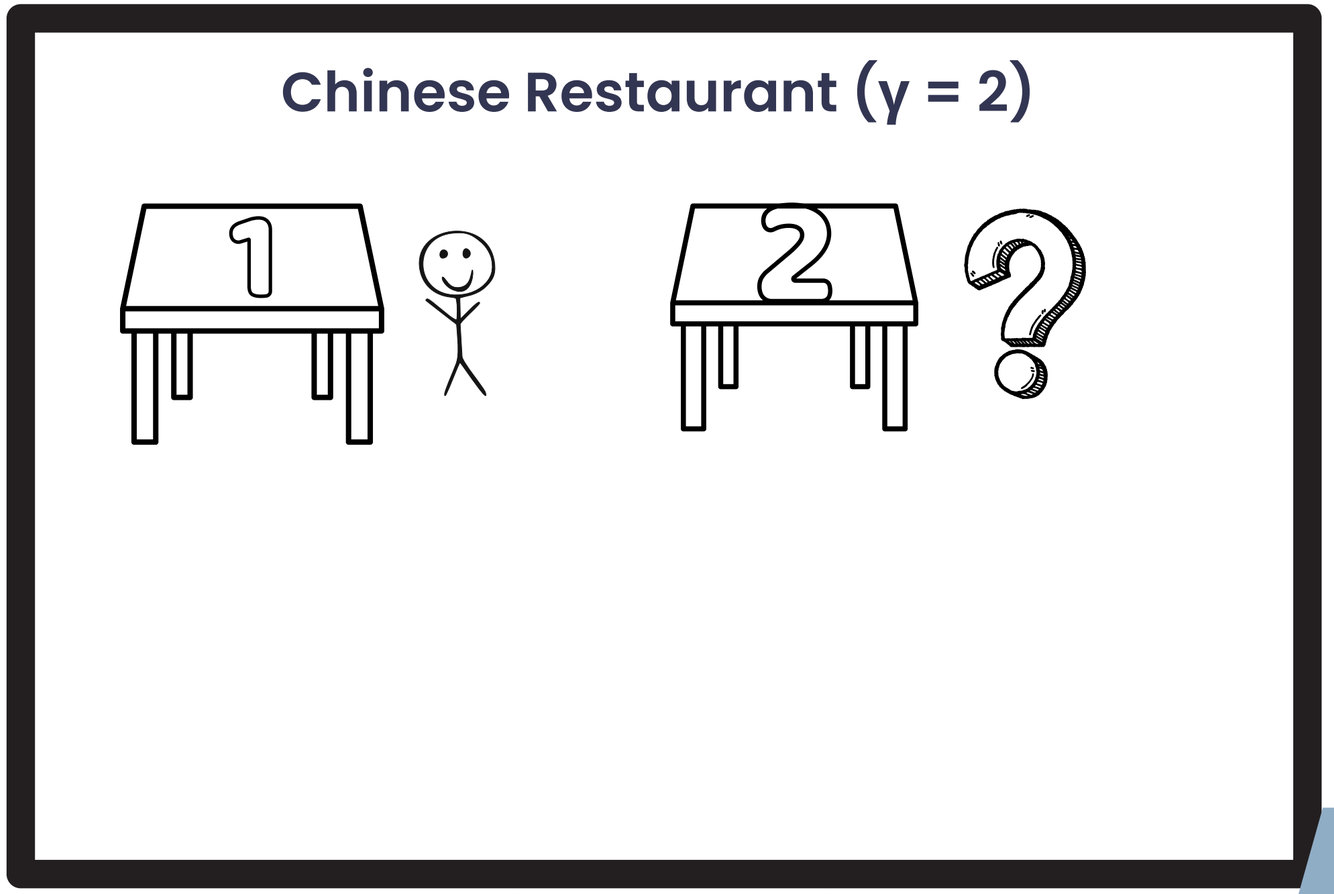
Second customer



Chinese Restaruant Process(CRP) - Prior

Second customer

P(new table) =y/ (y+m-1)
=2/ (2+2-1)
=2/ 3



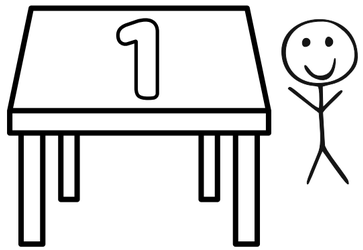
Chinese Restaruant Process(CRP) - Prior

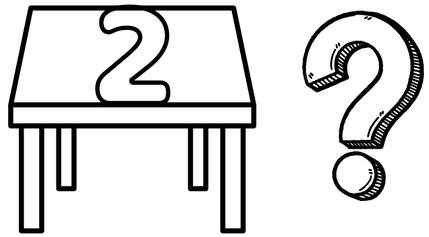
Second customer

P(new table) =2/3

P(join table 1)=m_1/ (y+m-1)
=1/ (2+2-1)
=1/ 3

Chinese Restaurant (y = 2)





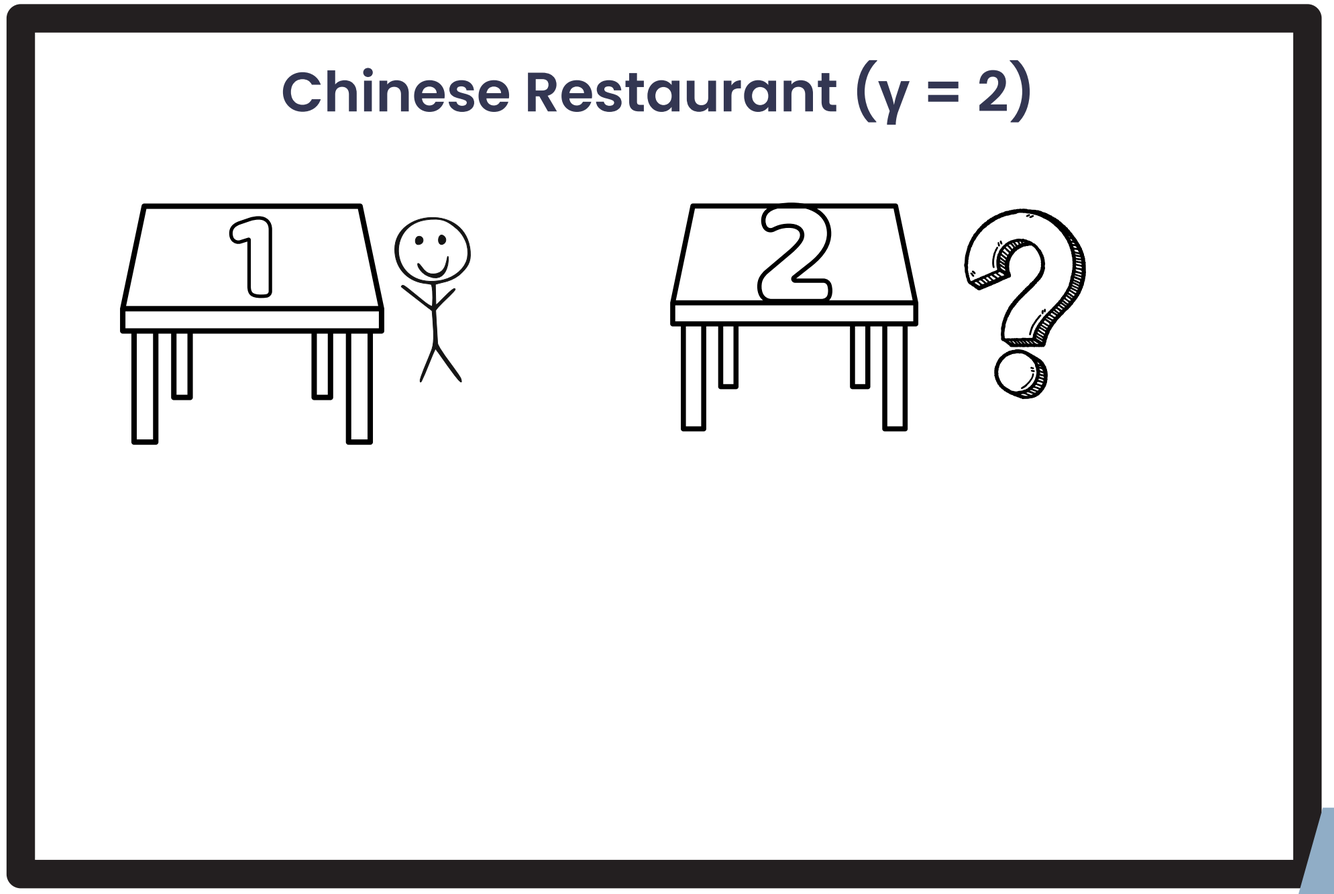
Chinese Restaruant Process(CRP) - Prior

Second customer

P(new table) =2/3

P(join table 1)=1/3

Randomly sample from Unif(0,1) to decide!



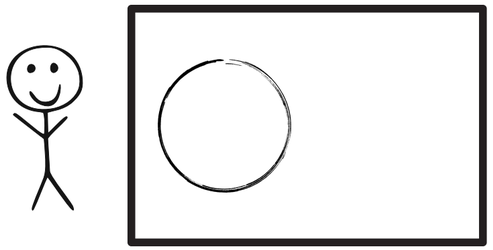
Chinese Restaruant Process

- **Number of table can be infinite!**

- **We obtain a distribution of partitions of customers**
  - **Used as a non-parametric prior**
  - **because we don't know how many possible clusters there are in advance!**

- **Now imagine not just one restaurant… but a whole hierarchy of restaurants ==> Nested Chinese Restaruant Process!**

Nested Chinese Restaruant Process

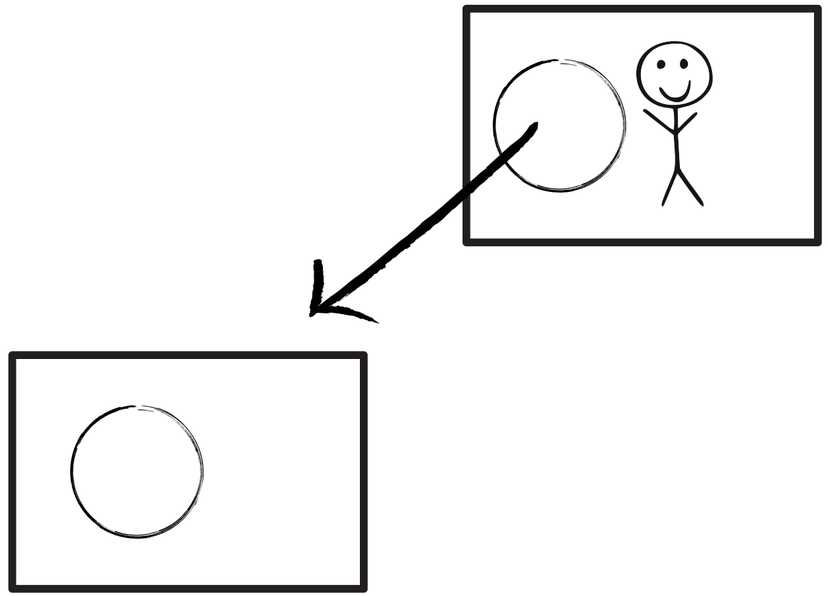


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

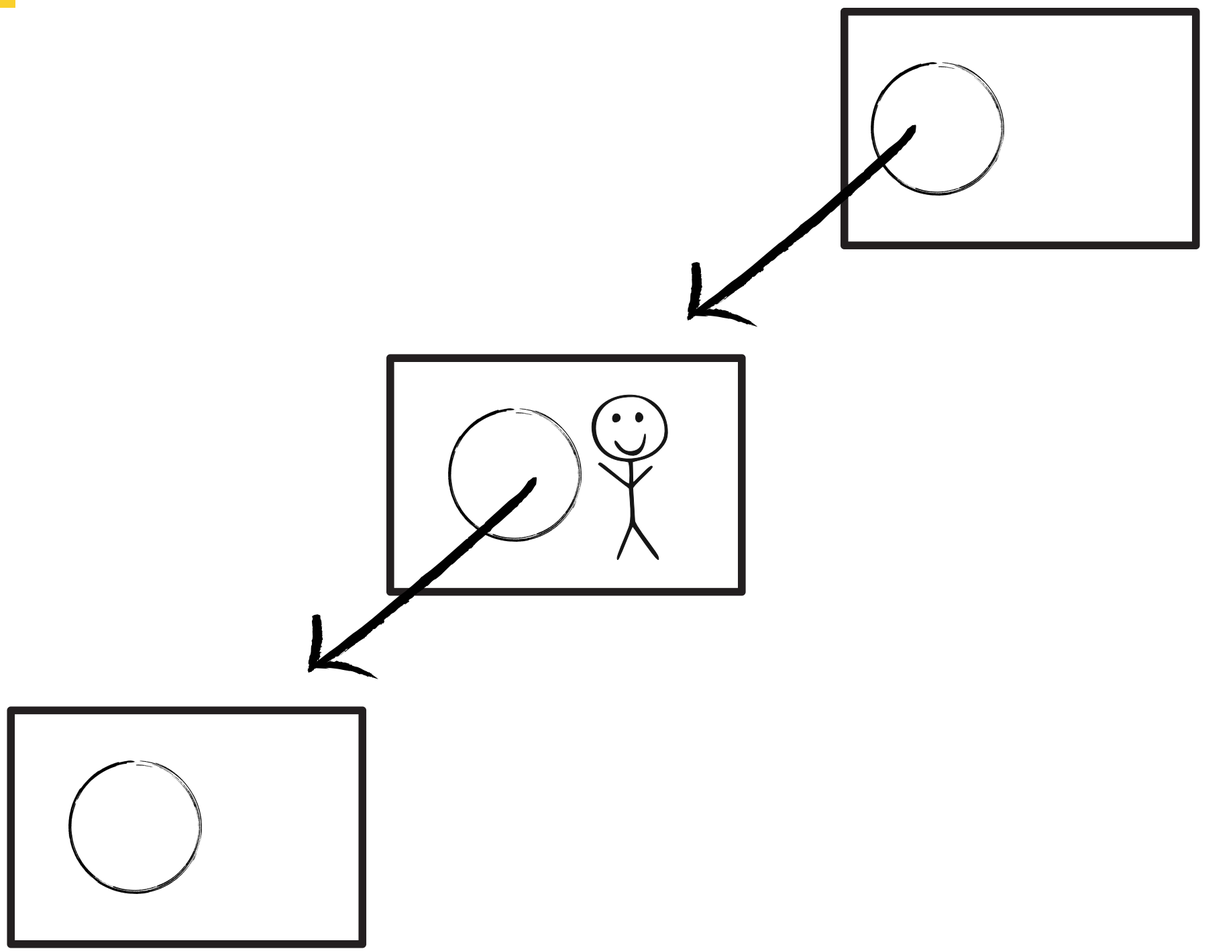


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

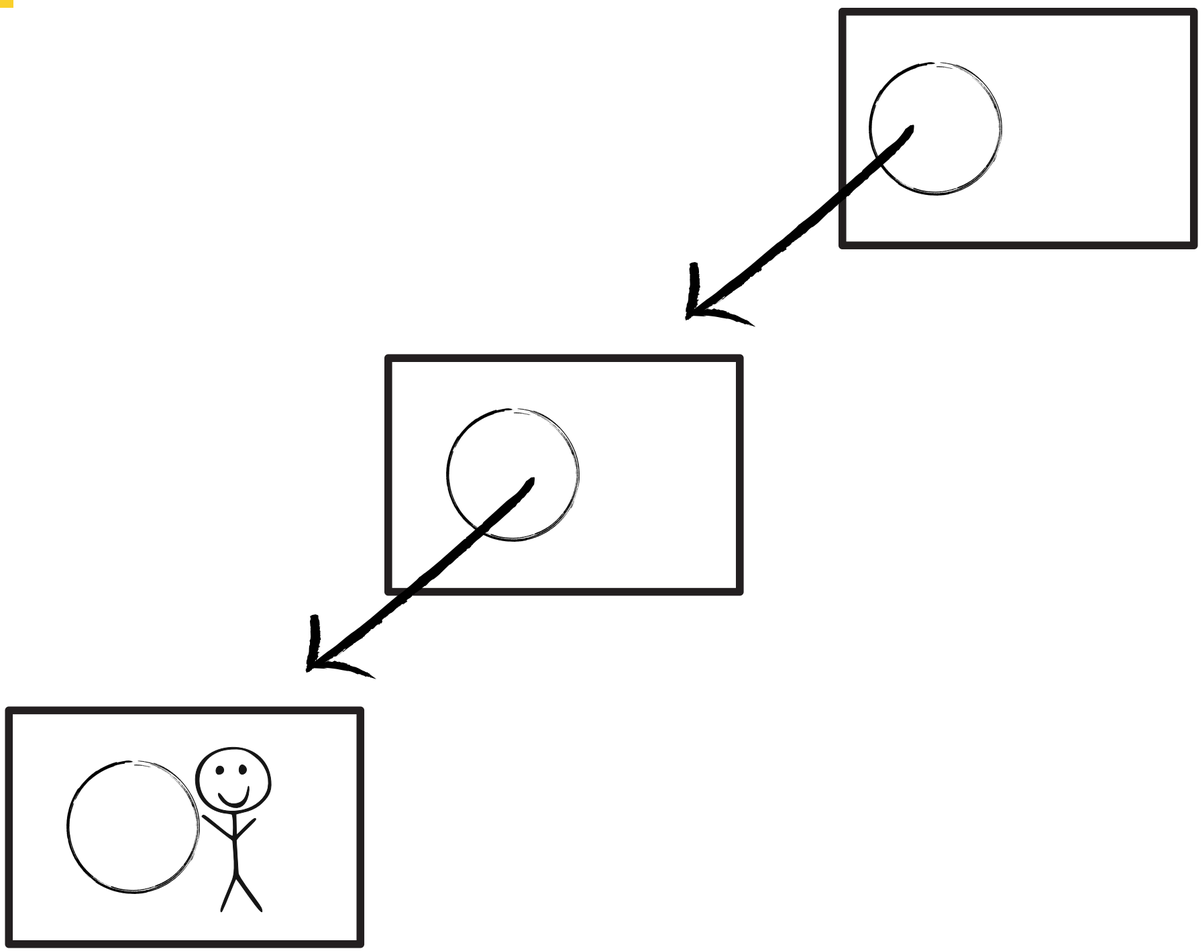


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

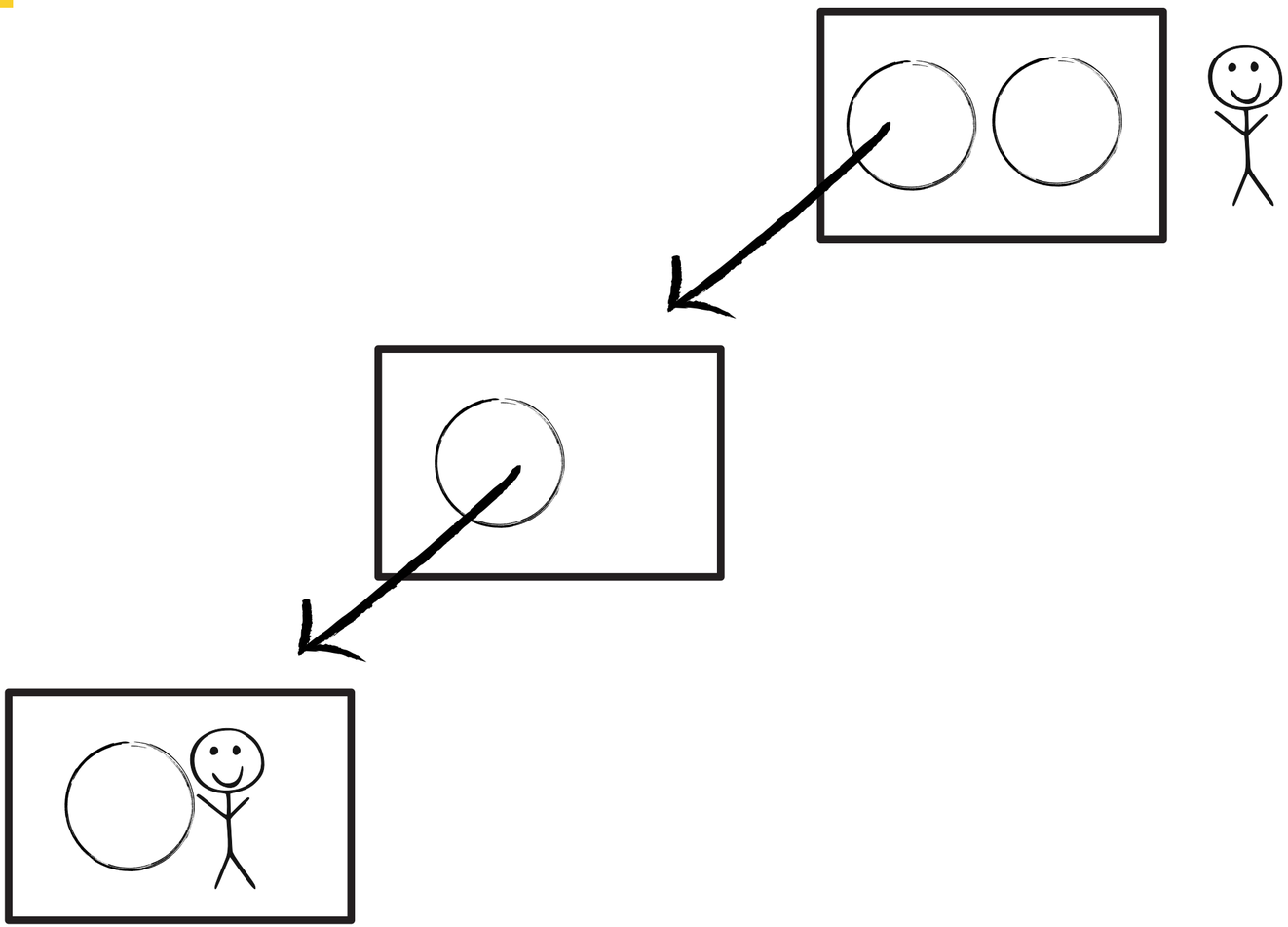


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

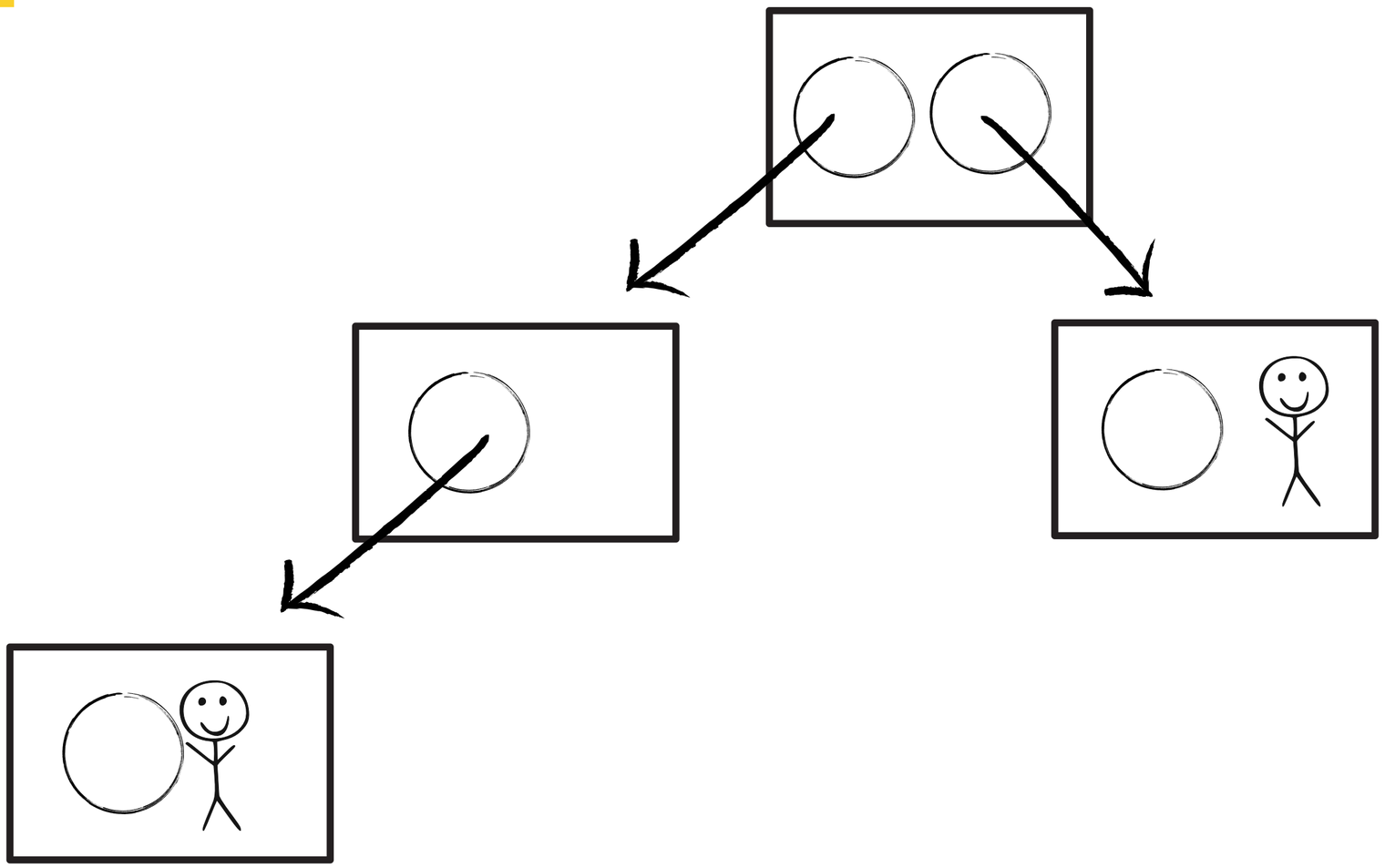


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

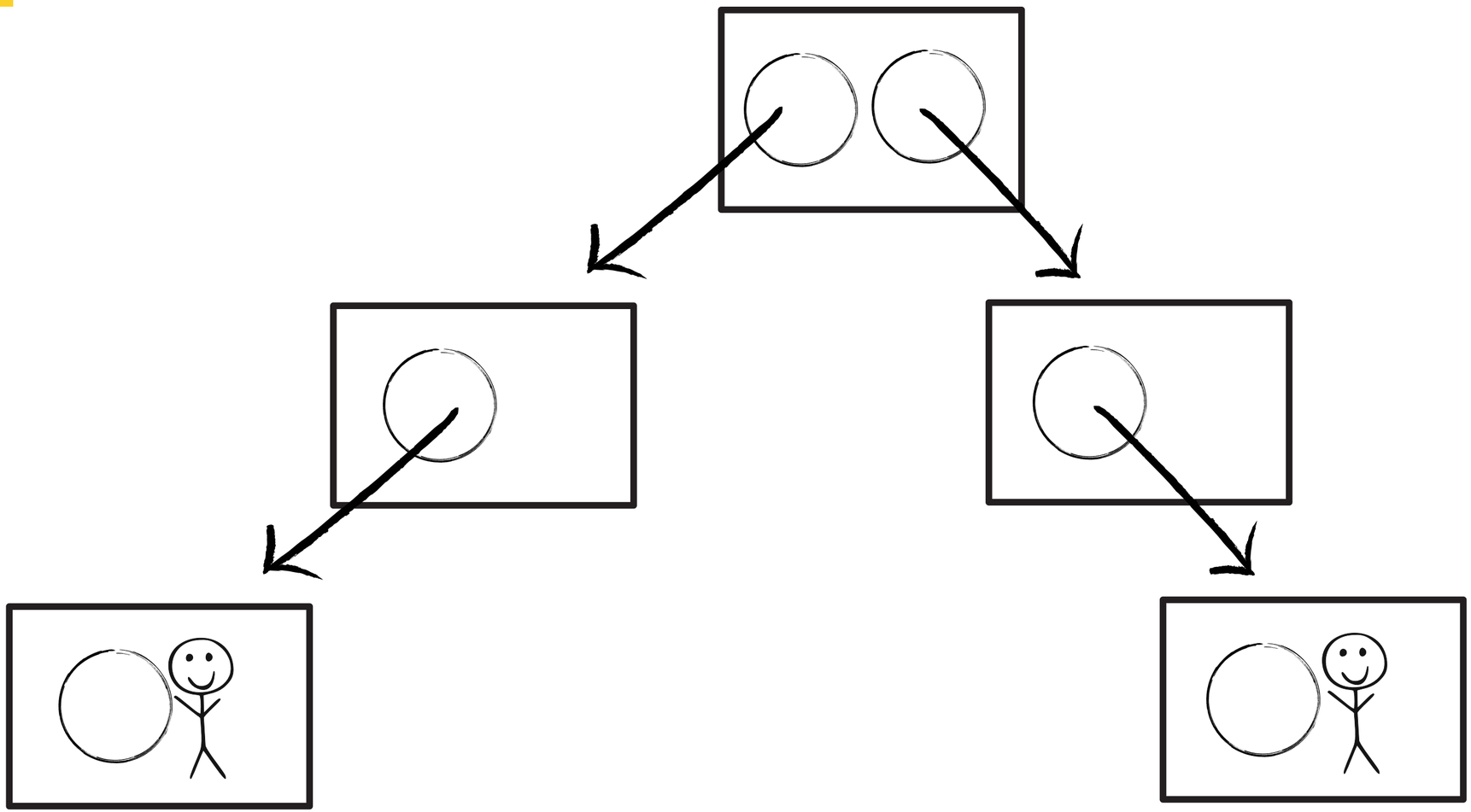


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

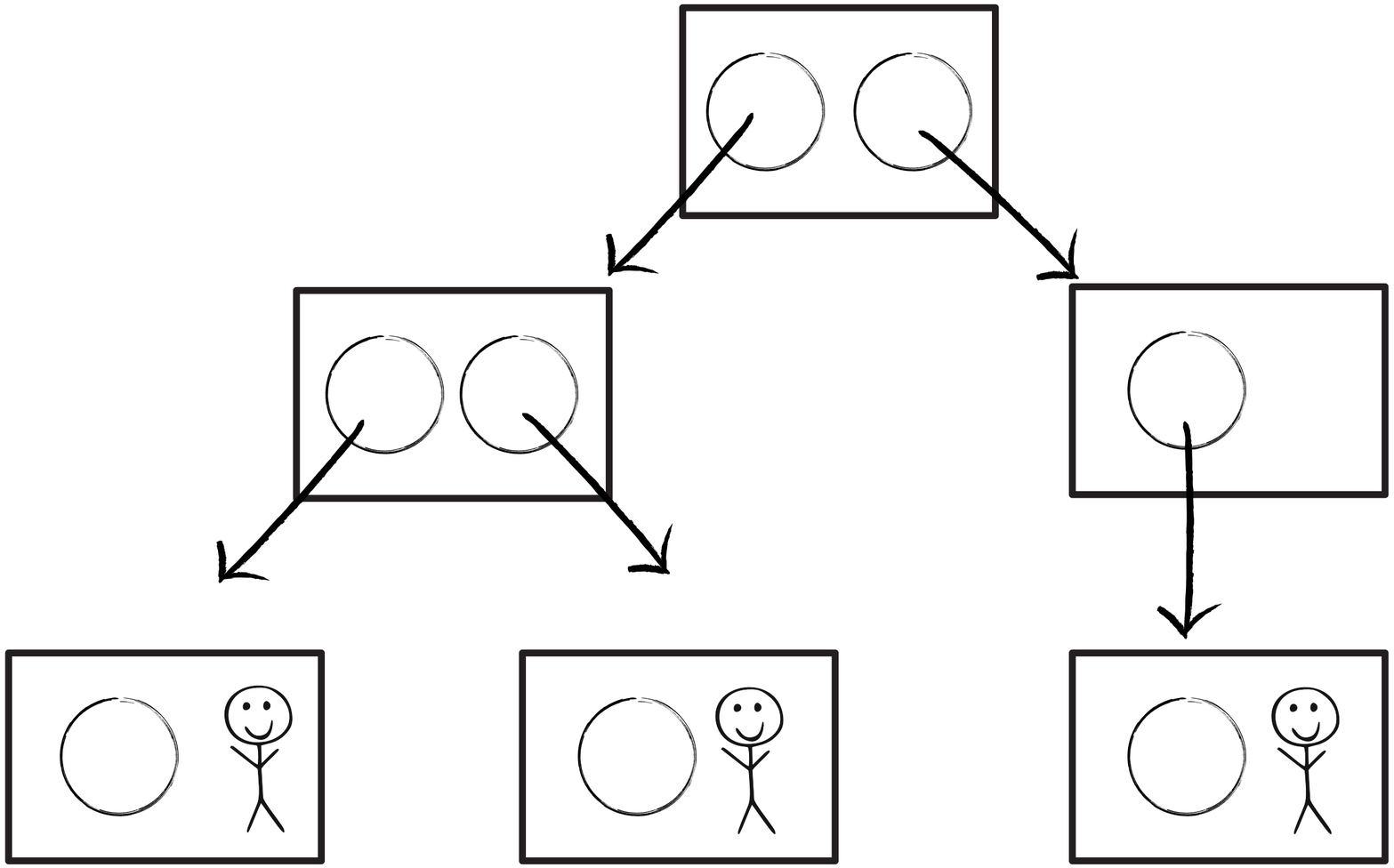


Day 1

Day 2

Day 3

Nested Chinese Restaruant Process

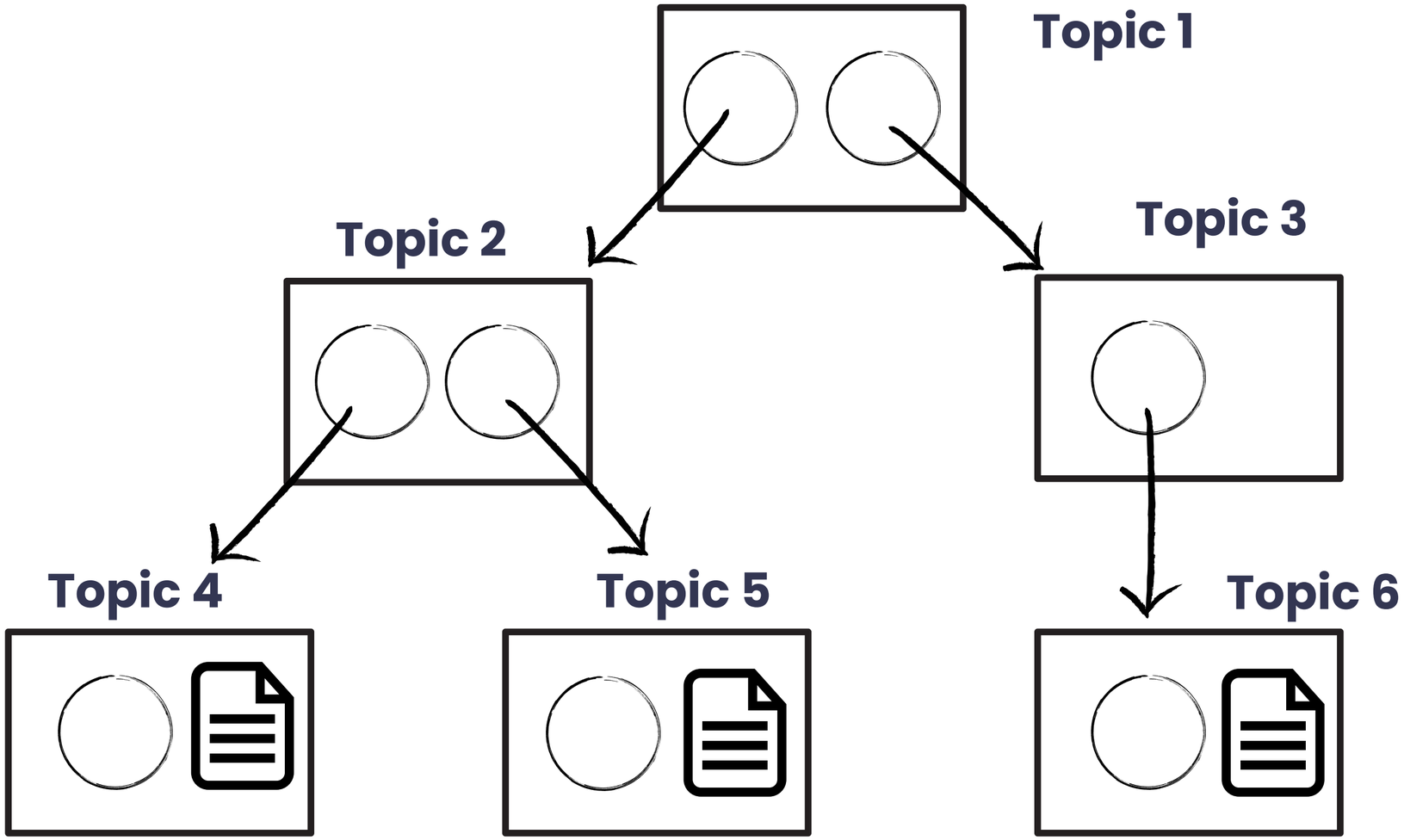


Day 1

Day 2

Day 3

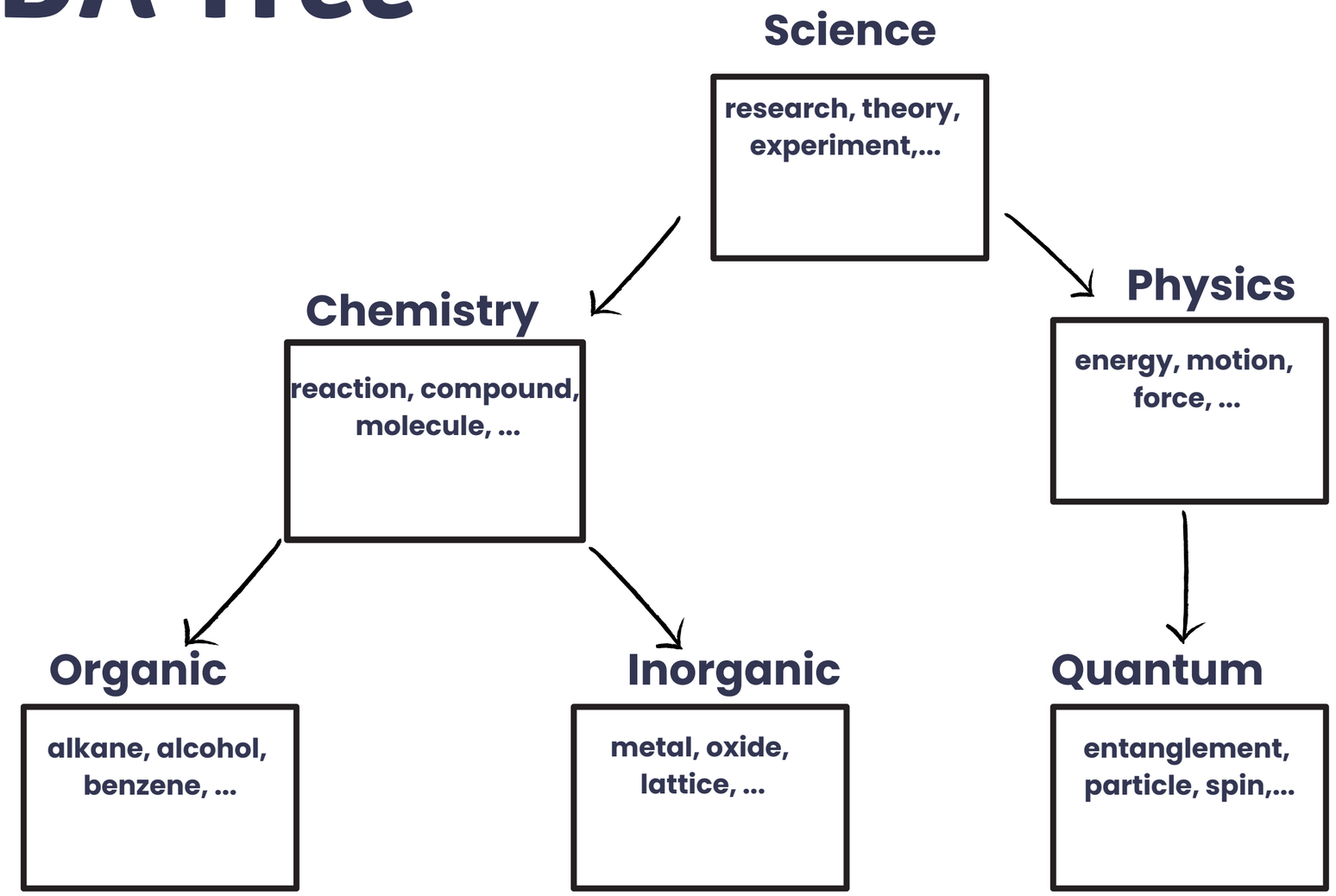
Nested Chinese Restaruant Process



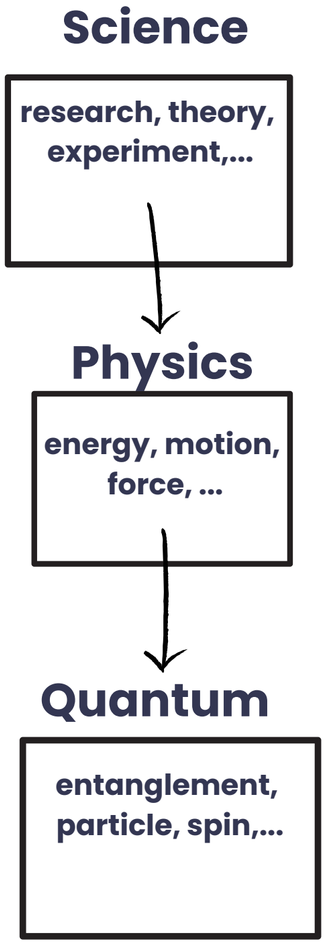
Nested Chinese Restaurant Process

- **Nonparametric!**
- **The tree can be infinitely branched and infinitely deep**
- **Each document chooses a path down the tree, selecting one topic per level**
  - More general topics will be closer to the root.

hLDA Tree


Science
research, theory, experiment,...
Chemistry
reaction, compound, molecule, ...
Physics
energy, motion, force, ...
Organic
alkane, alcohol, benzene, ...
Inorganic
metal, oxide, lattice, ...
Quantum
entanglement, particle, spin,...

Example: Document 1



c_1 = [Science, Physics, Quantum]

c_1 = [0.5,0.3,0.2]

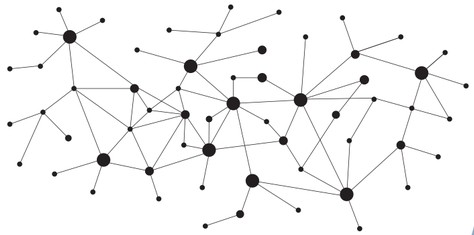
Posterior Inferece

- Closed form of posterior distribution is not available

- Collapsed Gibbs sampling
  - Some of the latent variables are integrated out
  - assess 2 values only:
    - the hierarchical document-topic distribution, $\underline{c}$, for each document
    - the topic-word distribution, $\underline{z}$, for each topic

Data

- Coauthor-Citation data from the Annal of Applied Statistics
- Total number of Documents = 3248
- Total number of Authors = 3607
- Coauthorship and citation information is available



Network building

- Node = Document

- Edges represent links between the documents
  - Direct citation between papers
  - Link to all other papers by the current paper’s authors

- Size of network:
  - Number of nodes = 3248
  - Number of edges = 21356

Results

Run SBM to obtain communities

1

Run hLDA to obtain topic distributions

2

Integration to extract deeper insights

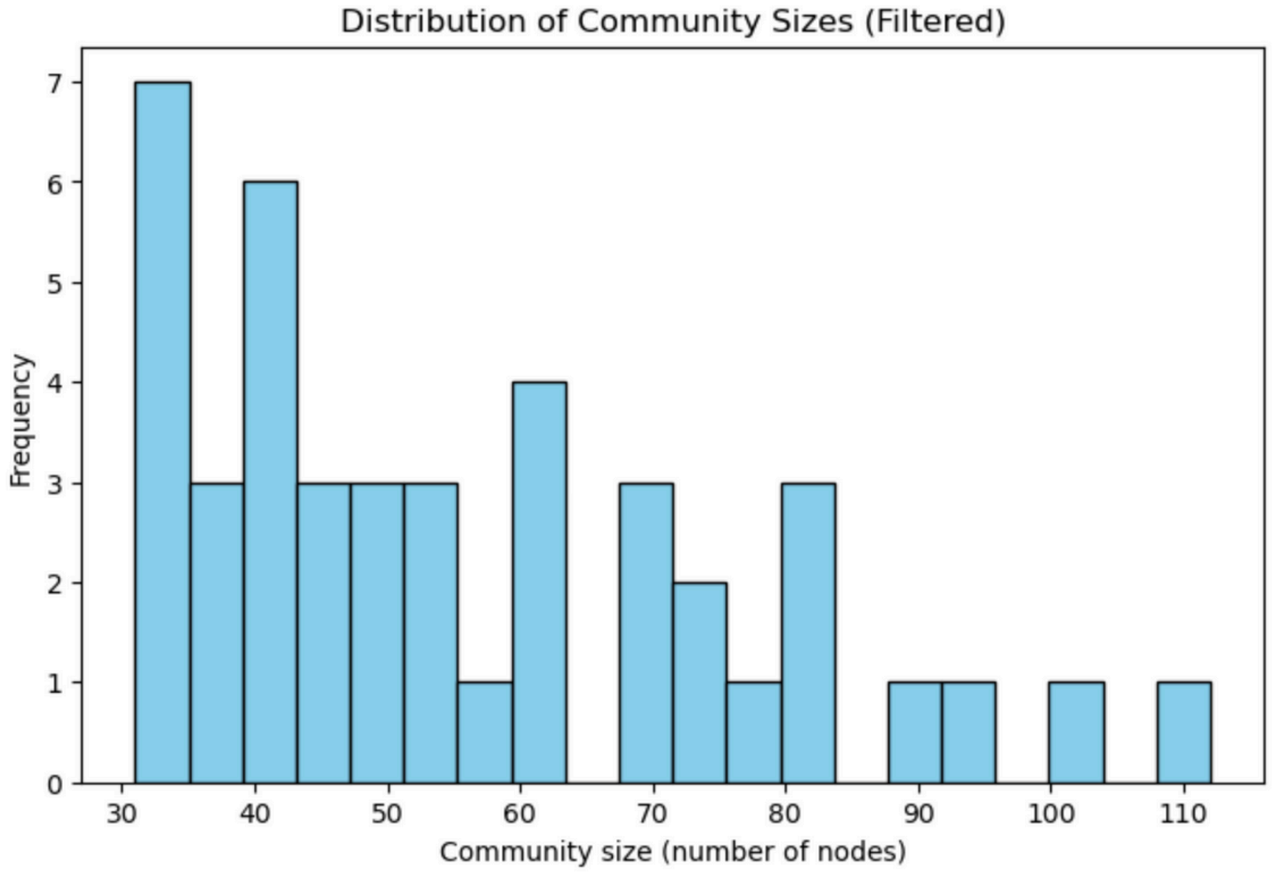
3

Results: SBM

- Applied to the largest Connecting Componet in the network
- Removed communities with less than 30 numbers

Results: SBM

- Number of communities: 43
- Min community size: 31
- Max community size: 112
- Average community size: 56



Results: hLDA

- Applied to the entire corpus (not community-specific) to learn a global topic hierarchy.
- Total number of topics = 669
- Topic count per level:
  - Level 1: 1
  - Level 2: 28
  - Level 3: 113
  - Level 4: 527

Results: Integration

Better interpretation of the communities detected

Discovering semantic similarities among communities

Results: Understanding Community

- Constructing The community hierarchical topic distribution

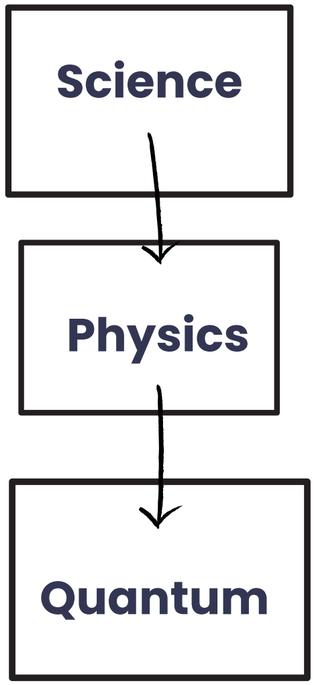
- The community topic distribution is obtained by:
  - flattening each document's topic distribution
  - adding them together
  - normalising
  - reconstructing the tree.

Results: Understanding Community

v = [topic0(L0), topic2(L1), ..., topic668(L2)]

v = [0,0,...,0]

Results: Understanding Community



Flatten

c_1 = [Science(L0), Physics(L1), Quantum(L2)]

c_1 = [0.5,0.3,0.2]

Results: Understanding Community

v = [topic0(L0), topic2(L1), ..., topic668(L2)]

c_1 = [Science(L0), Physics(L1), Quantum(L2)]

Results: Understanding Community

v = [topic0(L0), ..., topic10(L1), ..., topic100(L2), ... topic668(L2)]

c_1 = [topic0(L0), topic10(L1), topic100(L2)]

Results: Understanding Community

v = [0(topic0), ..., 0(topic10), ..., 0(topic100),... 0(topic668)]

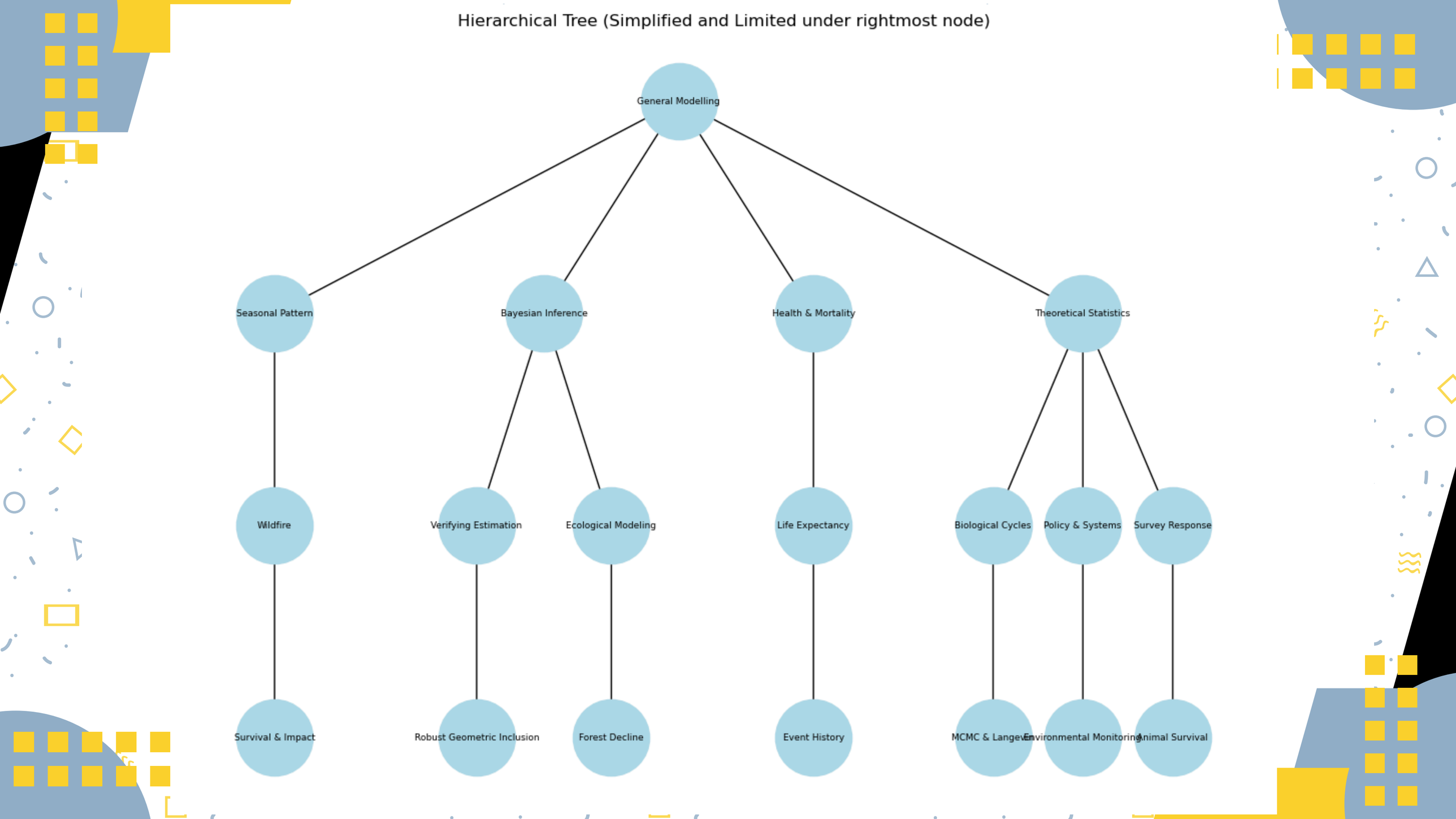
c_1 = [0.5(topic0), 0.3(topic10), 0.2(topic100)]

Results: Understanding Community

v = [0.5(topic0), ..., 0.3(topic10), ..., 0.2(topic100),... 0(topic668)]

Results: Understanding Community

- Normalise the vector *v*
- Remove topics with zero probability at 3 d.p.
- Reconstruct the hierarchical tree using the final topics


Hierarchical Tree (Simplified and Limited under rightmost node)
General Modelling
Seasonal Pattern
Bayesian Inference
Health & Mortality
Theoretical Statistics
Wildfire
Verifying Estimation
Ecological Modeling
Life Expectancy
Biological Cycles
Policy & Systems
Survey Response
Survival & Impact
Robust Geometric Inclusion
Forest Decline
Event History
MCMC & Langevin
Environmental Monitoring
Animal Survival

Results

- Previous result is helpful in understanding the individual communities
- However, due to the large number of distinct topics, it’s difficult to compare communities sementically
- We want to assess how similar the communities are semantically.

Results: Semantic Comparison

- Idea:
  - Construct a new Topic-Topic network where
    - Nodes: represent topics
      - Only the leaf nodes
    - Edges: represent similarity between topics
      - Defined by cosine similarity between topic-word distributions

Results: Semantic Comparison

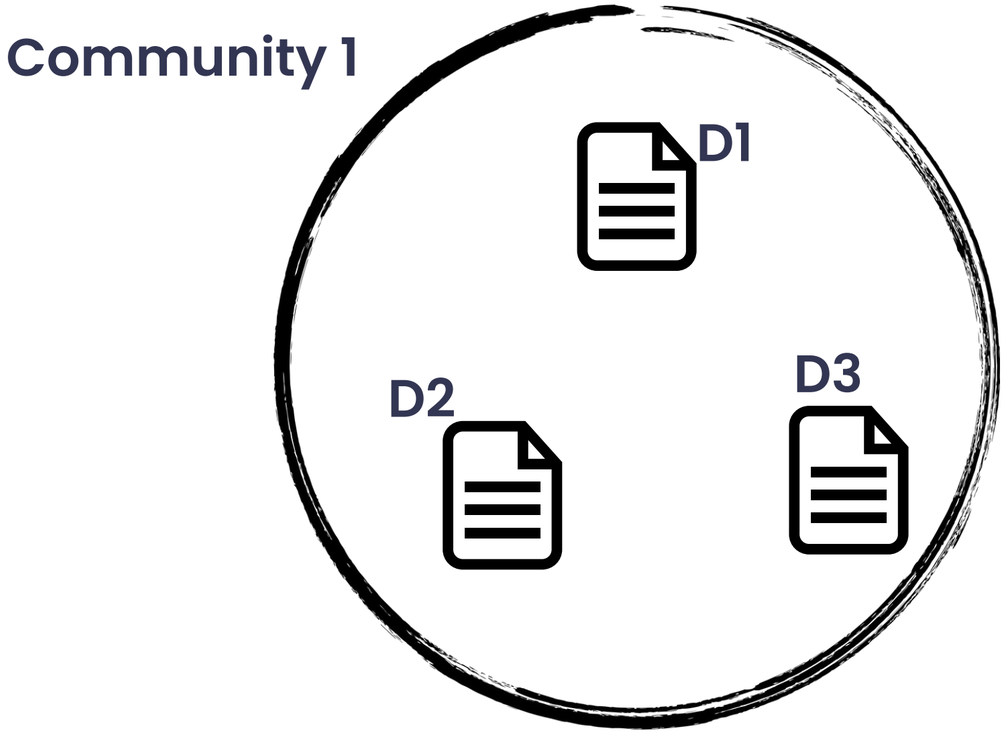
- Topic-Topic network has:
  - 472 nodes
  - 2137 edges

- Running SBM on this network yields 21 communities

Results: Semantic Comparison

- Map the topic community back to communities found in the document network

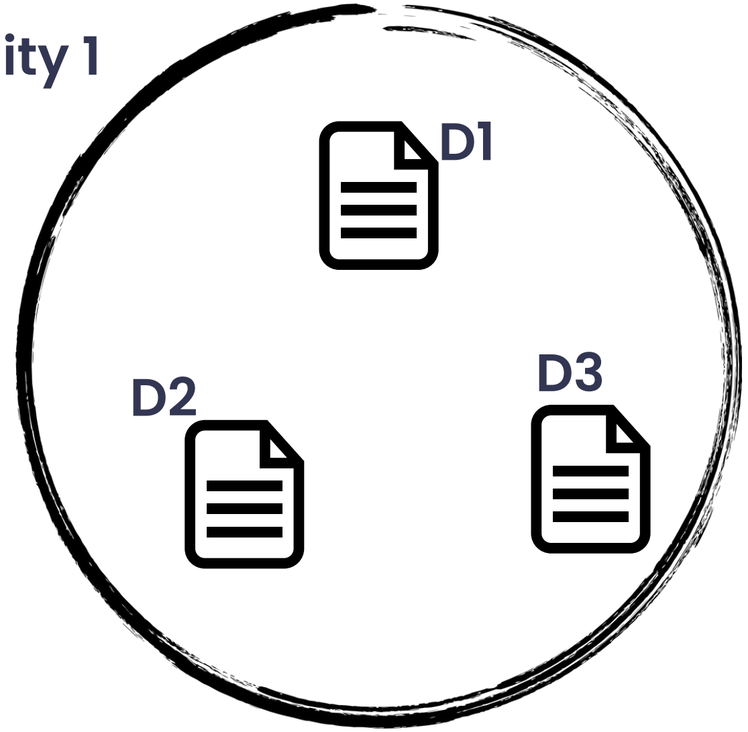
Results: Semantic Comparison



Results: Semantic Comparison

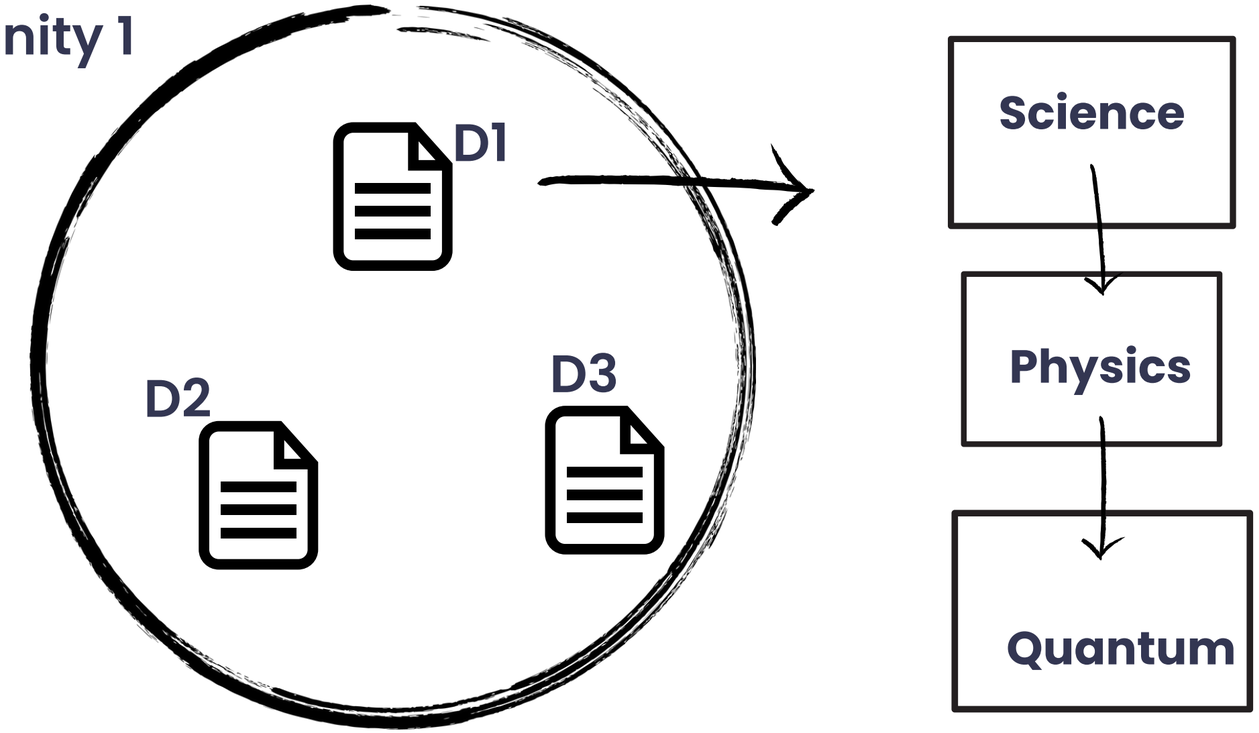
v_1 = [com1, com2, ..., com21]
v_1 = [0,0, ...,0]

Community 1



Results: Semantic Comparison

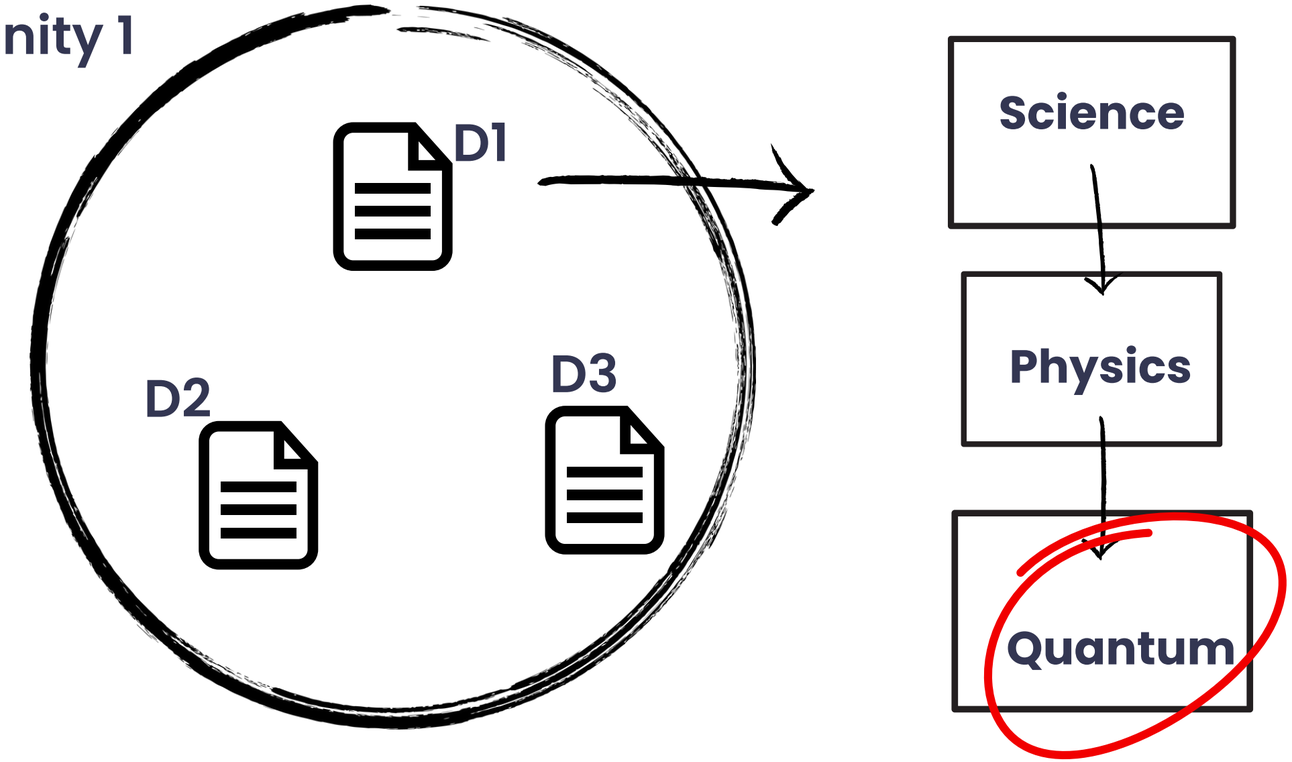
v_1 = [com1, com2, ..., com21]
v_1 = [0,0, ...,0]



Results: Semantic Comparison

v_1 = [com1, com2, ..., com21]
v_1 = [0,0, ...,0]

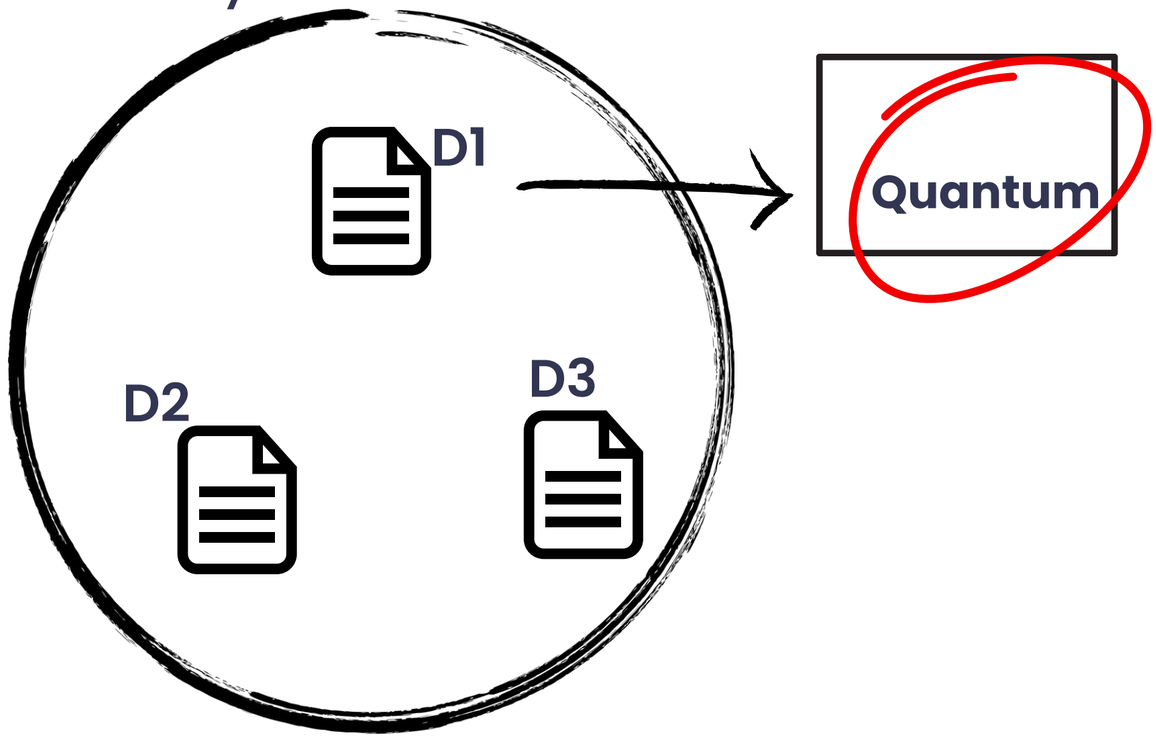
Community 1



Results: Semantic Comparison

v_1 = [com1, com2, ..., com21]
v_1 = [0,0, ...,0]

Community 1



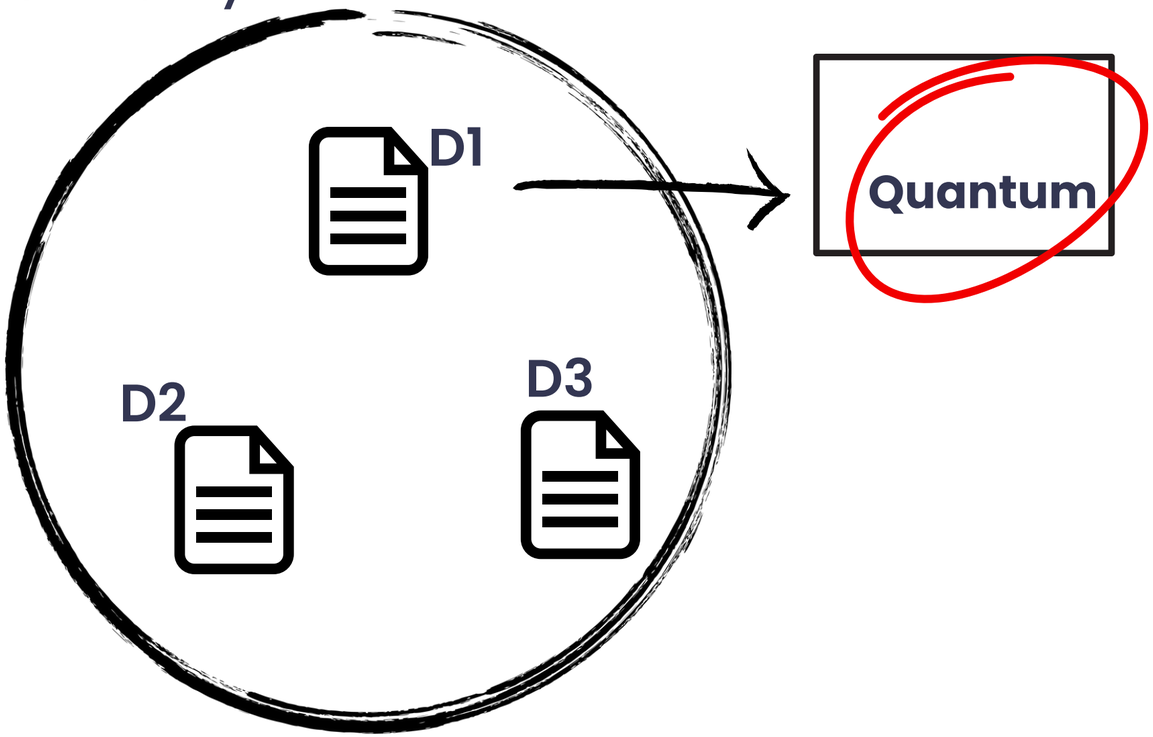
Topic community = 21

Results: Semantic Comparison

v_1 = [com1, com2, ..., com21]

v_1 = [0,0, ...,1]

Community 1



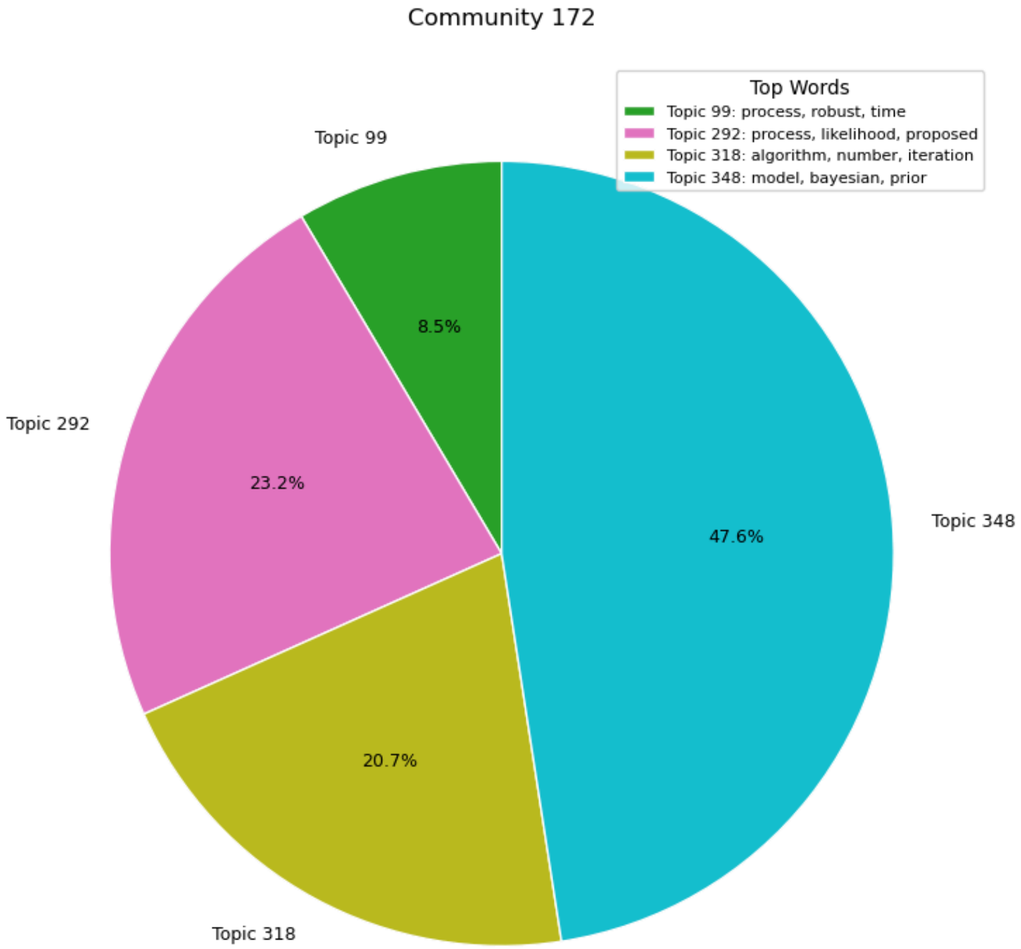
Topic community = 21

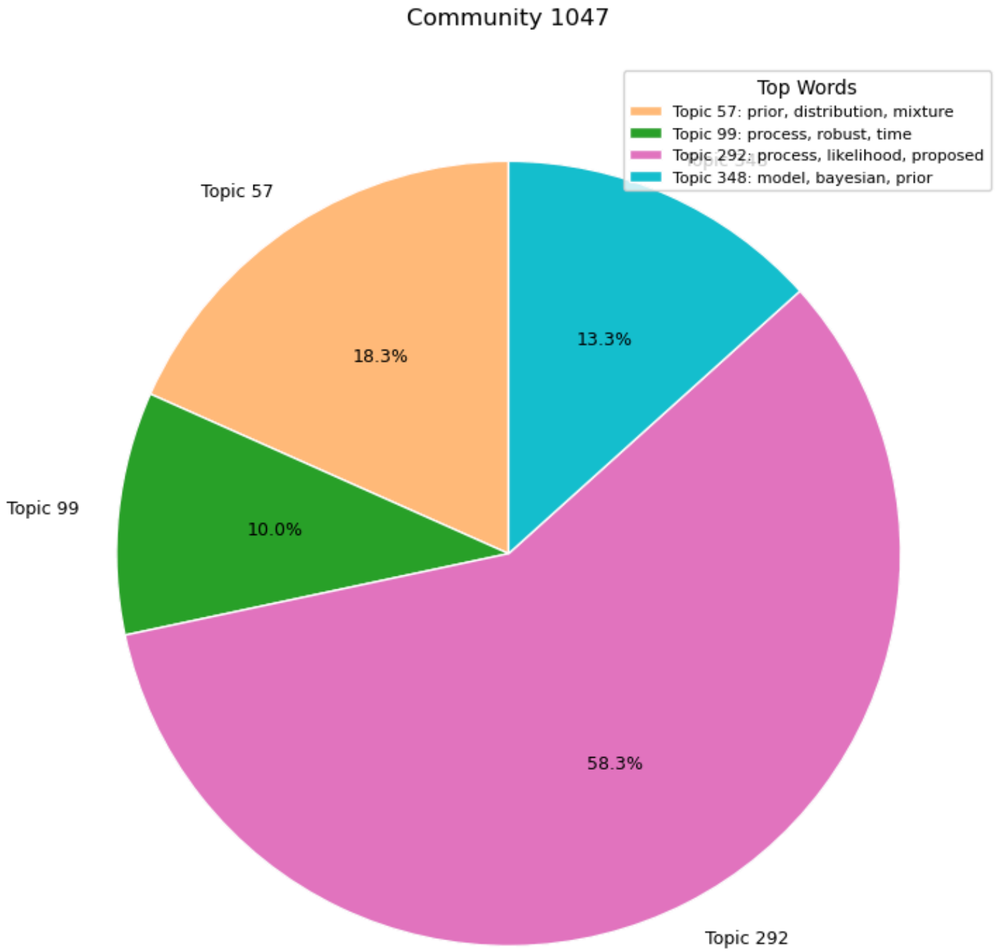
v_1[com21] += 1

Results: Semantic Comparison

- Normalise, taking the size of community into consideration
- Obtain a topic distribution of fixed length k(21)
- Allows for easier comparison across document communitie

Document Community Topic Distributions with Consistent Topic Colors





CONCLUSION

- When textual information is present, topic modelling is very helpful in understanding the result of community detection
- Can be applied to other networks such as social network(twitter, etc)
- Better recommendation/advertisement can be done

THANK YOU

FOR YOUR ATTENTION

REFERENCE

https://medium.com/data-science/community-detection-algorithms-9bd8951e7dae
(Community detection figure)