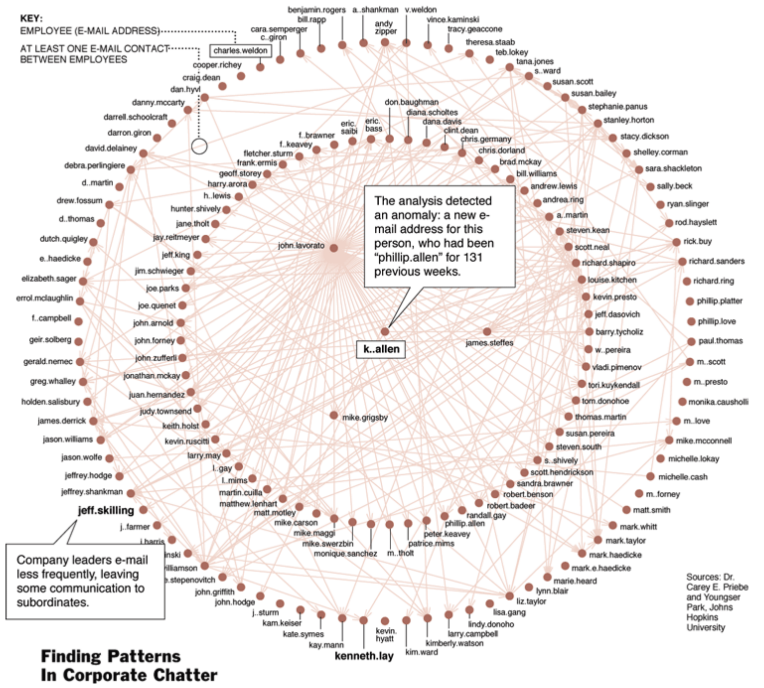
CS630 Graduate Algorithms
December 5, 2024
Dora Erdos and Jeffrey Considine

Graph randomization

Slides by Dora Erdos

Real life graph data

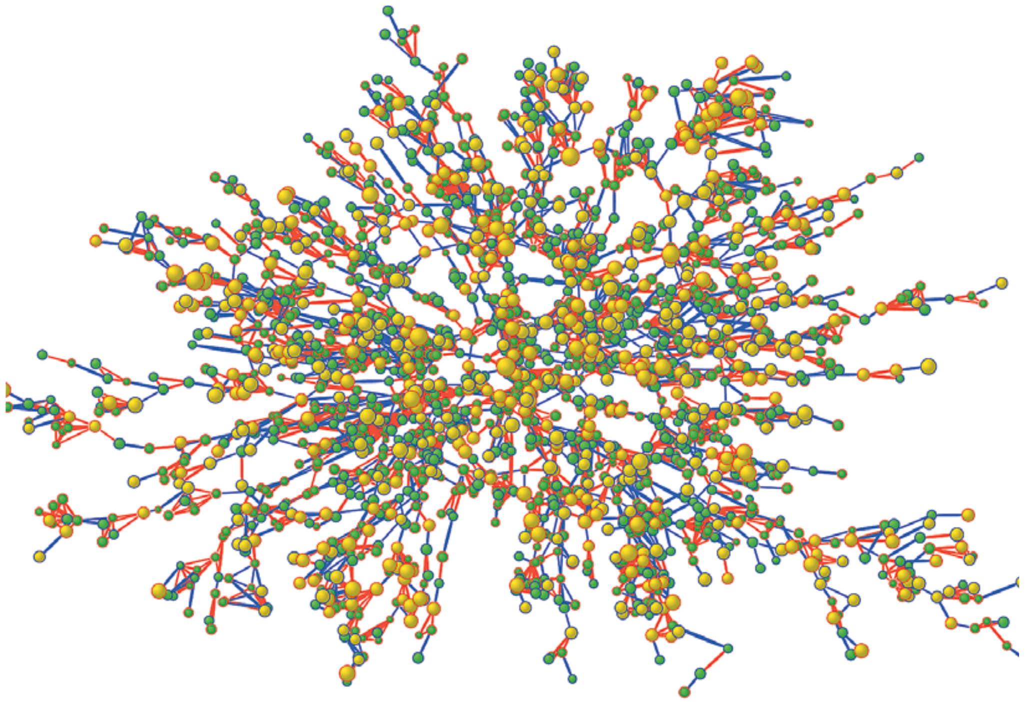


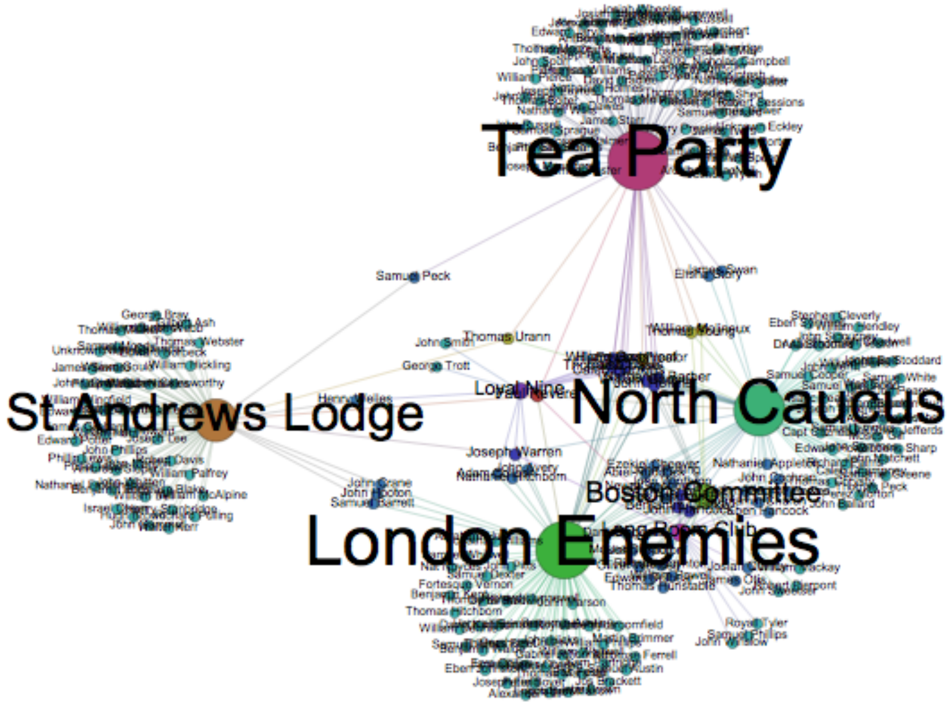
The analysis detected an anomaly: a new e-mail address for this person, who had been "phillip.allen" for 131 previous weeks.

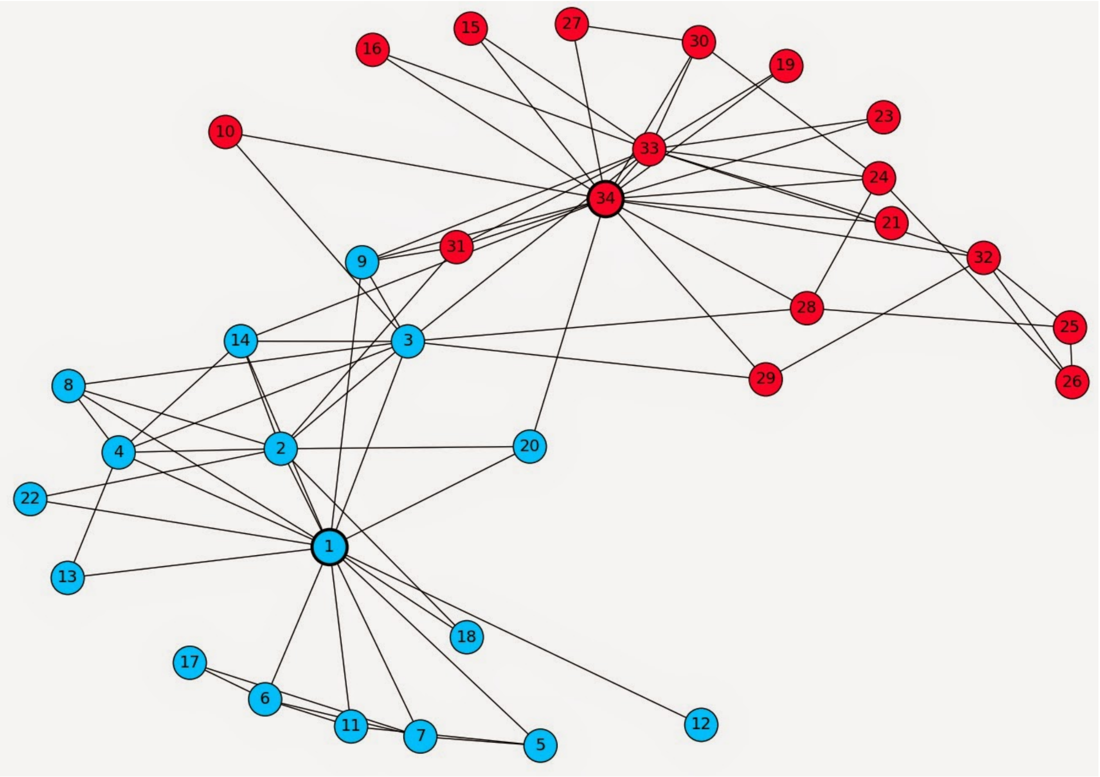
Company leaders e-mail less frequently, leaving some communication to subordinates.

Sources: Dr. Carey E. Priebe and Youngser Park, Johns Hopkins University

Finding Patterns In Corporate Chatter







Pattern or not?

These are adjacency matrices of graphs. What can you observe about them?

$$\begin{array}{|cccccccc|}\hline
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ \hline
\end{array}
\qquad
\begin{array}{|cccccccc|}\hline
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline
\end{array}
\qquad
\begin{array}{|cccccccc|}\hline
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline
\end{array}$$

Measuring networks

- Degree distributions
- Small world phenomena
- Clustering coefficient
- Mixing patterns
- Degree correlations
- Communities and clusters

Why?

1. Find patterns in data
2. Generate graphs that are similar to real world ones

Today:

We can measure the above properties for real-life data.

Generate random graphs that maintain these properties.

⇒ Use it to verify whether an observed pattern is "real" or just coincidence.

graph randomization

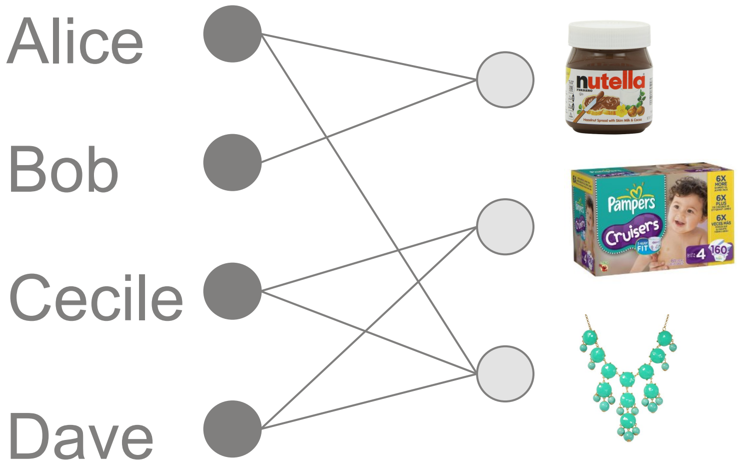
- motivation: study patterns and properties in graph - degree distribution , communities, virus propagation.
  - How significant is the result in the observed graph?

- goal: need some kind of hypothesis testing on graphs.
  - For this, generate random graph with similar properties

- similar = (almost) same value of graph statistics

- null hypothesis to examine: do the observed data mining results on the original graph convey any information in addition to the specified graph statistics?

Co-purchase data



AA^T

	Alice	Bob	Cecile	Dave
Alice	2	1	1	1
Bob	1	1	0	0
Cecile	1	0	2	2
Dave	1	0	2	2

$\text{Alice} \cap \text{Alice} = 2$

$\text{Alice} \cap \text{Bob} = 1$

$\text{Alice} \cap \text{Cecile} = 1$

$\text{Alice} \cap \text{Dave} = 1$

$\text{Bob} \cap \text{Bob} = 1$

$\text{Bob} \cap \text{Cecile} = 0$

$\vdots$

A

Alice	1	0	1
Bob	1	0	0
Cecile	0	1	1
Dave	0	1	1

$A^T A$

	2	0	1
	0	2	2
	1	2	3

$\cap$ $= 2$

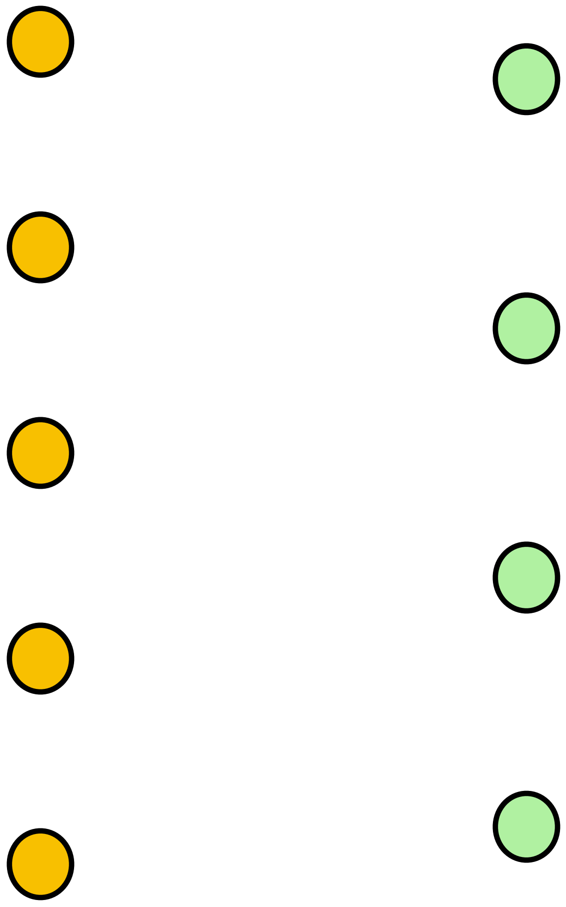
$\cap$ $= 0$

$\cap$ $= 1$

$\vdots$

Bipartite Graph generation problem

Find a bipartite graph with node degrees
[3, 3, 2, 2, 1] and [5, 3, 2, 1]



Bipartite Graph generation problem

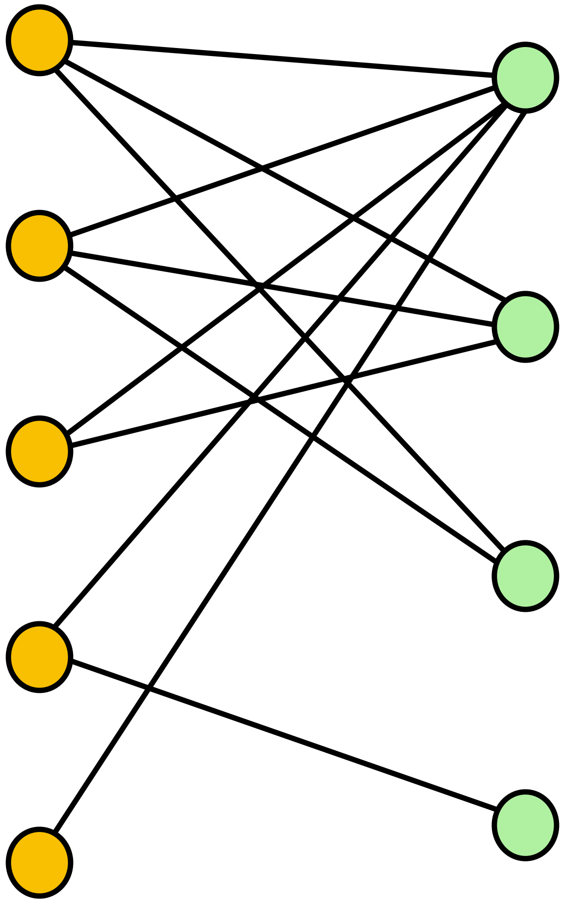
Find a bipartite graph with node degrees
[3, 3, 2, 2, 1] and [5, 3, 2, 1]

Such a graph exists iff

$$\sum_{i=1}^{5} d_i = \sum_{j=1}^{4} q_j$$

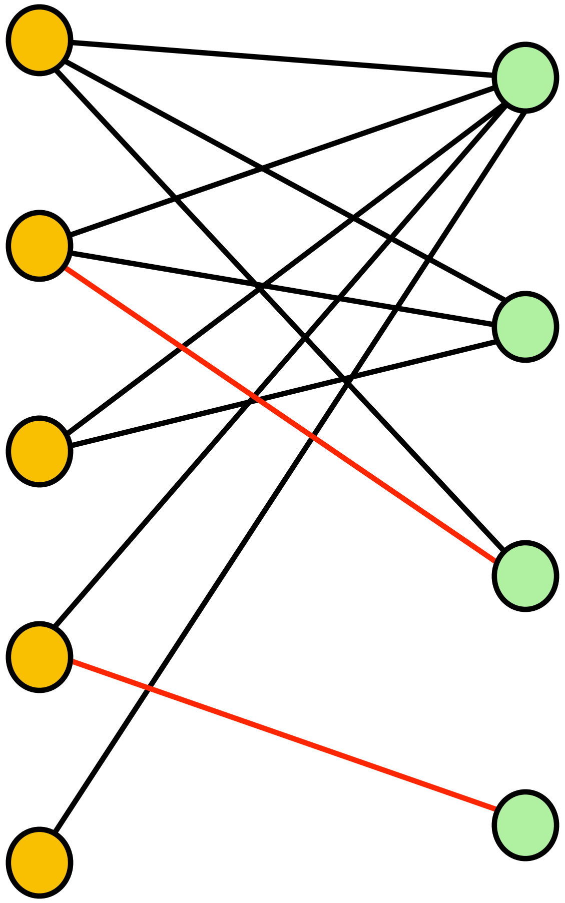
and

$$\sum_{i=1}^{k} d_i \leq \sum_{j=1}^{4} \min\{q_j, k\}$$



Bipartite Graph generation problem

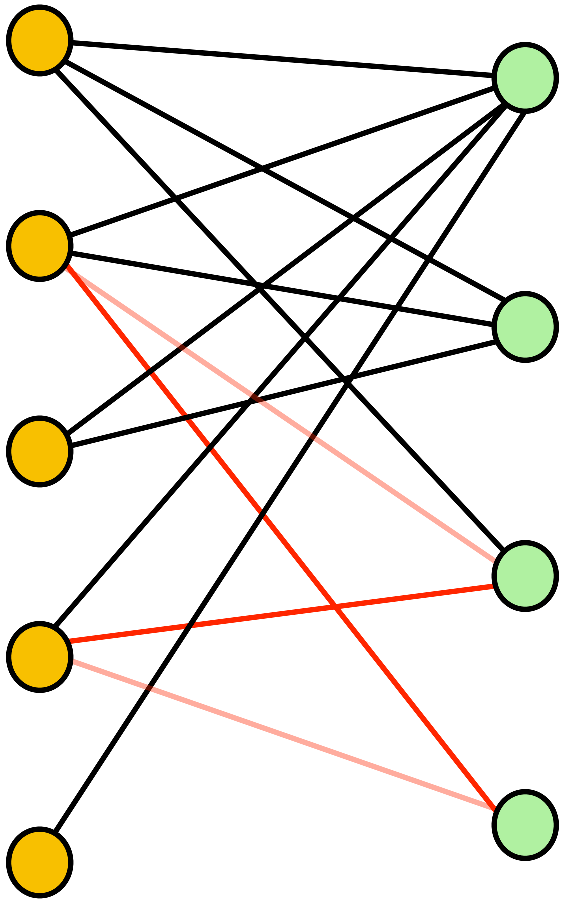
There are other graphs with the same degree sequence.



Bipartite Graph generation problem

There are other graphs with the same degree sequence.

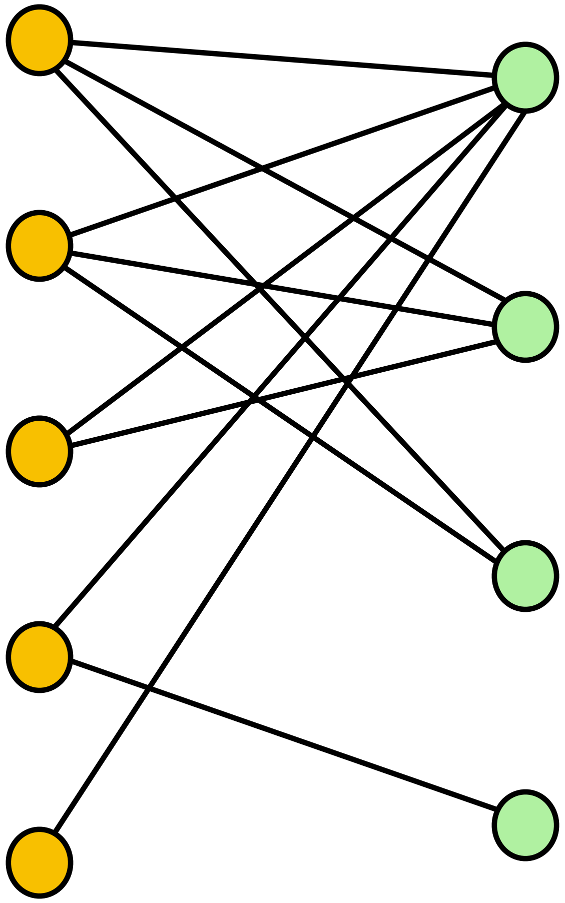
How to find them?
How can we sample one of the graphs with this degree sequence at random?



Bipartite Graph generation problem

Equivalent problems:

- pick a graph uniform at random with the given degree sequence.
- generate a binary matrix uniform at random with given row and column marginals



bipartite adjacency matrix

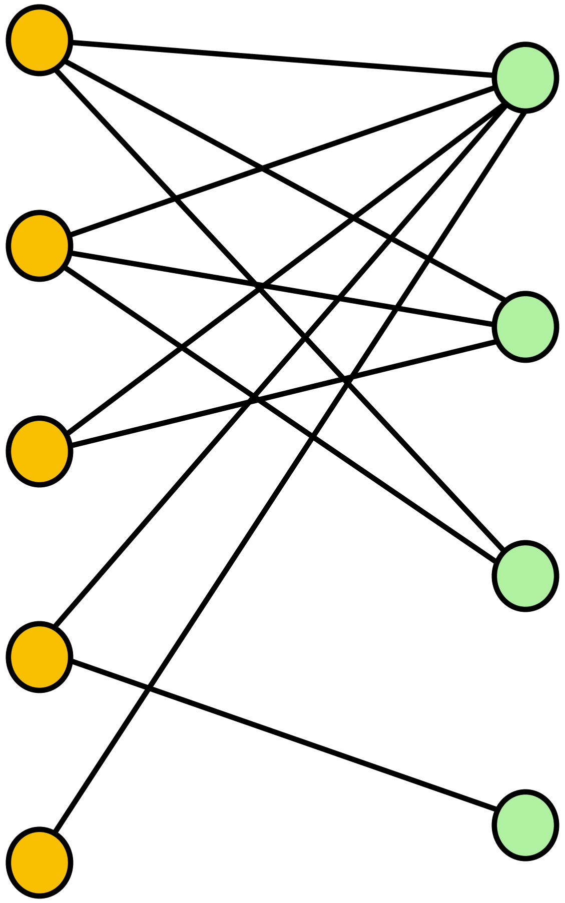
- rows: left class of nodes
- columns: right class
- 1 corresponds to an edge

$$\begin{array}{|cccc|}
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0
\end{array}$$

Bipartite Graph generation problem

Equivalent problems:

- pick a graph uniform at random with the given degree sequence.
- generate a binary matrix uniform at random with given row and column marginals



Row and column sums correspond to the node degrees.

- called marginals of the matrix

$$\begin{array}{cccc|c} 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 5 & 3 & 2 & 1 & \end{array}$$

Why maintain matrix marginals?

Ecology:

- rows correspond to species
- columns to locations
- 0/1 indicates presence or absence

- row sum - commonness of species
- column sum - diversity of location

Text:

- rows correspond to words
- columns to documents
- 0/1 indicates presence or absence

- row sum - shared words across docs
- column sum - column of each doc

Why do randomization?

Market basket:

- rows correspond to merchandise
- columns to costumers
- 0/1 indicates buying

- row sum - how many customers buy that item
- column sum - how much did a customer buy
- laundry detergent and fabric softener often appear in the same basket
- is there correlation?

- bread and milk often occur together
- is there a correlation?

Why do randomization?

General:

- We observe some pattern in the data
    - example: we find a subgraph with density d (this corresponds to a cluster in the adjacency matrix!)
- is this surprising or is it a random coincidence?
    - It is known that Erdos-Renyi random graphs contain a subgraph with density d with probability p — not surprising!
    - finding a subgraph that is a click, e.g. density = 1 — very surprising!

Why do randomization?

Clustering:

- K-means clustering on 1s in binary matrix
- find clusters
- shuffle (= sample a matrix with the same marginals at random) and find clusters again
- if we don’t find anything, that implies that the original clusters are statistically significant

$$\boxed{\begin{matrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{matrix}} \qquad \boxed{\begin{matrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{matrix}}$$

Results on swap randomization

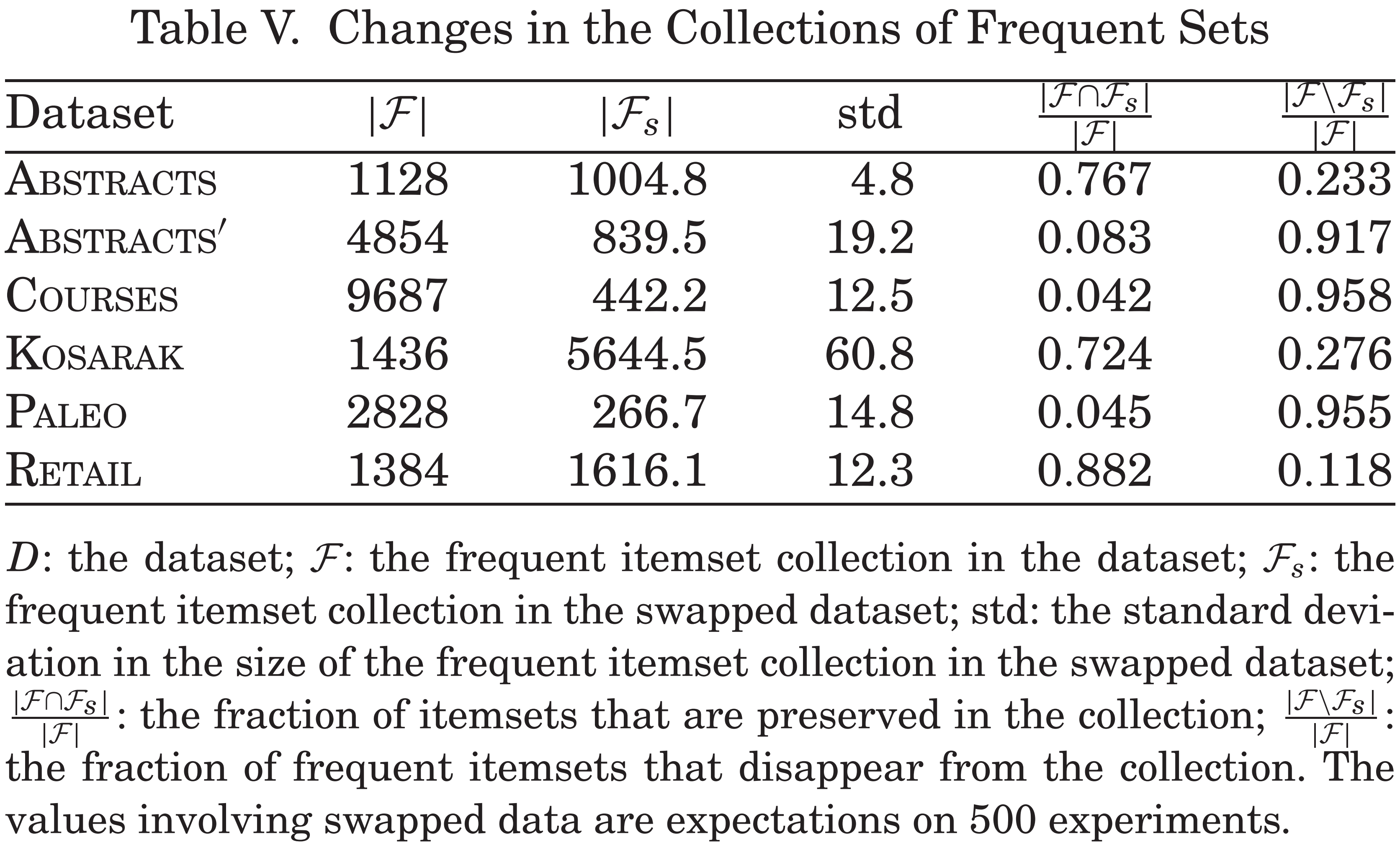


Table V. Changes in the Collections of Frequent Sets

Dataset	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{F}_s\rvert$	std	$\frac{\lvert\mathcal{F}\cap\mathcal{F}_s\rvert}{\lvert\mathcal{F}\rvert}$	$\frac{\lvert\mathcal{F}\setminus\mathcal{F}_s\rvert}{\lvert\mathcal{F}\rvert}$
ABSTRACTS	1128	1004.8	4.8	0.767	0.233
ABSTRACTS′	4854	839.5	19.2	0.083	0.917
COURSES	9687	442.2	12.5	0.042	0.958
KOSARAK	1436	5644.5	60.8	0.724	0.276
PALEO	2828	266.7	14.8	0.045	0.955
RETAIL	1384	1616.1	12.3	0.882	0.118

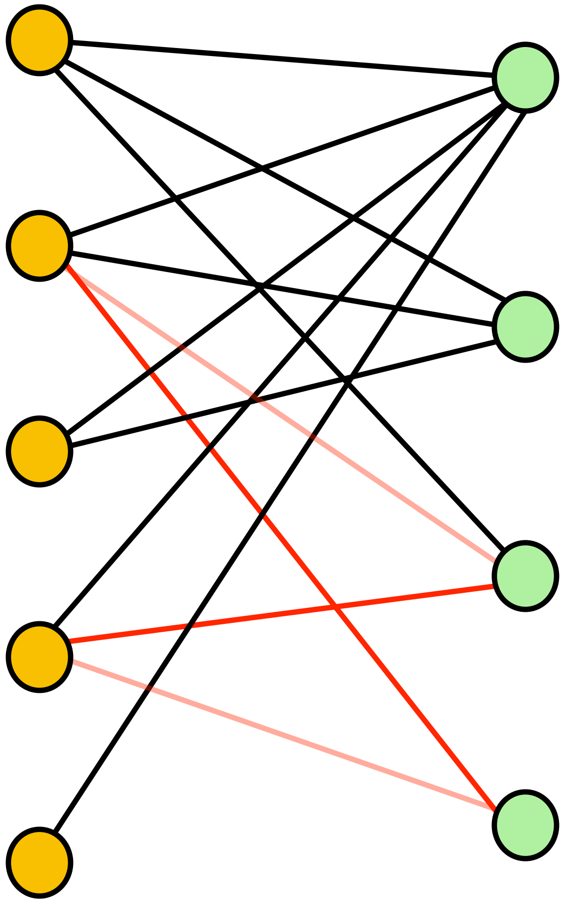
D: the dataset; $\mathcal{F}$: the frequent itemset collection in the dataset; $\mathcal{F}_s$: the frequent itemset collection in the swapped dataset; std: the standard deviation in the size of the frequent itemset collection in the swapped dataset; $\frac{\lvert\mathcal{F}\cap\mathcal{F}_s\rvert}{\lvert\mathcal{F}\rvert}$: the fraction of itemsets that are preserved in the collection; $\frac{\lvert\mathcal{F}\setminus\mathcal{F}_s\rvert}{\lvert\mathcal{F}\rvert}$: the fraction of frequent itemsets that disappear from the collection. The values involving swapped data are expectations on 500 experiments.

data from Gionis et al.

Swap randomization process - bipartite

bipartite adjacency matrix

$$\begin{array}{|cccc|c} 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & \boxed{1} & \boxed{0} & 3 \\ 1 & 1 & 0 & 0 & 2 \\ 1 & 0 & \boxed{0} & \boxed{1} & 2 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 5 & 3 & 2 & 1 & \end{array}$$

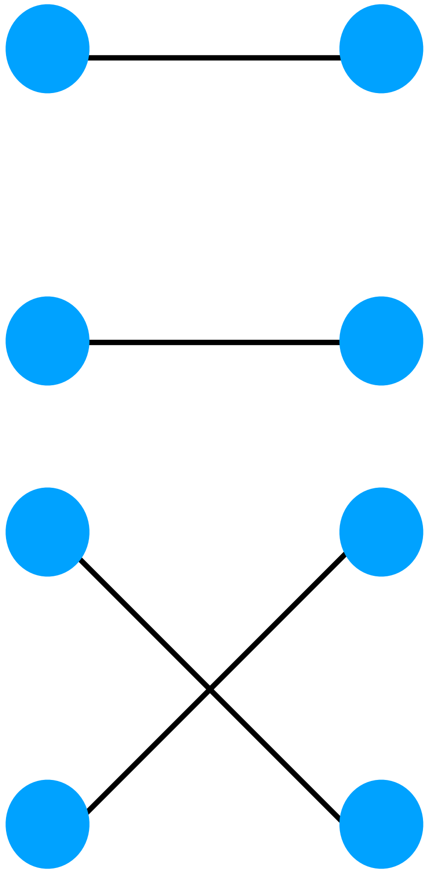


Local swap: find 2x2 submatrix with 1s and 0s in the two diagonals and swap the 0s and 1s.

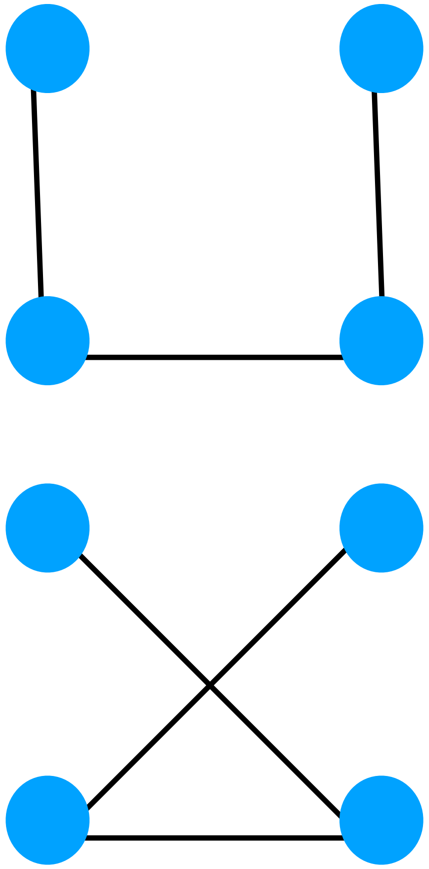
Swap randomization process - general graph

1. LocalSwap - as before
    1. maintains degree distribution
2. NeighborSwap: switch neighbors
    1. preserves the clustering coefficient
3. Flip - flip neighbor of rand node an edge, but only if degree diff of neighbors is 1
    1. if there is a gap in the sequence of degrees, then no flip occurs between two parts

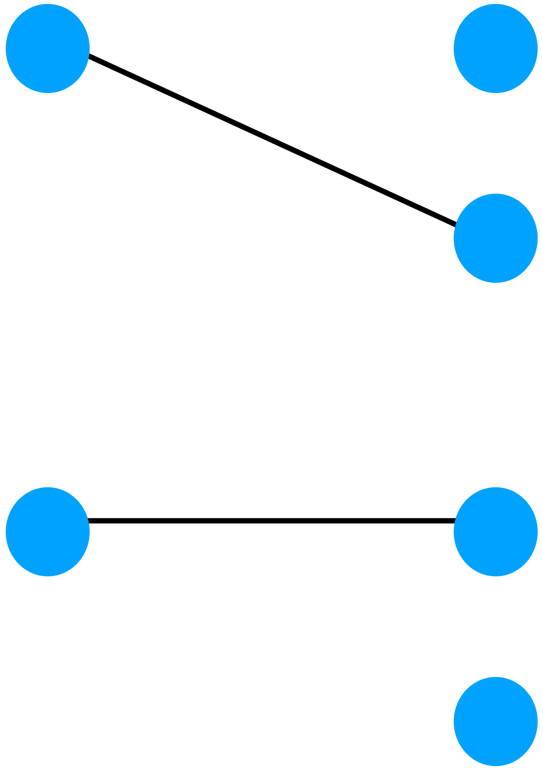
1.



2.



3.



Markov chain on swap-space

- How to sample a graph (bipartite or general) with given degrees uniform at random?
- How to sample a matrix with given marginals uniform at random?

Markov chain

Markov chain: a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event

- set of states S = {S_1,S_2,…,S_n}

- nxn transition matrix P
  - P_{ij} probability of moving to state j when at state i
  - P is a stochastic matrix, i.e. its rows sum to 1 $\sum_{j=1}^{n} P_{ij} = 1$

- memoryless: the next state in the process only depends on the current state and not the past states.
- this property is sometimes called Markov property.

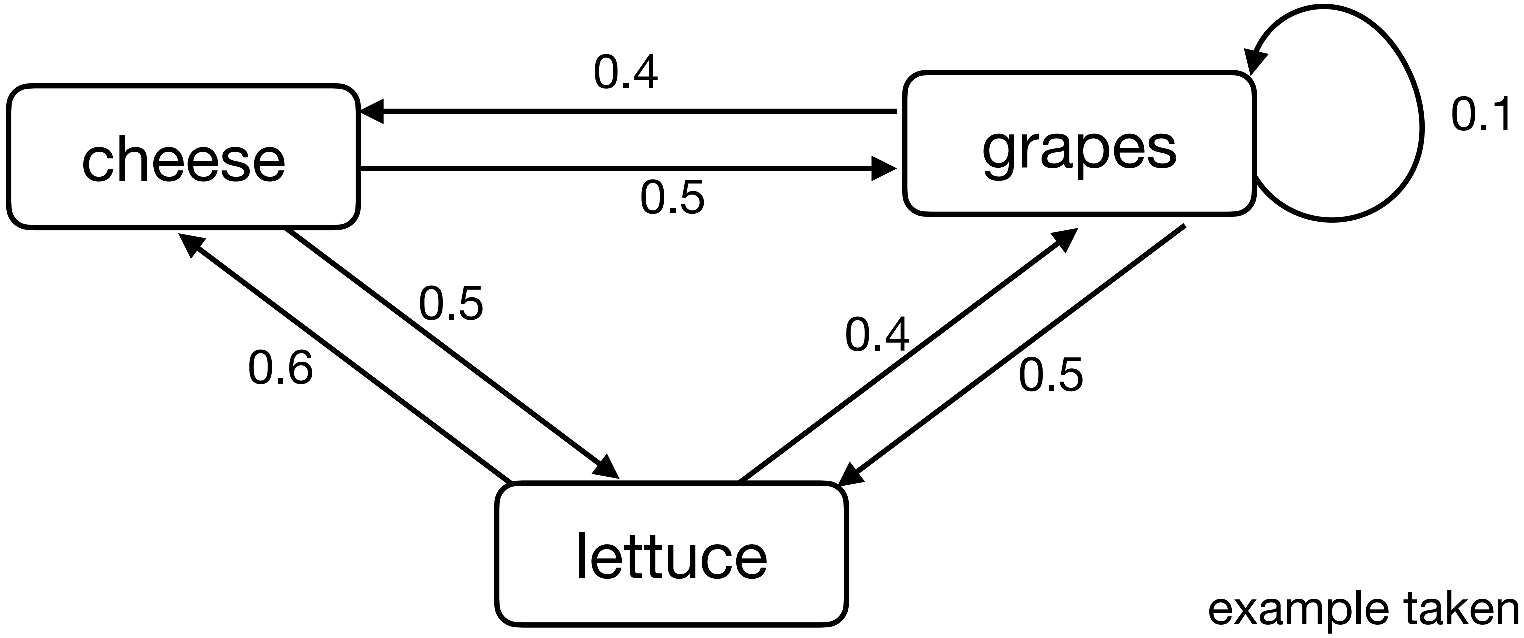
(this is called a first order Markov chain. If each transition would depend on the previous k, then it would be called kth order).

Markov chain examples

Some animal only eats grapes, lettuce and cheese based on the following rules:

- eats once a day
- cheese today ⇒ tomorrow grapes or lettuce with equal probability
- grapes today ⇒ tomorrow grapes with probability 0.1, cheese with 0.4 or lettuce with 0.5
- lettuce today ⇒tomorrow grapes with probability 0.4 and cheese with 0.6.

states= the three types of food
transition probability= the probability of eating a certain food tomorrow, given what it ate today.
memoryless: tomorrow's choice only depends on what it ate today



example taken from Wikipedia

Markov chain examples

Consider sequences of n numbers. Two sequences are considered to be “neighbors” if we get one from the other by swapping two numbers.

states = all possible sequences (orderings) of the n numbers (how many states are there?)

transition probability between two states/sequences i and j:

- 0 if i and j are not neighbors
- ???? if i and j are neighbors

Markov Chain Monte Carlo (MCMC) sampling

One application of Markov chains is sampling.

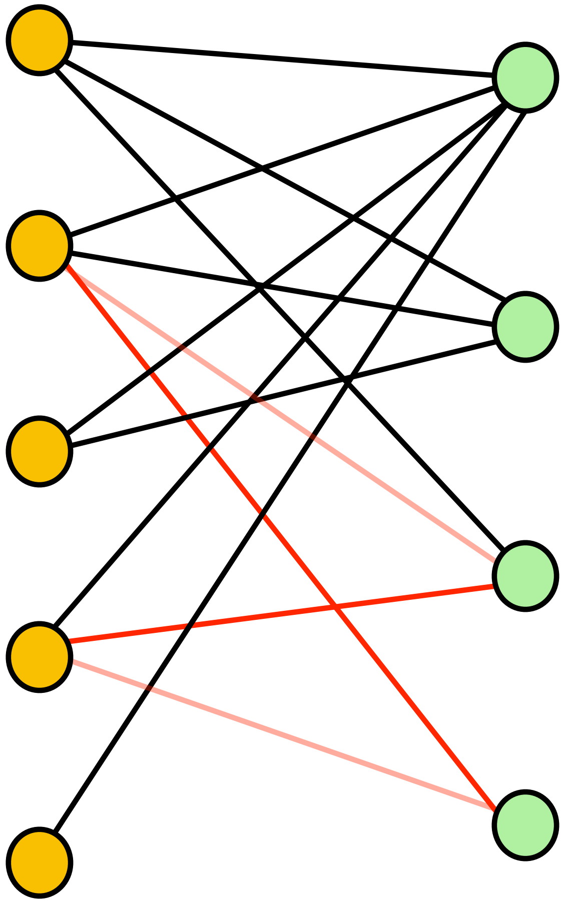
- States correspond to the instances in the sample space.
- Transition probabilities depend on the application.

Intuition: We want to sample a random instance. We do it by performing a random walk on the Markov chain. After we (approximately) reach the stationary distribution the state the walker is in is random among the states. And the corresponding sample instance is a random sample among the instances.

Swap randomization process - bipartite

bipartite adjacency matrix

$$\begin{array}{|cccc|c}
\hline
1 & 1 & 1 & 0 & 3\\
1 & 1 & \boxed{1} & \boxed{0} & 3\\
1 & 1 & 0 & 0 & 2\\
1 & 0 & \boxed{0} & \boxed{1} & 2\\
1 & 0 & 0 & 0 & 1\\
\hline
5 & 3 & 2 & 1 &
\end{array}$$



Local swap: find 2x2 submatrix with 1s and 0s in the two diagonals and swap the 0s and 1s.

Markov chain on swap-space

Perform a random walk on the space of possible matrices!

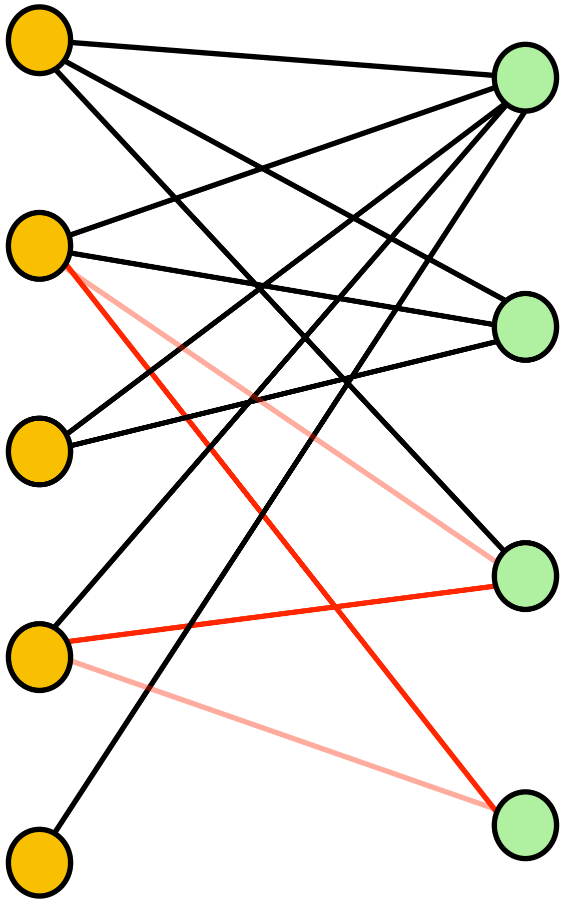
- Each state corresponds to a matrix with the given marginals.
- There is an edge between them if you can get from one to the other by a local swap
- perform a random walk on this space for sufficient many steps.
    - Start the walk from the known matrix instance, e.g. $q^{(0)}=[1,0,0,\ldots,0]$
- After "many" steps on the walk (i.e. manys waps) the matrix corresponding to the state we are in is a uniform random matrix with the given marginals.

Swap randomization process - bipartite

How should we implement this?

bipartite adjacency matrix

$$\begin{array}{|cccc|c} \hline 1 & 1 & 1 & 0 & 3 \\ 1 & 1 & \boxed{1} & \boxed{0} & 3 \\ 1 & 1 & 0 & 0 & 2 \\ 1 & 0 & \boxed{0} & \boxed{1} & 2 \\ 1 & 0 & 0 & 0 & 1 \\ \hline 5 & 3 & 2 & 1 & \end{array}$$



Local swap: find 2x2 submatrix with 1s and 0s in the two diagonals and swap the 0s and 1s.

Swap randomization process

- How do we find the Markov chain? How to generate all possible neighbors of a state?
- More specifically, how do we choose the next random step, i.e. find a swap?

  approach 1: pick 4 nodes at random, until you find 4 that you can swap

  approach 2: first create the Markov chain, then start walking

Swap randomization process

Approach 1:

algorithm: pick 4 nodes at random until we find a tuple we can swap

worst case $\binom{n}{4}$ iterations for one swap

This Markov Chain is connected $\Rightarrow$ we can get from any matrix to any other matrix by a sequence of local swaps

Swap randomization process

Algorithm generates a neighboring graph uniform at random
Markov chain is connected

Uniform random sample: each state should have the same probability

- add self-loops so that all nodes/states have the same degree
  - inefficient
- Metropolis-Hastings: if state G’ has higher degree than state G, then step from G’ to G with probability $\min\{1, deg(G)/deg(G')\}$
  - more efficient, fast convergence

Swap randomization process

Thm. [Gionis. et al] Running time analysis: for bounded degree graphs find_adjacent takes constant expected number of iterations

Metropolis Hastings:
given d(G) and neighbor G’, d(G’) can be computed in time min{n,m}

Reconstruction from sparse co-occurrence data

Open information extraction

Elizabeth Windsor is the ruler of England.

Elizabeth is the queen of the United Kingdom.

Elizabeth II. is the queen of England.

Beckham plays for LA Galaxy.

Elizabeth is the senator of Massachusetts.

D. Beckham plays soccer for England.

Elizabeth Warren lives in D.C.

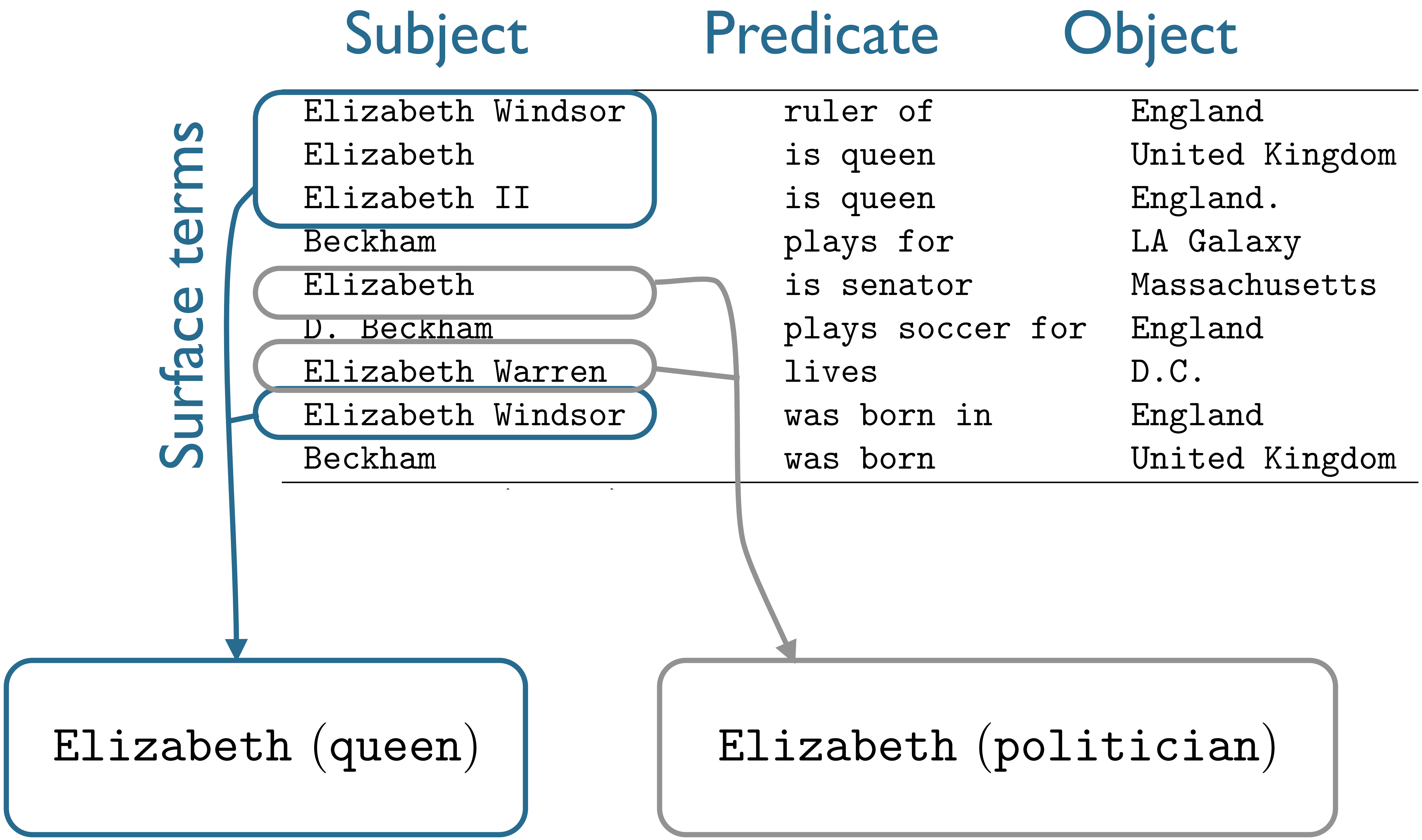
Elizabeth Windsor was born in England.

Beckham was born in the United Kingdom.

Open information extraction

	Subject	Predicate	Object
Surface terms	Elizabeth Windsor	ruler of	England
	Elizabeth	is queen	United Kingdom
	Elizabeth II	is queen	England.
	Beckham	plays for	LA Galaxy
	Elizabeth	is senator	Massachusetts
	D. Beckham	plays soccer for	England
	Elizabeth Warren	lives	D.C.
	Elizabeth Windsor	was born in	England
	Beckham	was born	United Kingdom

Open information extraction



Subject	Predicate	Object
Elizabeth Windsor	ruler of	England
Elizabeth	is queen	United Kingdom
Elizabeth II	is queen	England.
Beckham	plays for	LA Galaxy
Elizabeth	is senator	Massachusetts
D. Beckham	plays soccer for	England
Elizabeth Warren	lives	D.C.
Elizabeth Windsor	was born in	England
Beckham	was born	United Kingdom

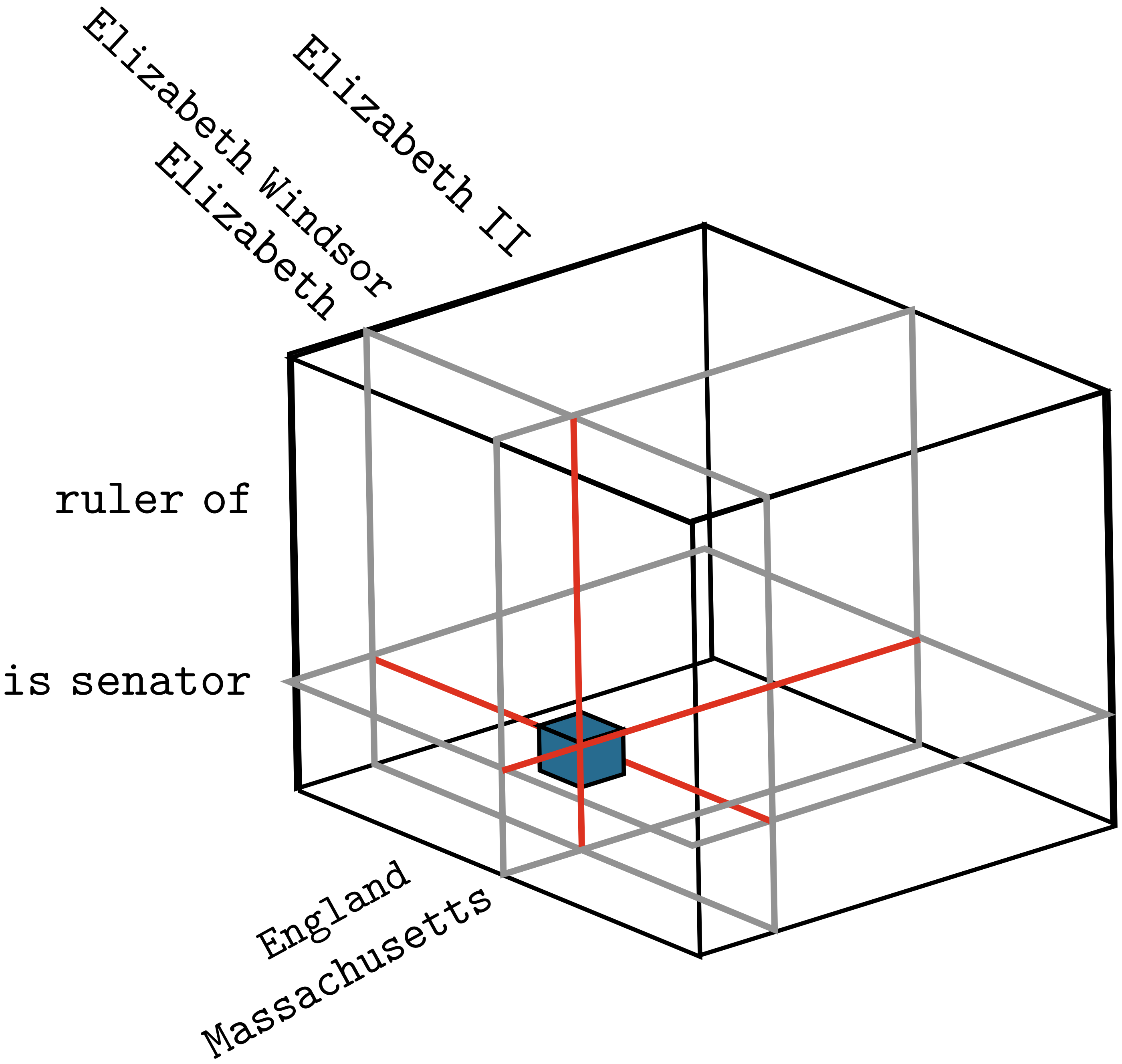
Elizabeth (queen)

Elizabeth (politician)

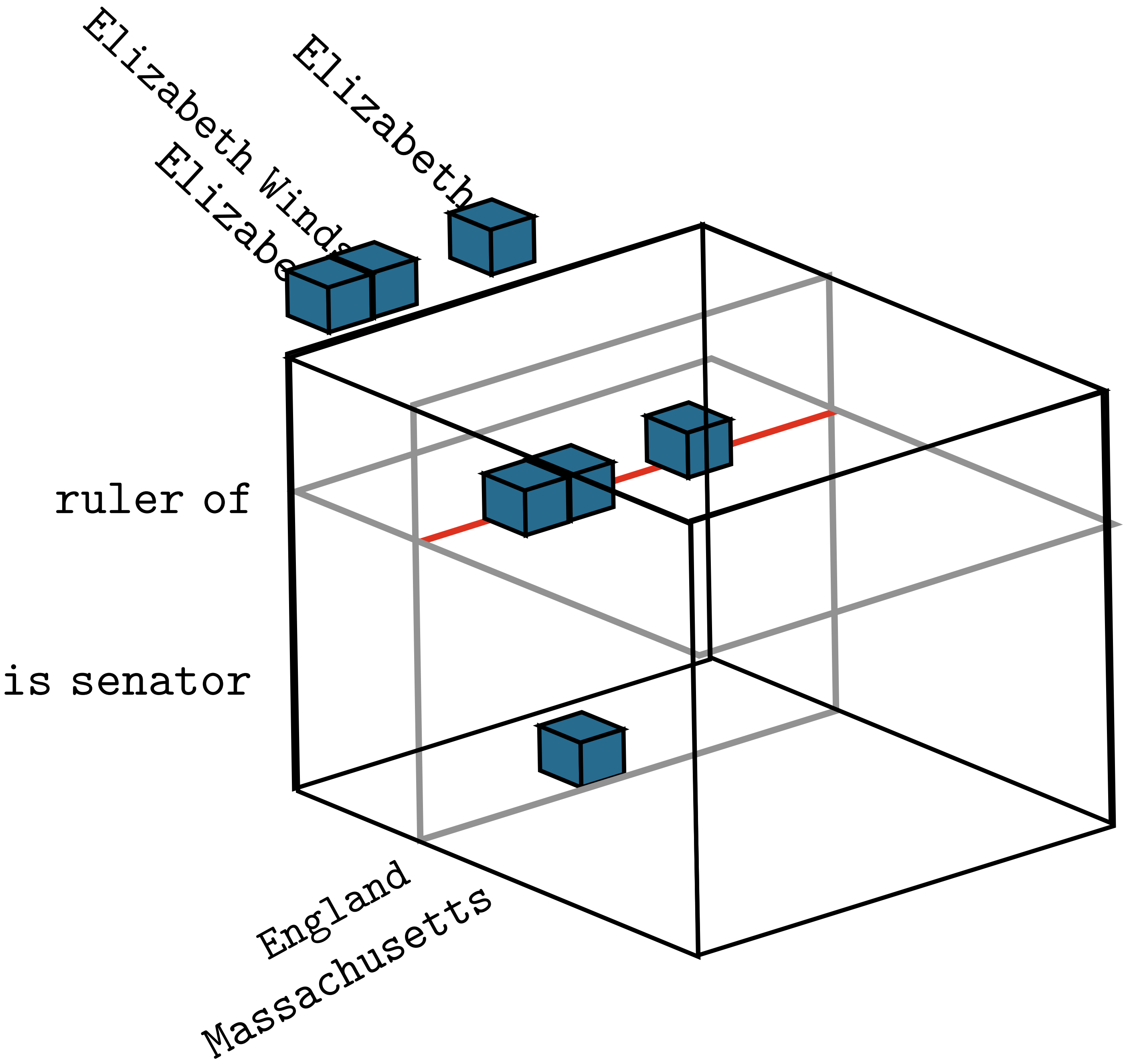
Open information extraction

	Subject	Predicate	Object
Surface terms	Elizabeth Windsor	ruler of	England
	Elizabeth	is queen	United Kingdom
	Elizabeth II	is queen	England.
	Beckham	plays for	LA Galaxy
	Elizabeth	is senator	Massachusetts
	D. Beckham	plays soccer for	England
	Elizabeth Warren	lives	D.C.
	Elizabeth Windsor	was born in	England
	Beckham	was born	United Kingdom
Facts	Elizabeth (queen)	isRulerOf	United Kingdom
	Beckham (soccer player)	isBornIn	United Kingdom
	Beckham (soccer player)	playsFor	LA Galaxy
	Elizabeth (politican)	isSenator	Massachusetts
	Beckham (soccer player)	playsFor	United Kingdom
	Elizabeth (politician)	livesIn	D.C.
	Elizabeth (queen)	isBornIn	United Kingdom

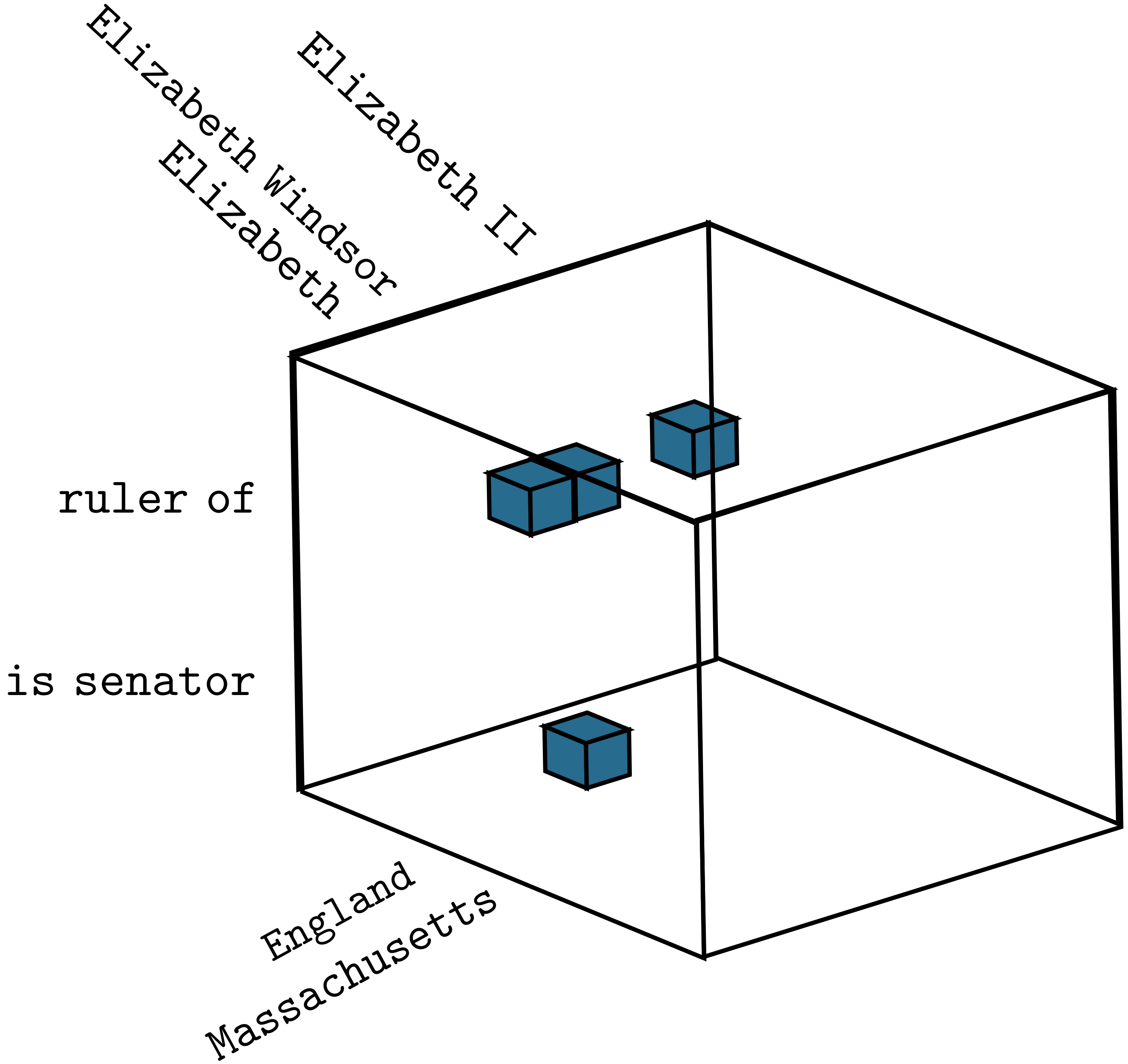
Discovering facts with Boolean tensor decomposition



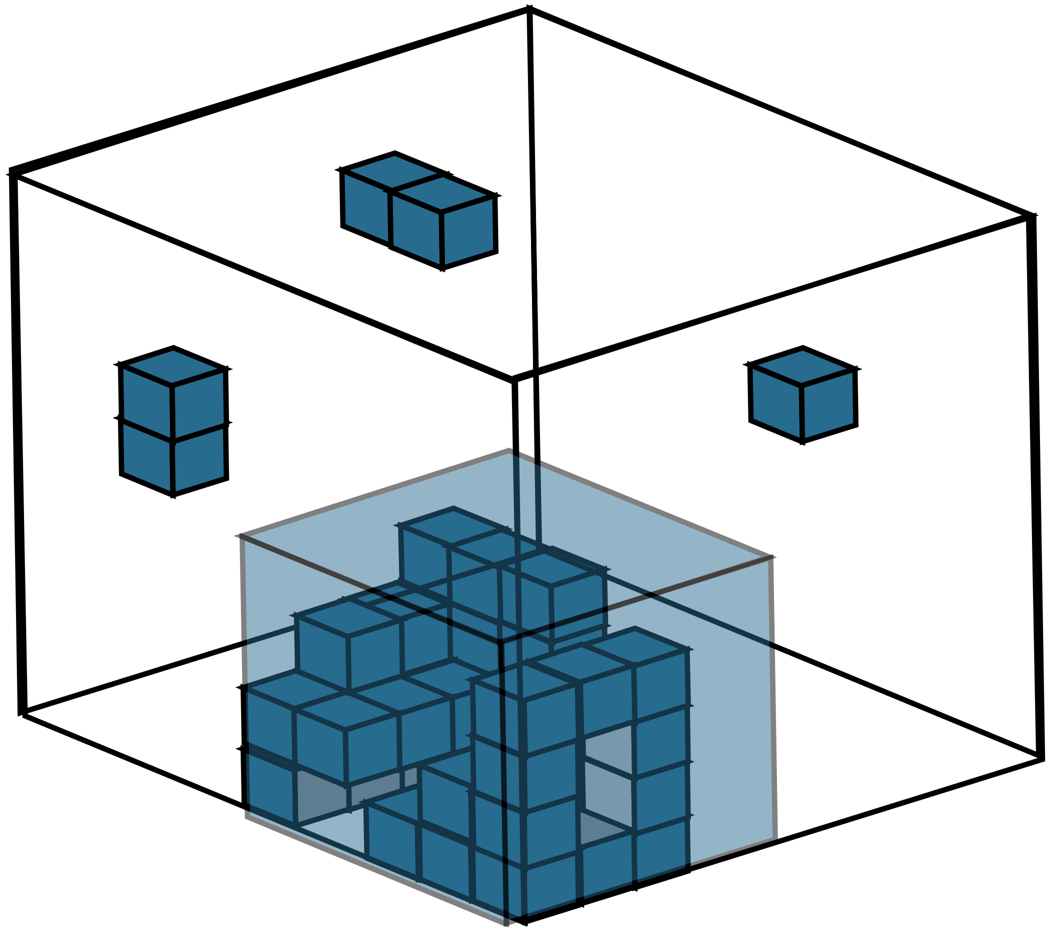
Discovering facts with Boolean tensor decomposition



Discovering facts with Boolean tensor decomposition



Dense blocks



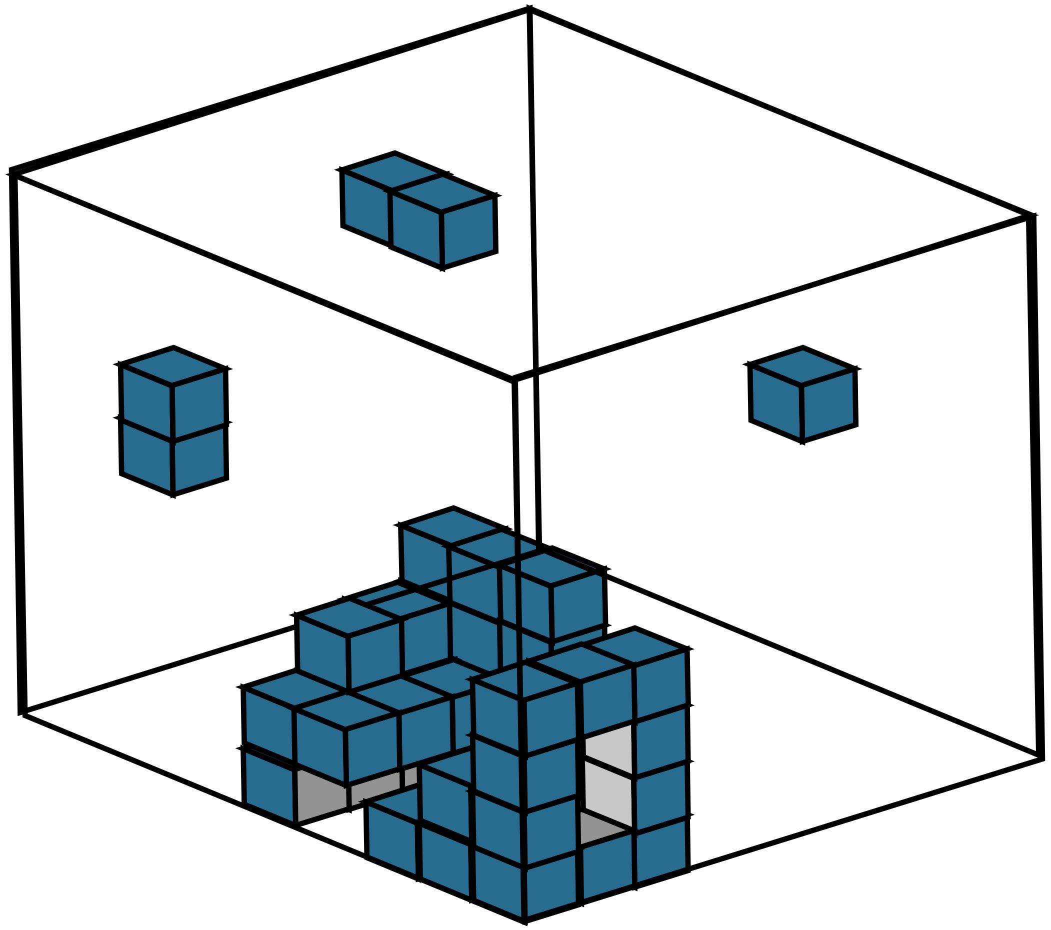
Idea 1:

Find dense blocks in tensor

Idea 2:

Define block as convex hull of dense part

Dense blocks



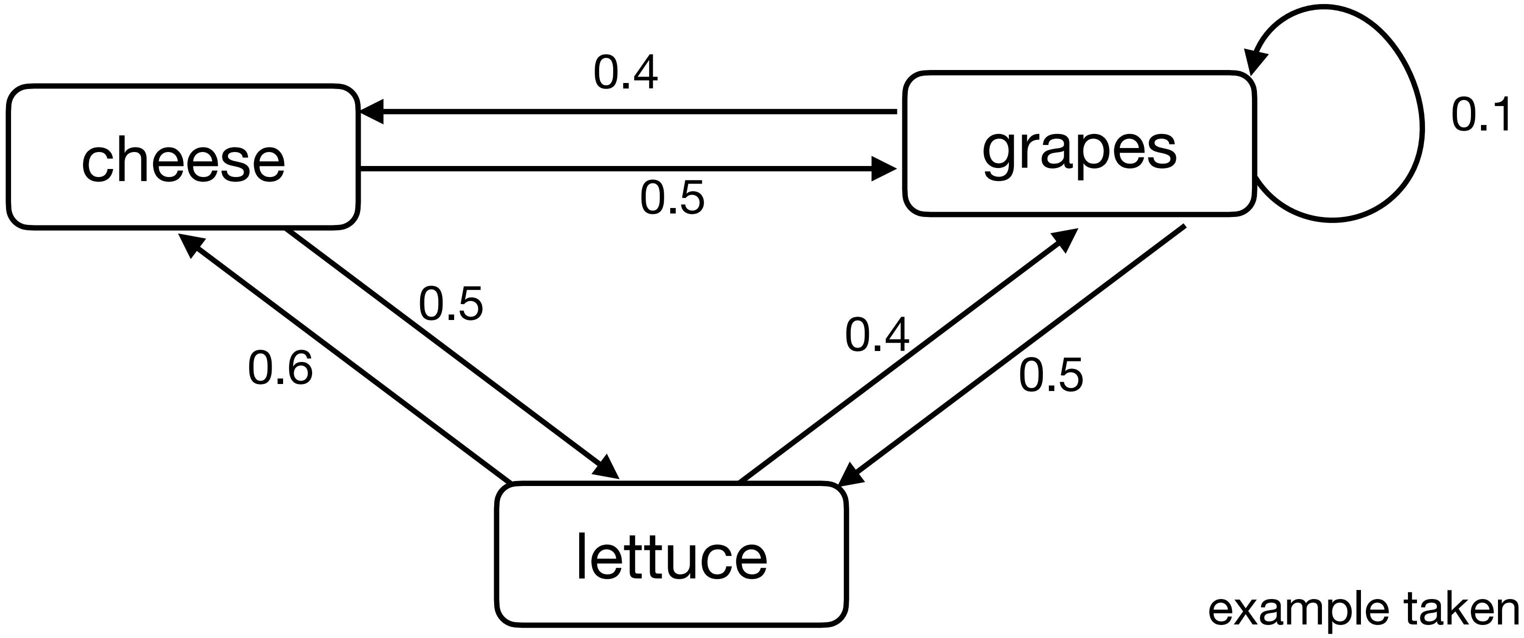
- Define tensor with a graph
- Perform random walks on graph
- Discovered nodes define the convex hull

Markov chain examples

Some animal only eats grapes, lettuce and cheese based on the following rules:

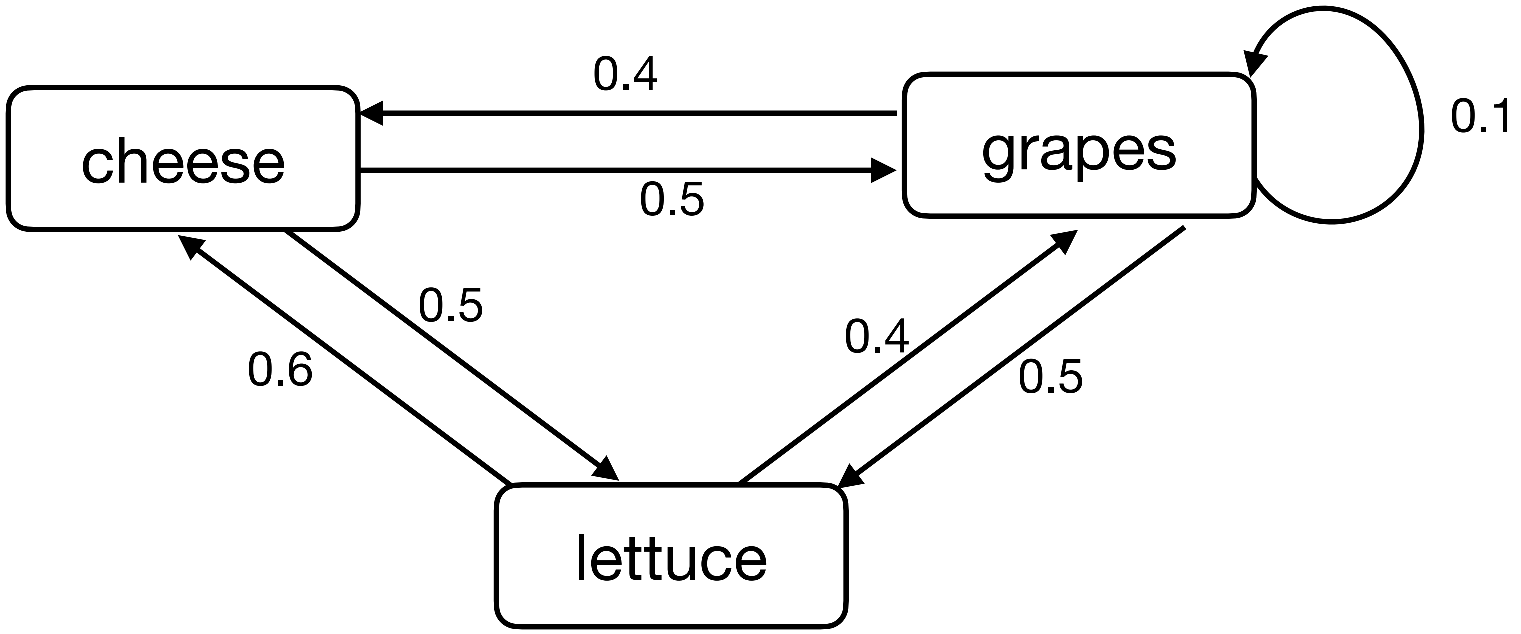
- eats once a day
- cheese today ⇒ tomorrow grapes or lettuce with equal probability
- grapes today ⇒ tomorrow grapes with probability 0.1, cheese with 0.4 or lettuce with 0.5
- lettuce today ⇒tomorrow grapes with probability 0.4 and cheese with 0.6.

states= the three types of food
transition probability= the probability of eating a certain food tomorrow, given what it ate today.
memoryless: tomorrow's choice only depends on what it ate today



example taken from Wikipedia

Markov chain examples



Once we have established the states and transition probabilities, the Markov process itself is modeled by a random walk on this graph.

Note that the food choice for the next day only depends on today's food. hence the memoryless property holds.

Markov chain examples

Consider sequences of n numbers. Two sequences are considered to be “neighbors” if we get one from the other by swapping two neighboring numbers.

states = all possible sequences (orderings) of the n numbers (how many states are there?)
transition probability =

- 0 if two sequences are not neighbors
- 1/(n-1) if they are (why?)

Note that bubble sort is one possible walk on this Markov chain!