Unsupervised Learning
1. Clustering Algorithm

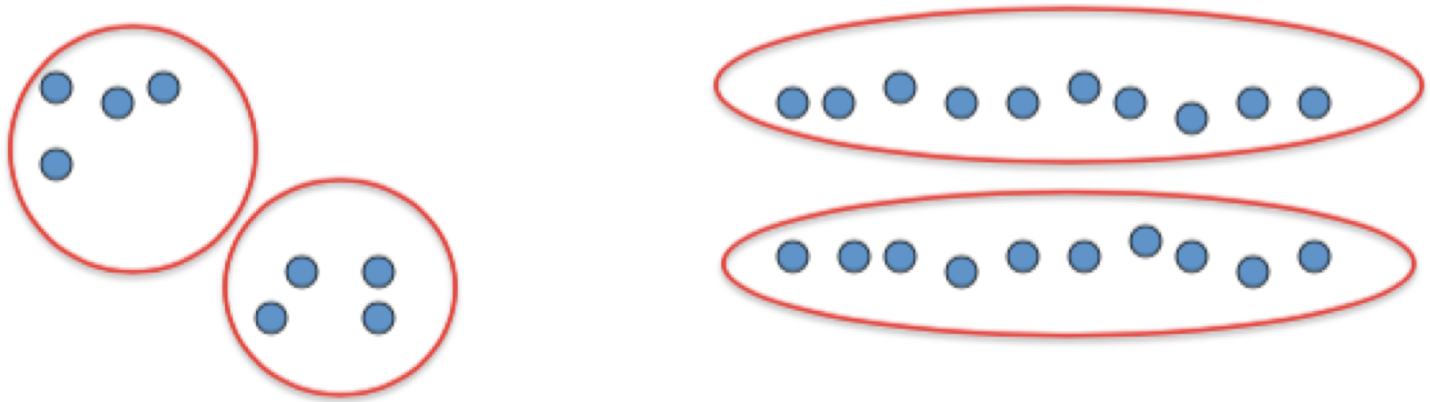
Clustering

Clustering:

- Unsupervised learning
- Requires data, but no labels
- Detect patterns e.g. in
  - Group emails or search results
  - Customer shopping patterns
  - Regions of images
- Useful when don't know what you're looking for
- But: can get gibberish

Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns



- What could "similar" mean?
  - One option: small Euclidean distance (squared)

$$\text{dist}(\vec{x}, \vec{y}) = ||\vec{x} - \vec{y}||_2^2$$

  - Clustering results are crucially dependent on the measure of similarity (or distance) between "points" to be clustered

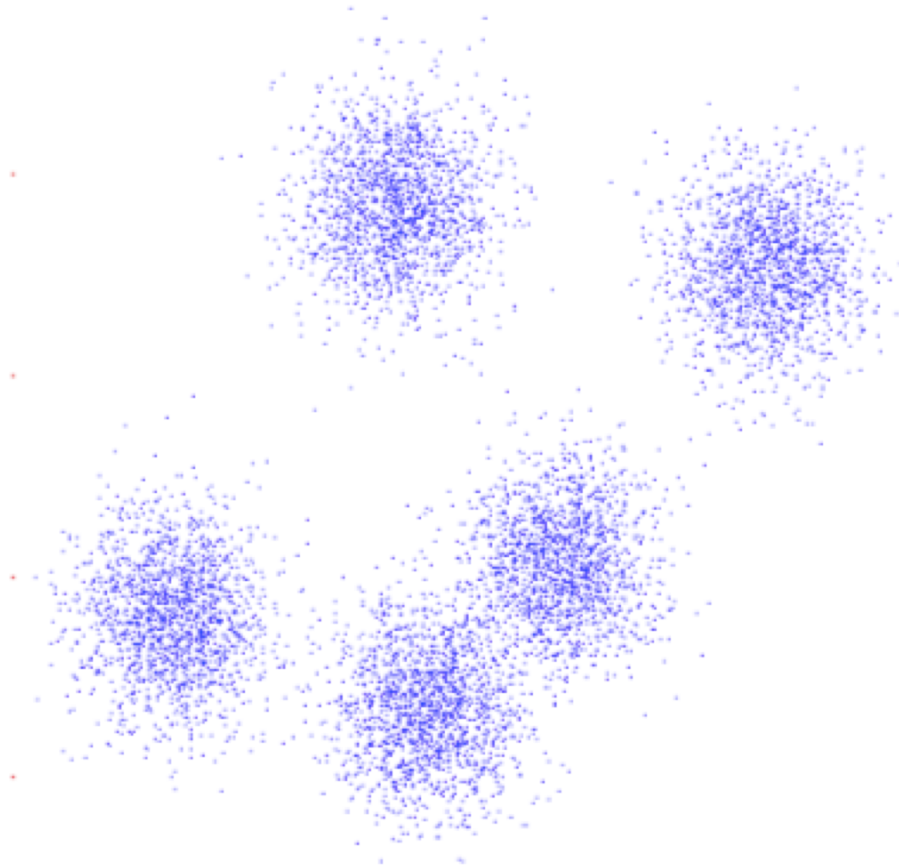
Clustering

Image segmentation

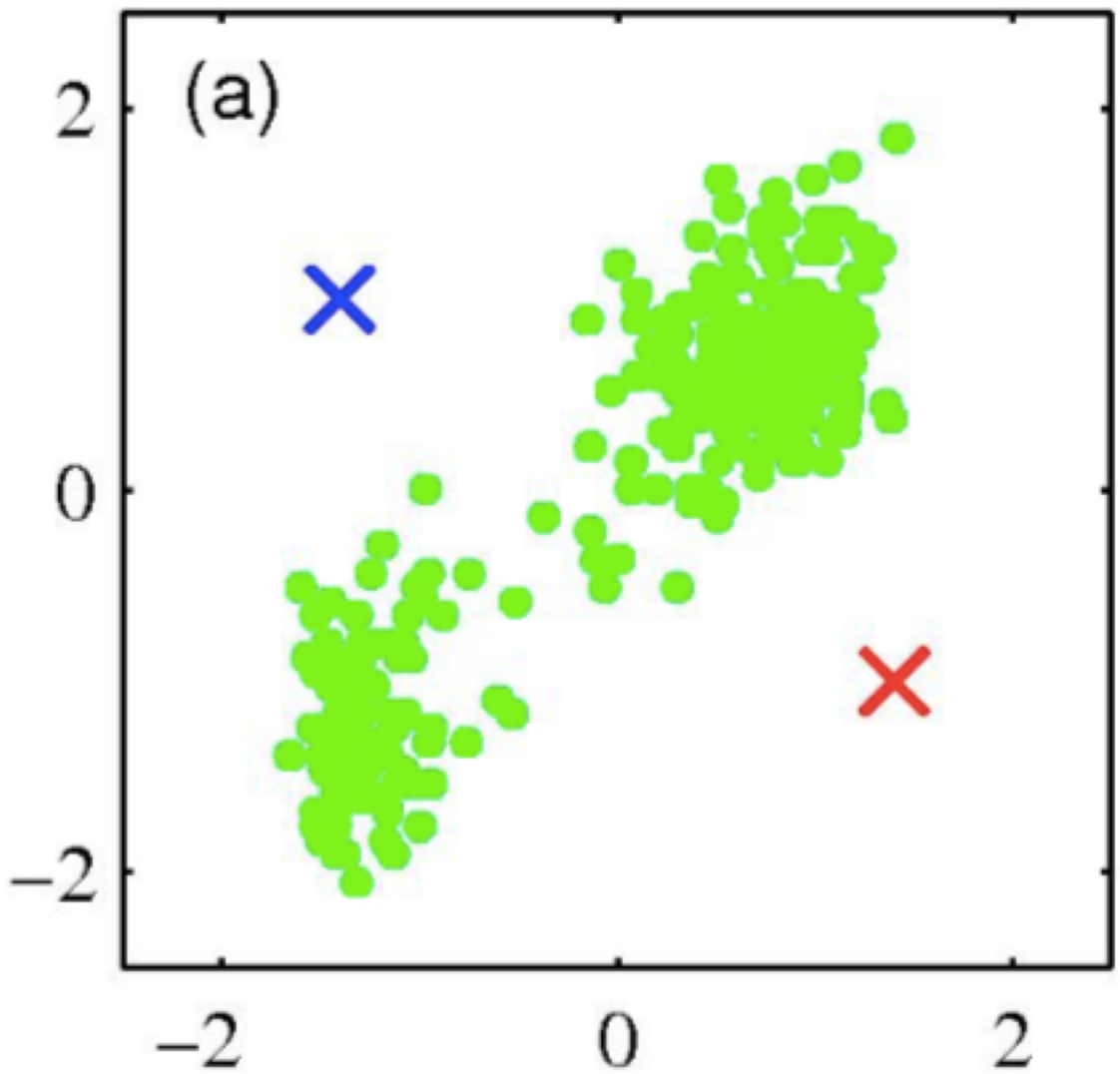
Goal: Break up the image into meaningful or perceptually similar regions

K-Means

- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change



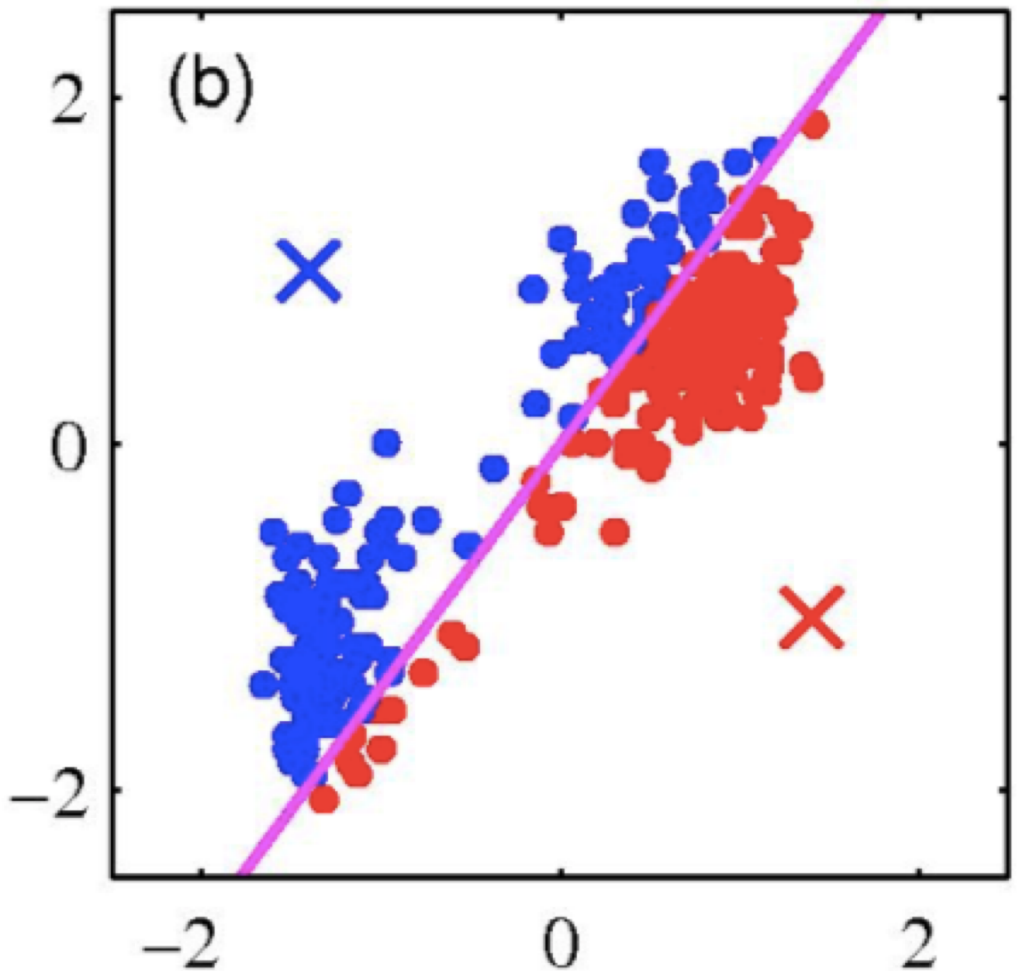
K-Means



- Pick *K* random points as cluster centers (means)

Shown here for *K*=2

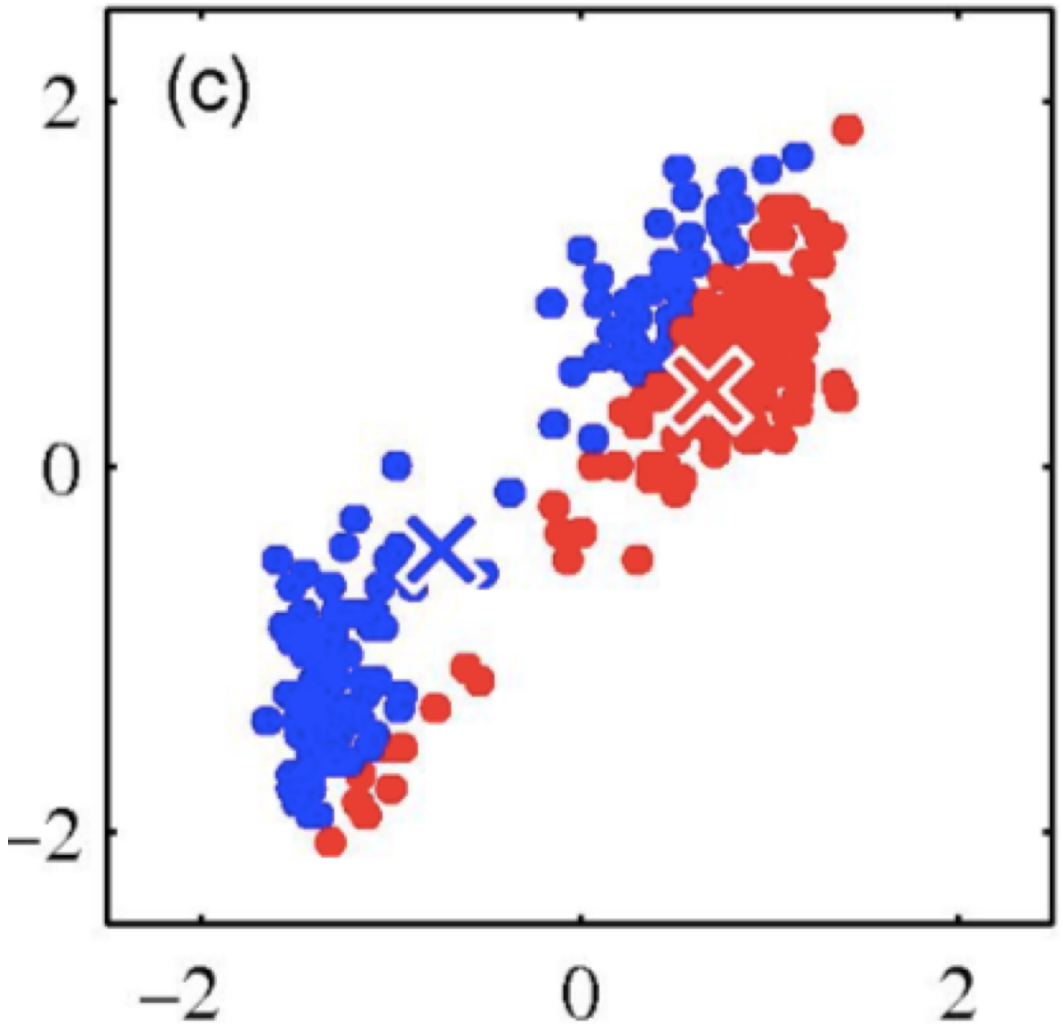
K-Means



Iterative Step 1

- Assign data points to closest cluster center

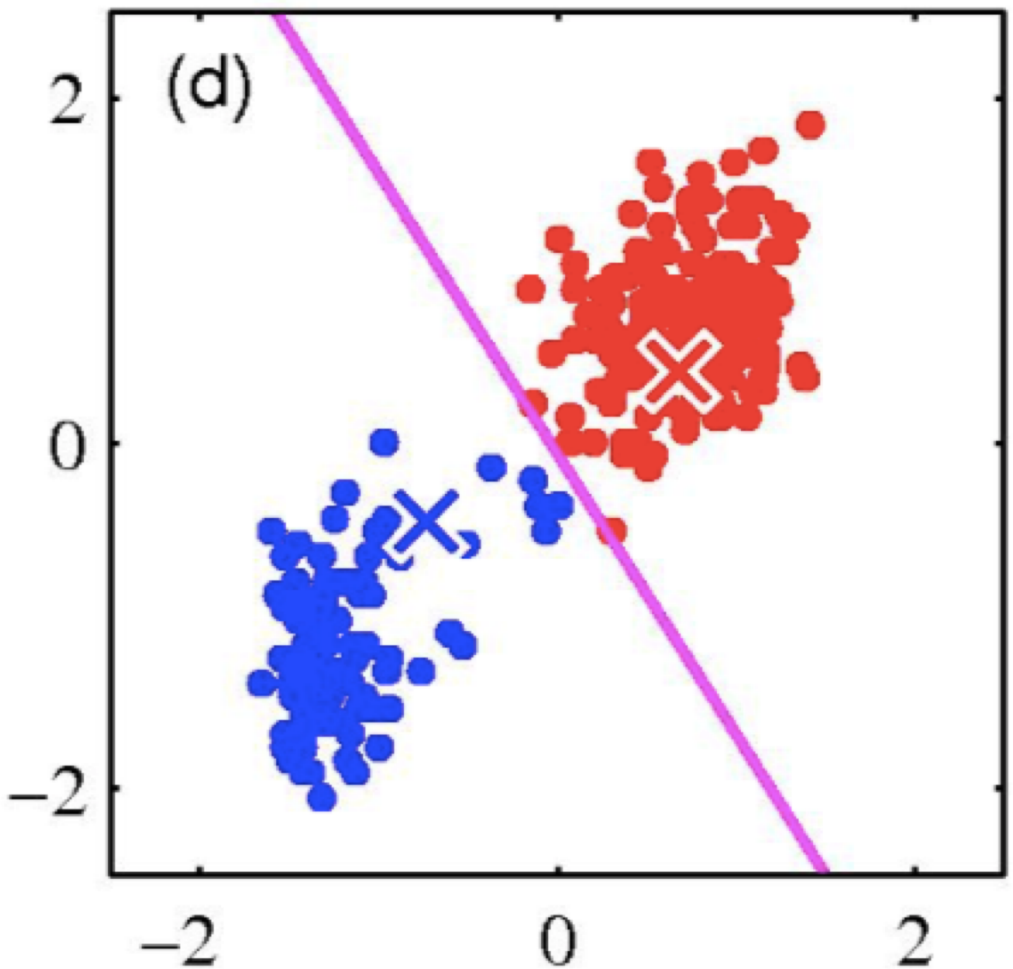
K-Means



Iterative Step 2

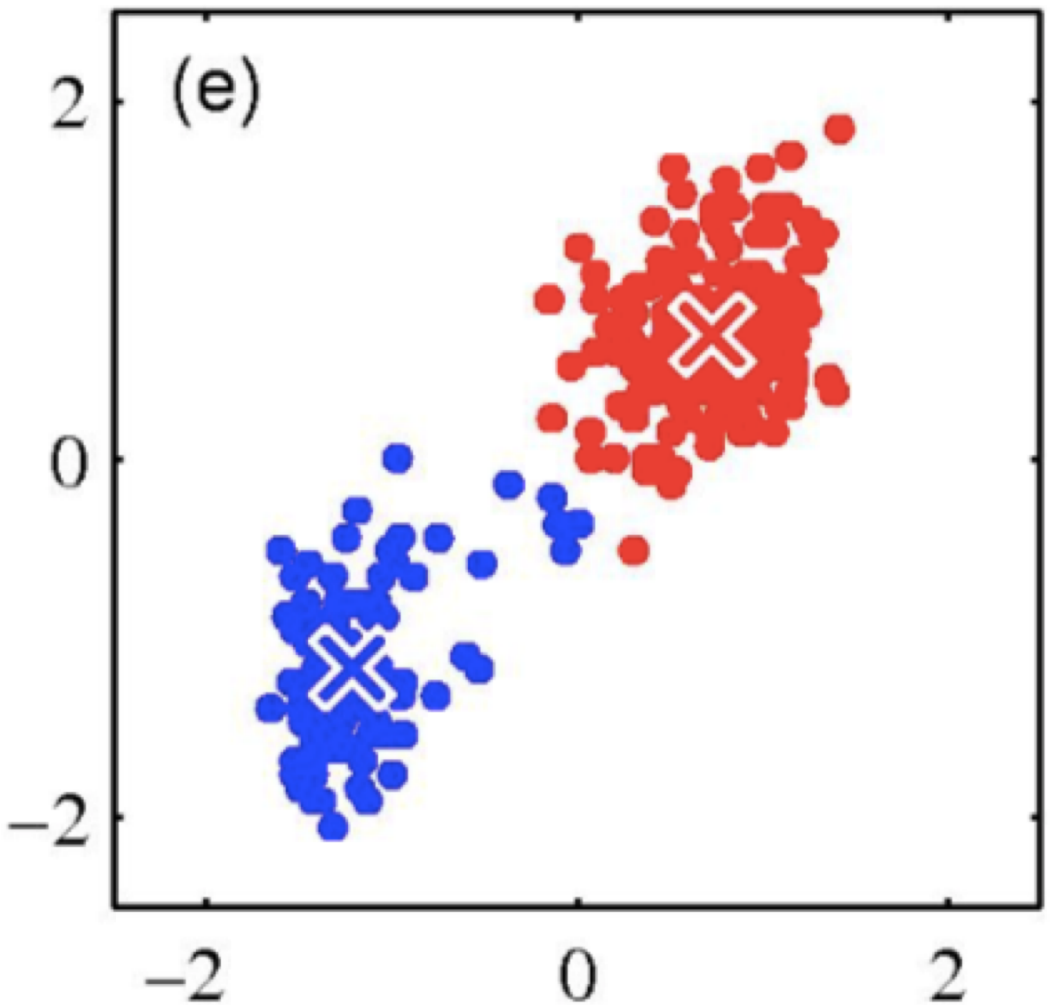
- Change the cluster center to the average of the assigned points

K-Means

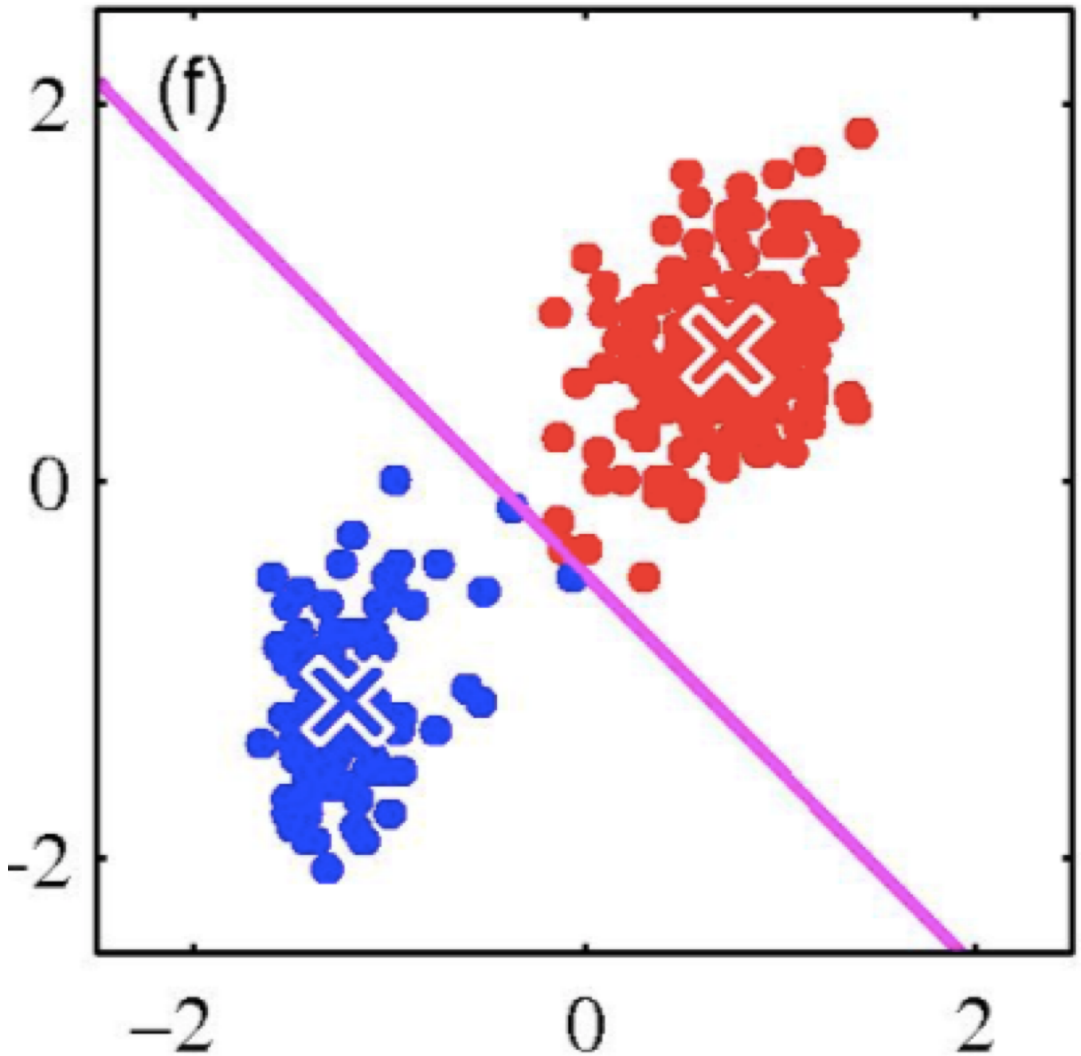


- Repeat until convergence

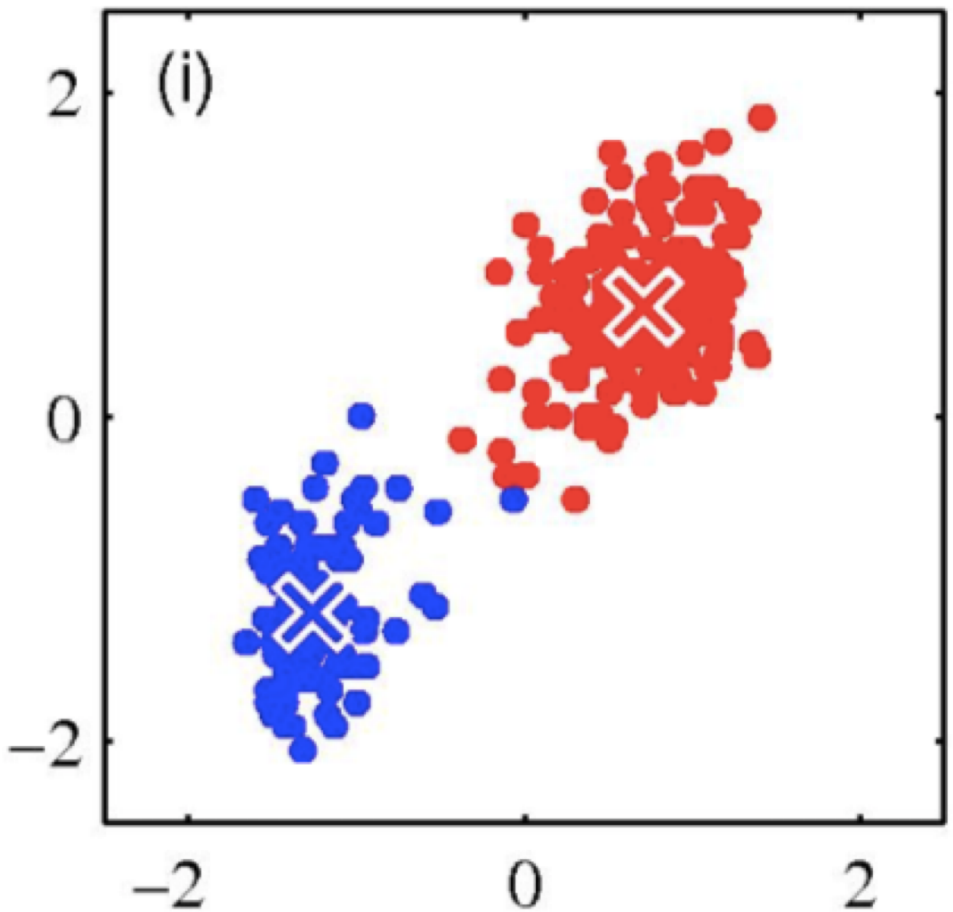
K-Means



K-Means



K-Means



Unsupervised Learning
2. Association Rule Mining

Association rules

- Given a set of transactions **D**, find rules that will predict the occurrence of an item (or a set of items) based on the occurrences of other items in the transaction

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Examples of association rules

{Diaper} → {Beer},
{Milk, Bread} → {Diaper,Coke},
{Beer, Bread} → {Milk},

An even simpler concept: frequent itemsets

- Given a set of transactions **D**, find combination of items that occur frequently

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Examples of frequent itemsets

{Diaper, Beer},
{Milk, Bread}
{Beer, Bread, Milk},

Session outline

- **Task 1:** Methods for finding all frequent itemsets efficiently
- **Task 2:** Methods for finding association rules efficiently

Definition: Frequent Itemset

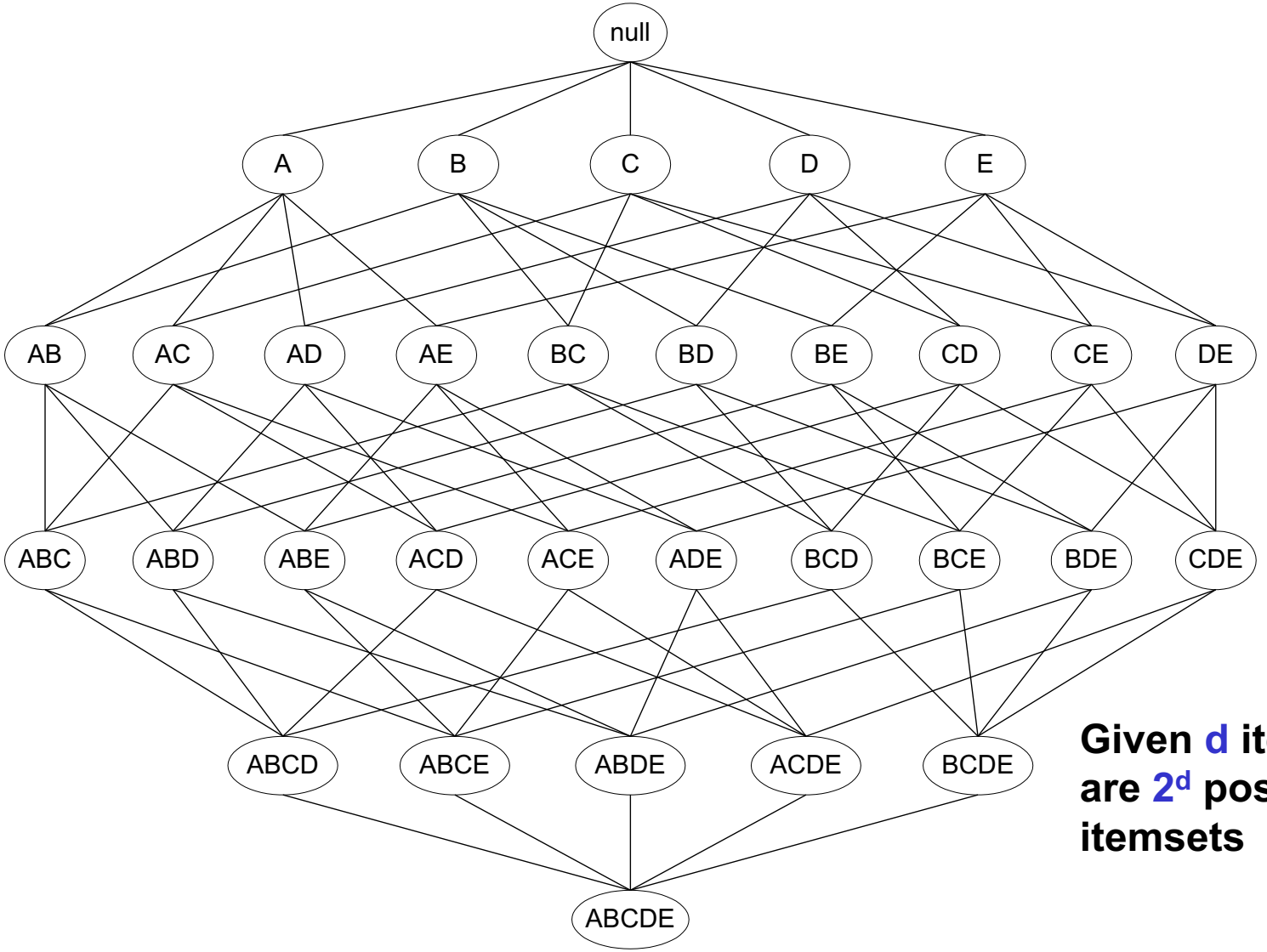
- **Itemset**
  - A set of one or more items
    - E.g.: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset (number of transactions it appears)
  - E.g. **σ({Milk, Bread,Diaper}) = 2**
- **Support**
  - Fraction of the transactions in which an itemset appears
  - E.g. **s({Milk, Bread, Diaper}) = 2/5**
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a ***minsup*** threshold

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Why do we want to find frequent itemsets?

- Find all combinations of items that occur together
- They might be interesting (e.g., in placement of items in a store ☺)
- Frequent itemsets are only positive combinations (we do not report combinations that do not occur frequently together)
- Frequent itemsets aims at providing a summary for the data

How many itemsets are there?

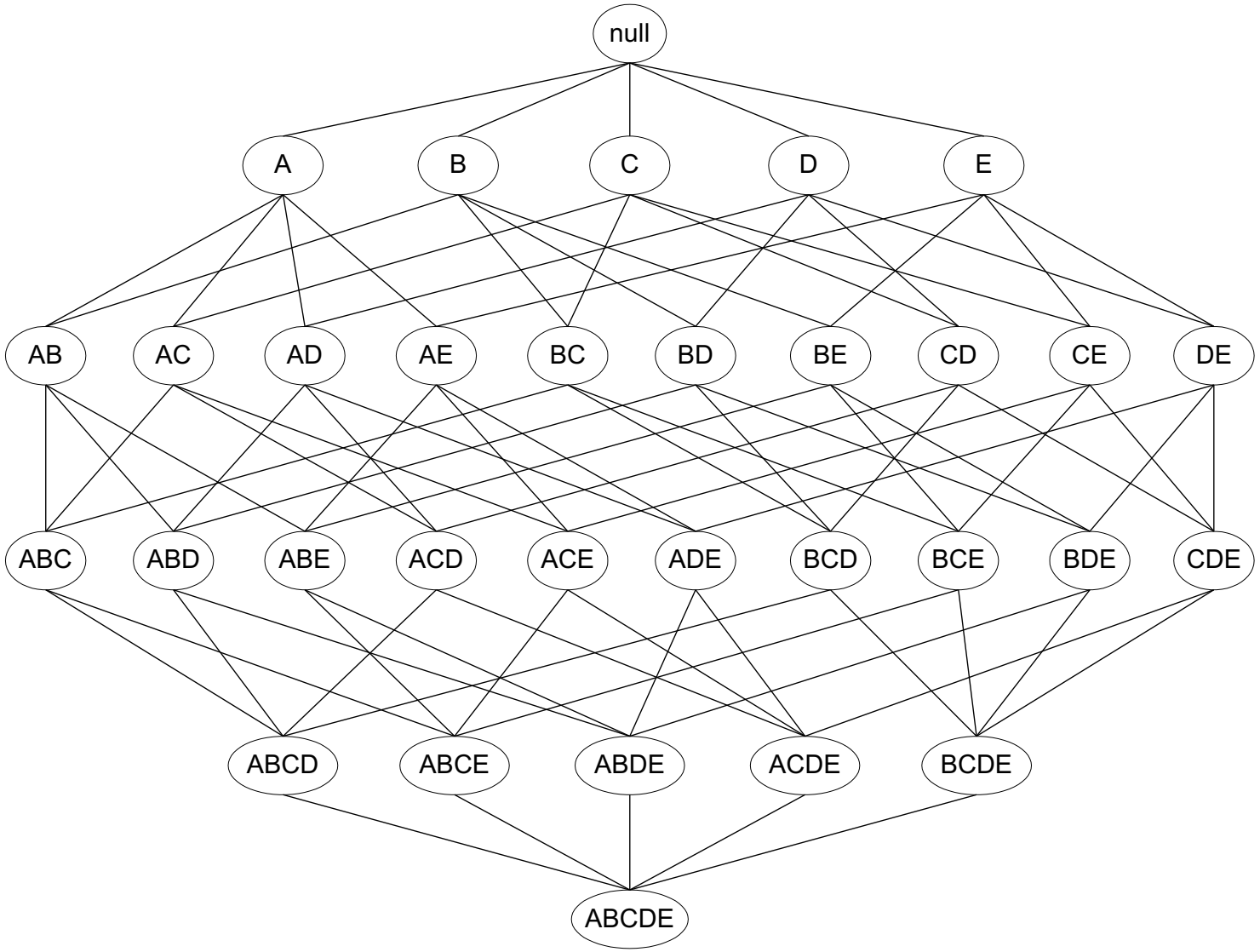


Given d items, there are 2^d possible itemsets

When is the task sensible and feasible?

- If **minsup = 0**, then all subsets of ***I*** will be frequent and thus the size of the collection will be very large
- This summary is very large (maybe larger than the original input) and thus not interesting
- The task of finding all frequent sets is interesting typically only for relatively large values of **minsup**

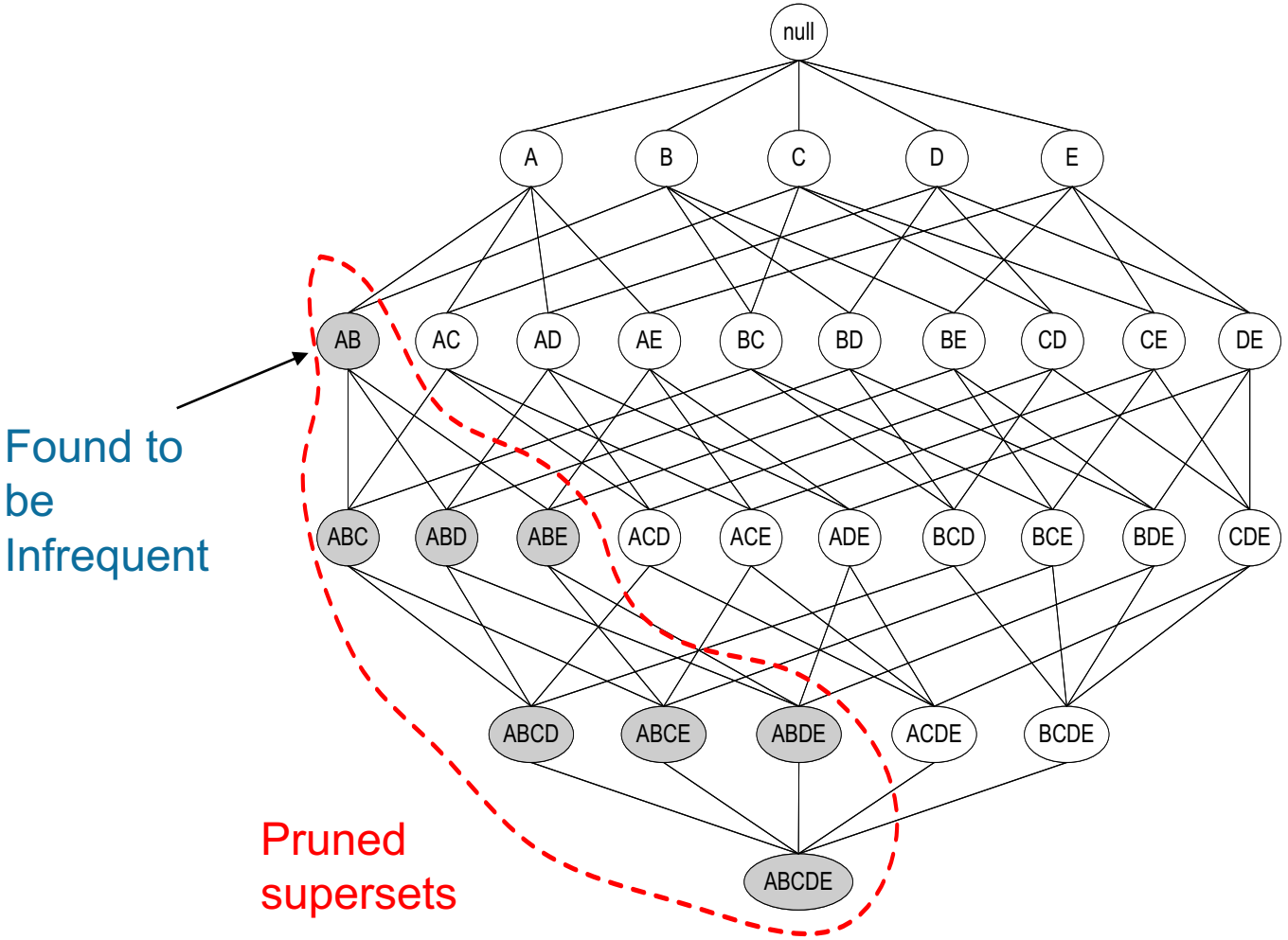
Brute-force approach for finding all frequent itemsets



- Complexity?
  - Match every candidate against each transaction
  - Compute the frequency of each itemset from the data
  - Count in how many transactions each itemset occurs
  - If the support of an itemset is above minsup report it as a frequent itemset
  - Computationally Expensive Approach

Apriori principle

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
  - The support of an itemset ***never exceeds*** the support of its subsets
  - This is known as the ***anti-monotone*** property of support



Illustrating the Apriori principle

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)

Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

minsup = 3/5

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,
${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
6 + 6 + 1 = 13

Apriori Algorithm

1. Find frequent 1-items and put them to L_k ($k=1$)
2. Use L_k to generate a collection of *candidate* itemsets C_{k+1} with size ($k+1$)
3. Scan the database to find which itemsets in C_{k+1} are frequent and put them into L_{k+1}
4. If L_{k+1} is not empty
   - $k=k+1$
   - Goto step 2

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form **X** → **Y**, where **X** and **Y** are non-overlapping itemsets
  - Example:
    {Milk, Diaper} → ***{Beer}***

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Rule Evaluation Metrics**
  - **Support (s)**
    - Fraction of transactions that contain both **X** and **Y**
  - **Confidence (c)**
    - Measures how often items in **Y** appear in transactions that contain **X**

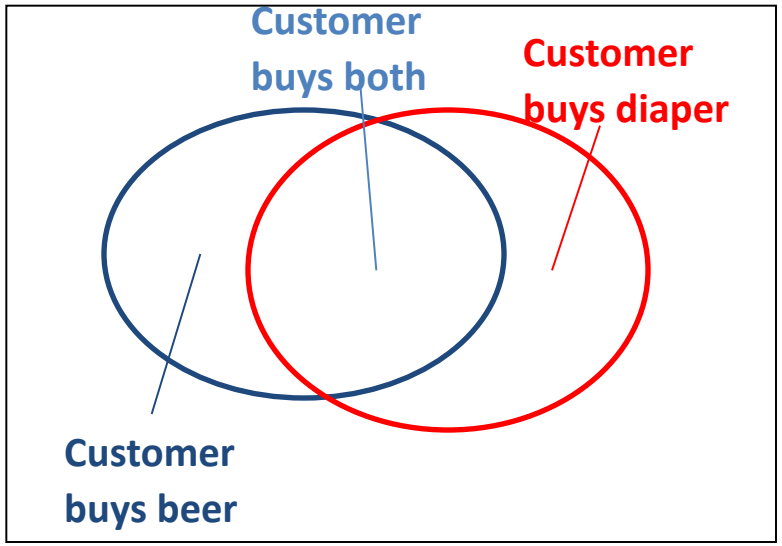
Example:

$$\{\text{Milk}, \text{Diaper}\} \rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Rule Measures: Support and Confidence



Find all the rules $X \rightarrow Y$ with minimum confidence and support

- support, *s*, probability that a transaction contains $\{X \cup Y\}$
- confidence, *c*, ***conditional probability*** that a transaction having **X** also contains **Y**

TID	Items
100	A,B,C
200	A,C
300	A,D
400	B,E,F

Let minimum support 50%, and minimum confidence 50%, we have

- $A \rightarrow C$ (50%, 66.6%)
- $C \rightarrow A$ (50%, 100%)

Example

TID	date	items_bought
100	10/10/99	{F,A,D,B}
200	15/10/99	{D,A,C,E,B}
300	19/10/99	{C,A,B,E}
400	20/10/99	{B,A,D}

What is the ***support*** and ***confidence*** of the rule: {B,D} → {A}

- Support:
  - percentage of tuples that contain {A,B,D} = 75%
- Confidence:

$$\frac{\text{number of tuples that contain } \{A,B,D\}}{\text{number of tuples that contain } \{B,D\}} = 100\%$$

Association-rule mining task

- Given a set of transactions **D**, the goal of association rule mining is to find **all** rules having
  - support ≥ ***minsup*** threshold
  - confidence ≥ ***minconf*** threshold

Mining Association Rules

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements