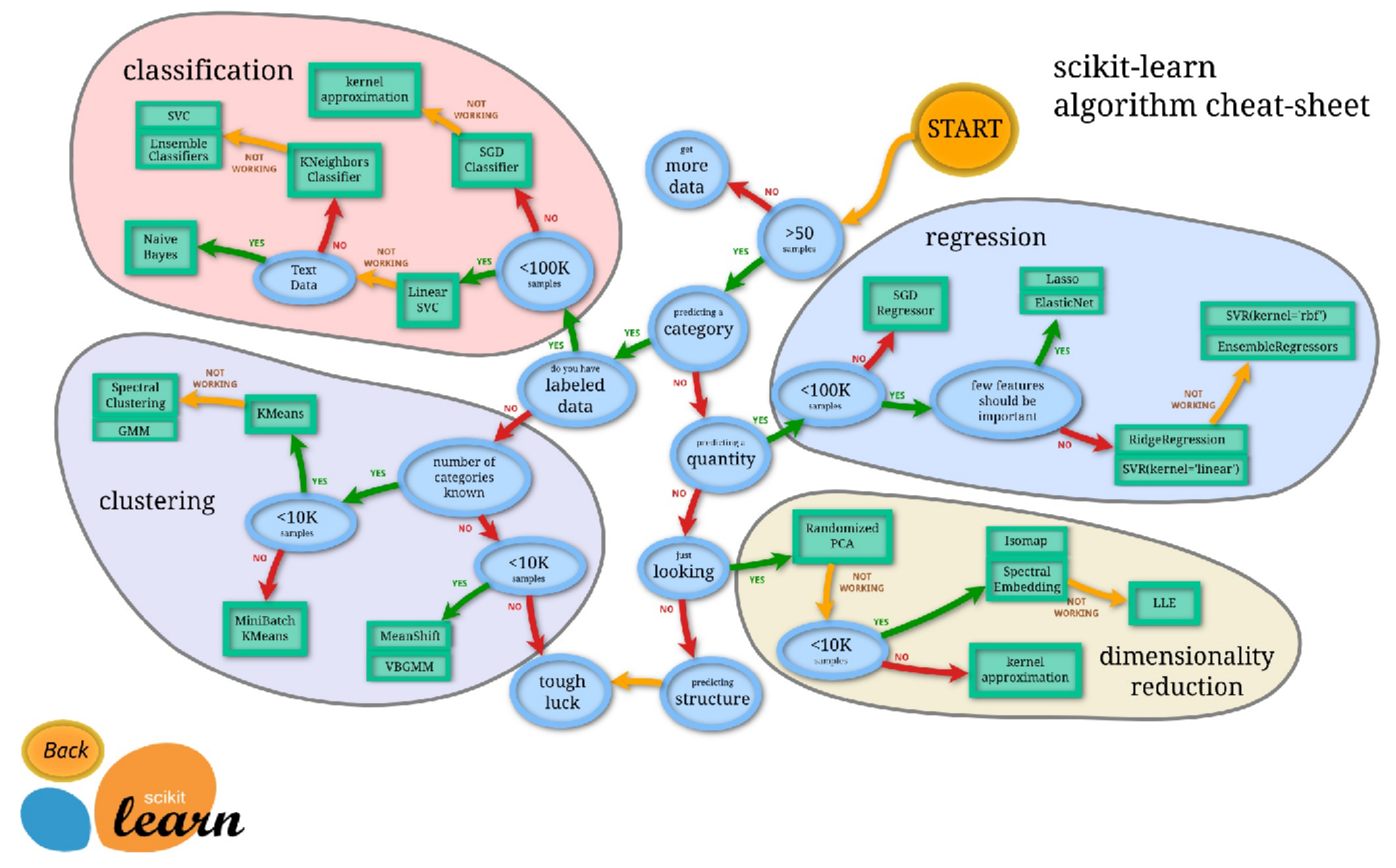
数据分类问题：朴素贝叶斯

Data Classification with Naive Bayes

An Overview of Machine Learning Algorithms



Email Classification

Spam vs Ham

Image Classification

ImageNet/VGG

Document Classification

News, Sports, Finance

Other Problems

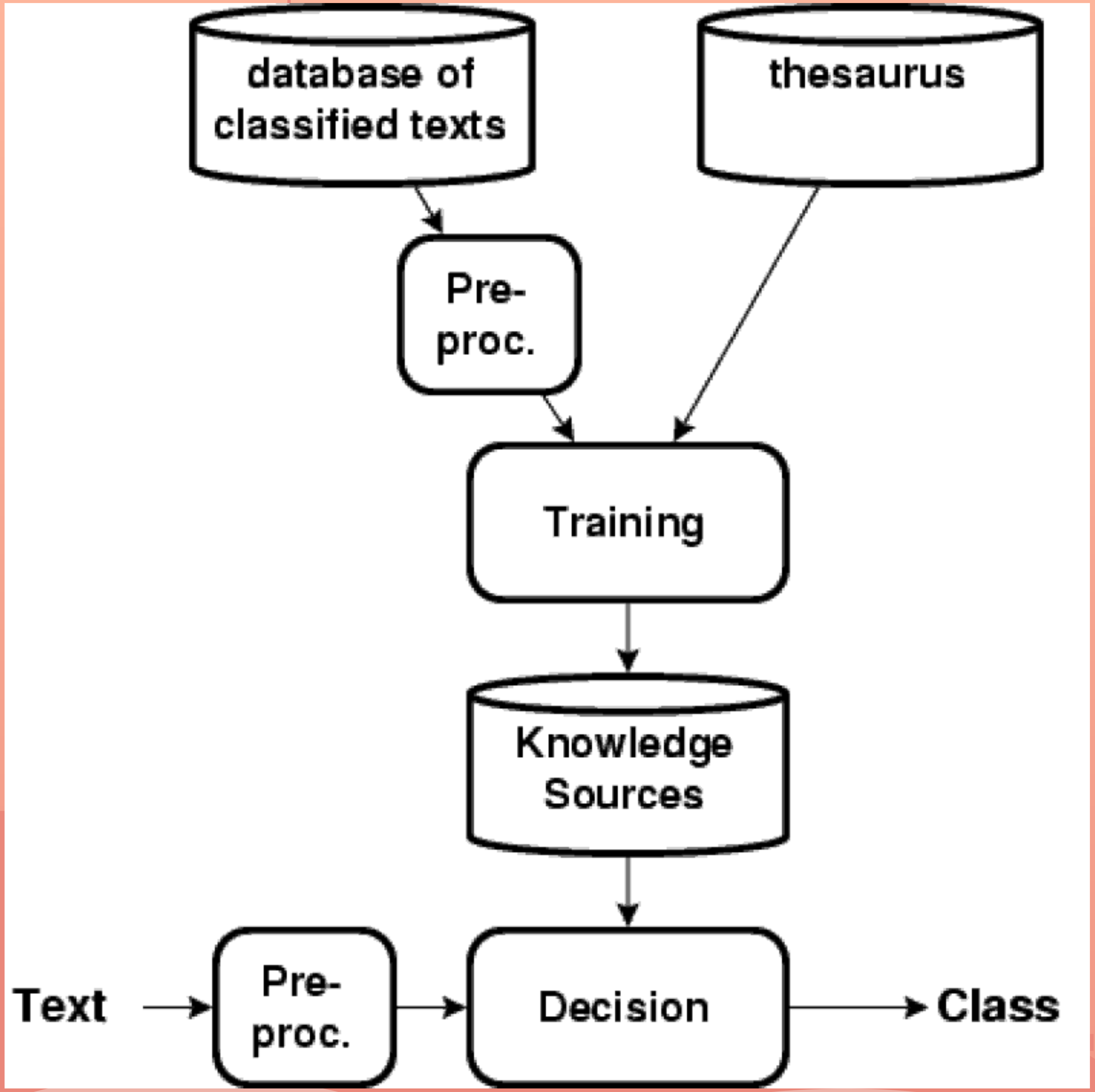
Sentiment Analysis
Movie Ranking

Classification vs Regression

Continuous values: Regression
Discrete values: Classifications

An example:
How is the temperature tomorrow?
How is the whether tomorrow (sunny, rainy, cloudy?)

特性	分类(监督学习)	回归
输出类型	离散数据	连续数据
目的	寻找决策边界	找到最优拟合
评价方法	精度 (accuracy)、混淆矩阵等	SSE (sum of square errors) 或拟合优度



Document Classification

- A document space X
  - Documents are represented in this space – typically some type of high-dimensional space.
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., relevant vs. nonrelevant).
- A training set D of labeled documents with each labeled document $\langle d, c \rangle \in X \times C$

Using a learning method or learning algorithm, we then wish to

learn a classifier Υ that maps documents to classes:

$$\Upsilon : X \rightarrow C$$

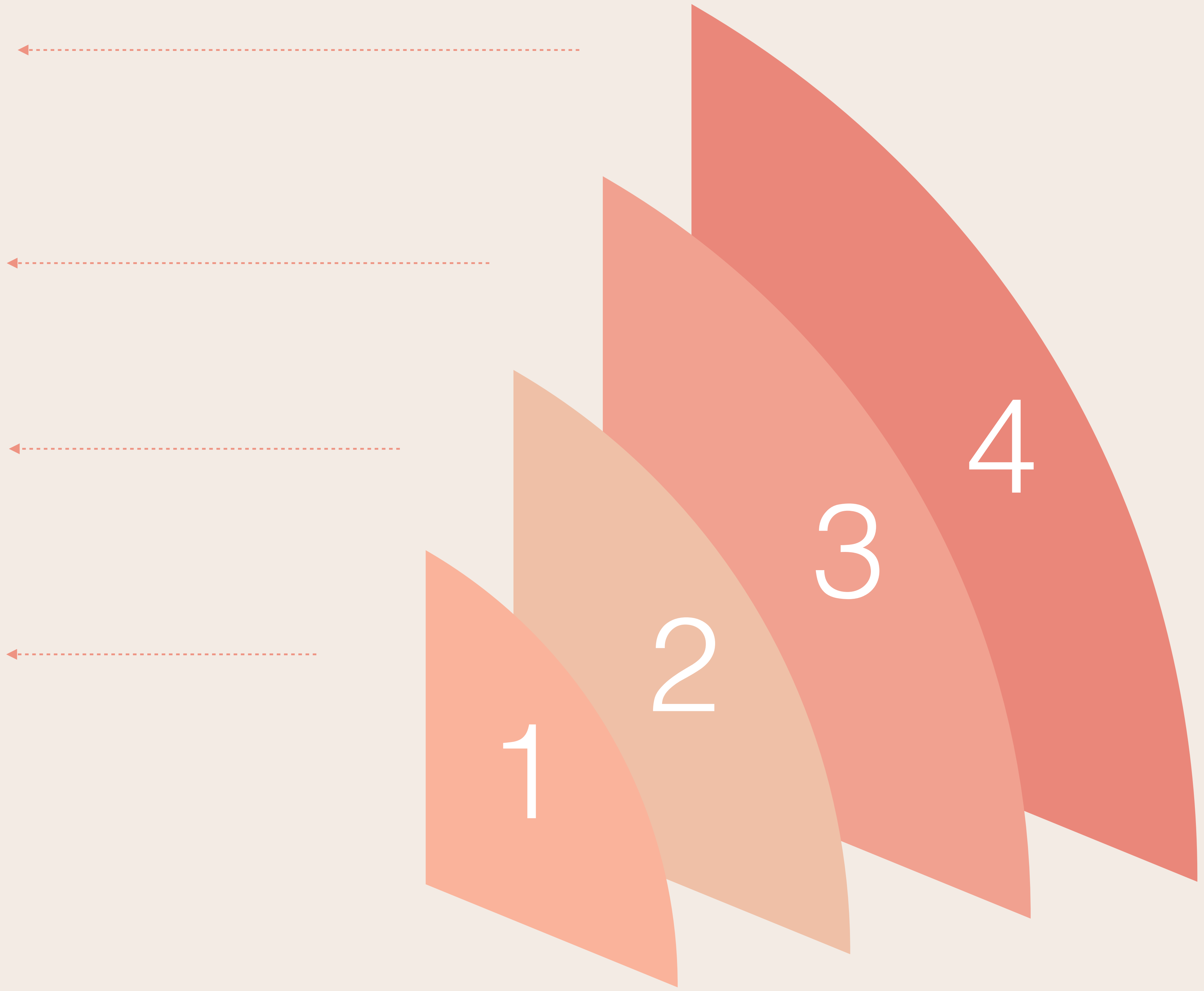
Before You Start

The first step——data exploration

Enron Email Dataset

```
From: ``'' <takworlld@hotmail.com>
Subject: real estate is the only way... gem
oalvgkay
Anyone can buy real estate with no money down
Stop paying rent TODAY !
There is no need to spend hundreds or even
thousands for similar courses
I am 22 years old and I have already purchased
6 properties using the
methods outlined in this truly INCREDIBLE
ebook.
Change your life NOW !
==============================================
===
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
==============================================
===
```

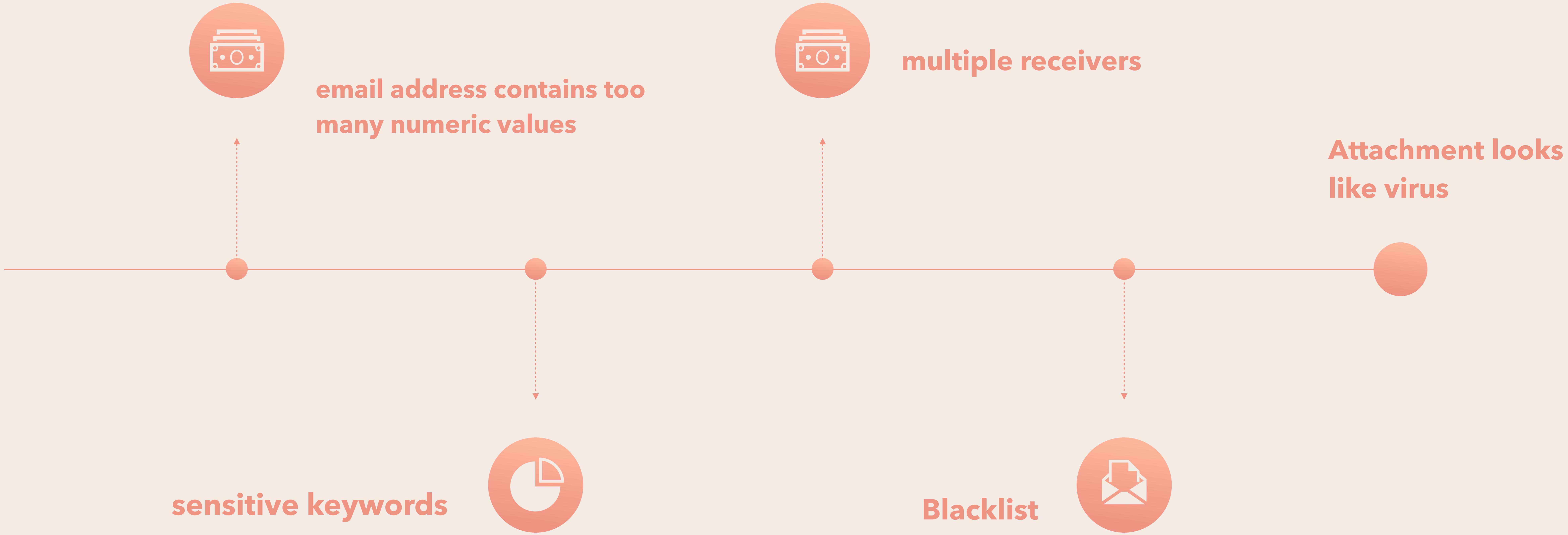
How would you write a program that would automatically detect
and delete this type of message?



Our First Try: Rule-based Approach

If the rule is defined correctly….

But maintaining the rules are challenging



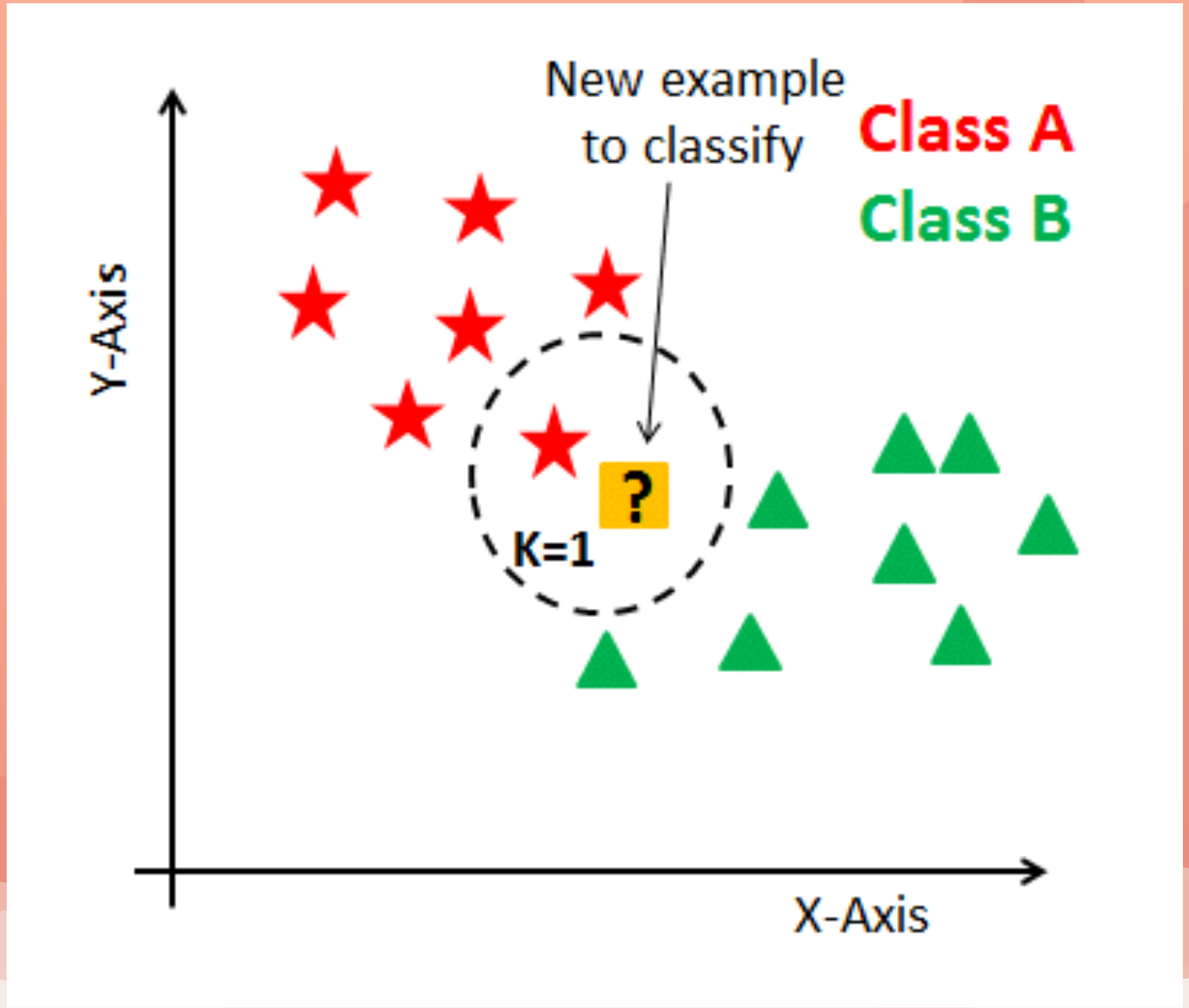
Supervised Learning

- Supervised learning
  - Naive Bayes (simple, common)
  - k-Nearest Neighbors (simple, powerful)
  - Support-vector machines (new, generally more powerful)
  - … plus many other methods
  - No free lunch: requires hand-classified training data
  - But data can be built up (and refined) by amateurs
- Many commercial systems use a mixture of methods

KNN Classifications

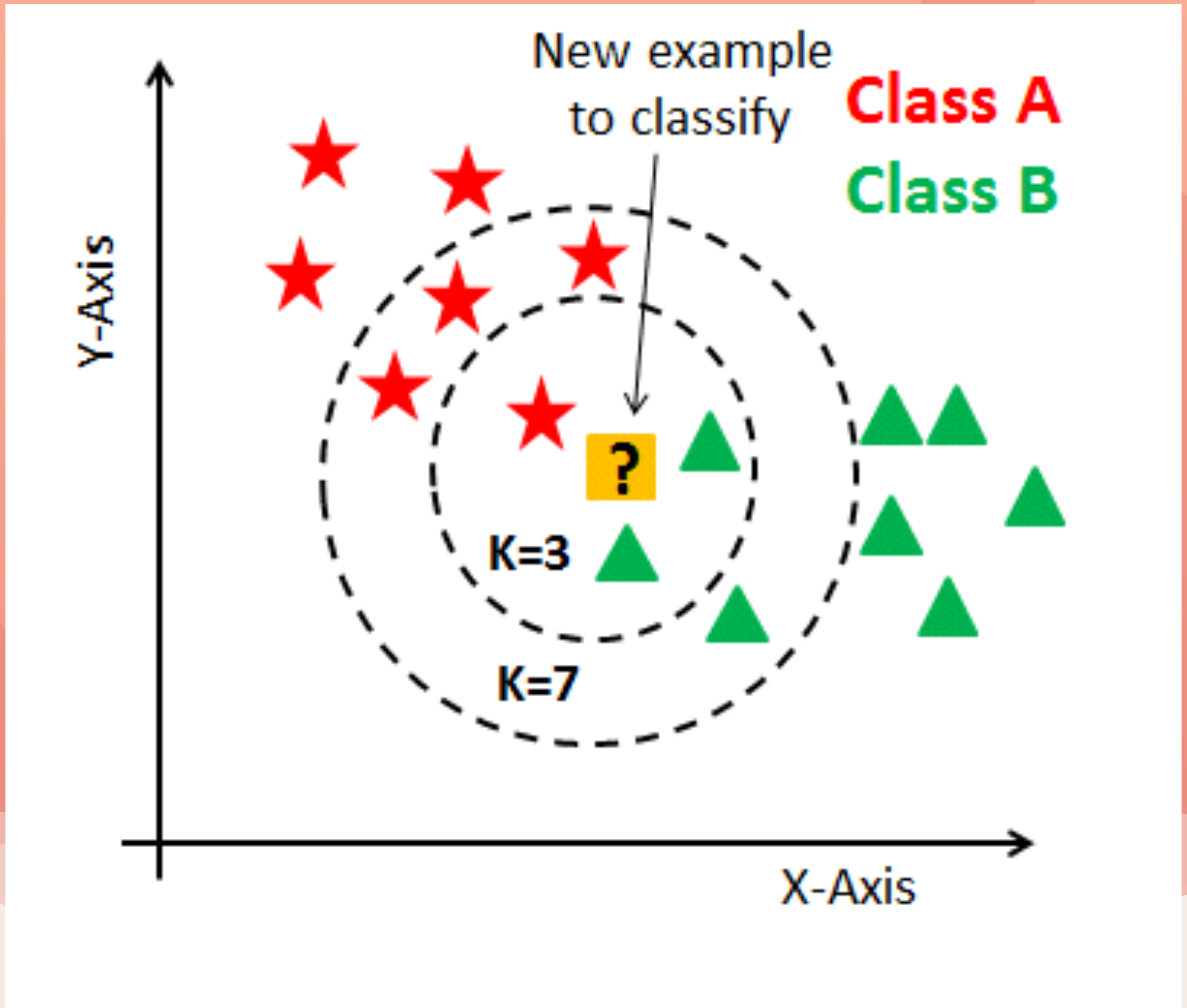
1. Find the k nearest neighbors
2. Ask them to make decisions

The simple part: vote for decisions
The hard part: find the neighbors



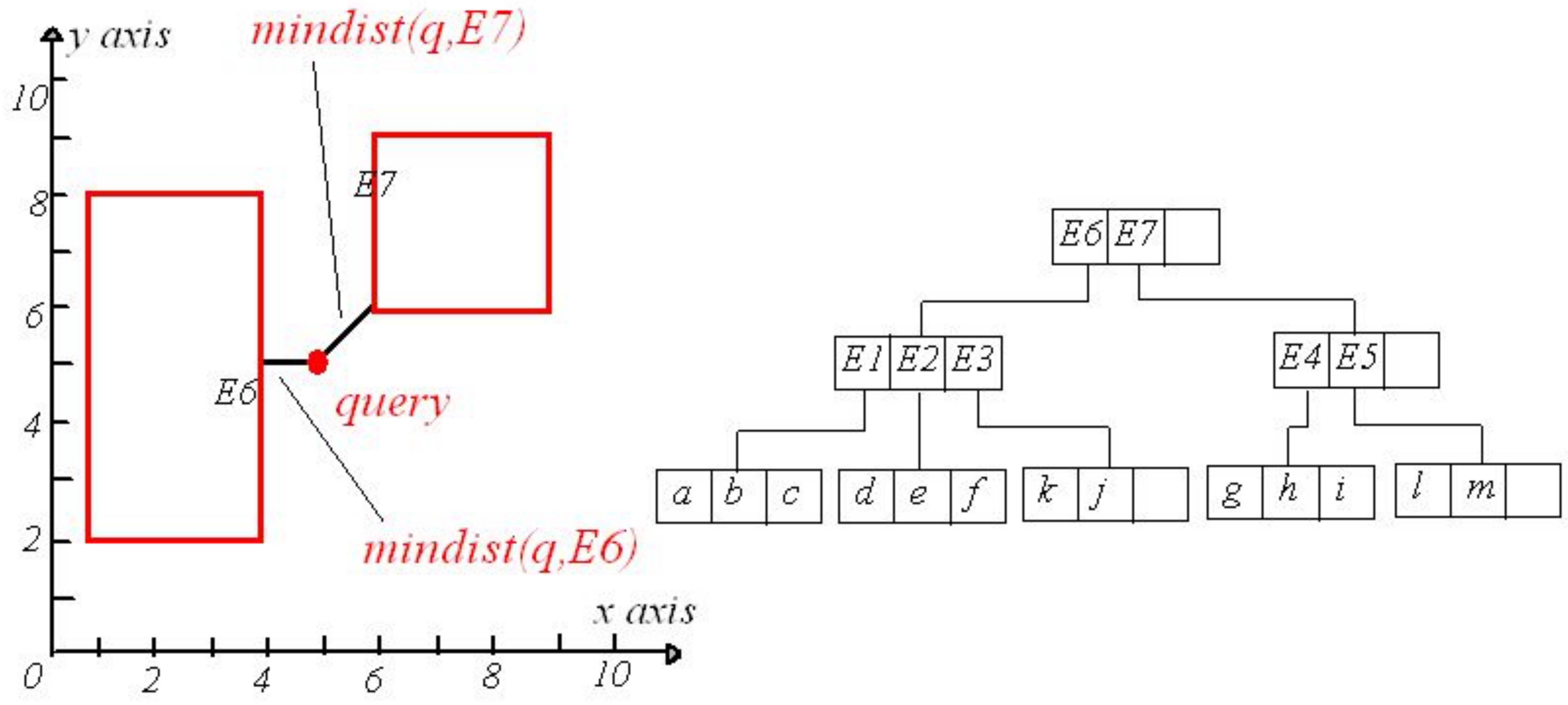
How to decide K??

- K is the most important parameter
- K=N a large class becomes larger
- K=1 bias result



Nearest Neighbor Processing

- The R-tree can accelerate NN search, too.
- Concept: *mindist*(*q*, *E*)
  - The minimum distance between a point *q* and a rectangle *E*



Curse of Dimensionality

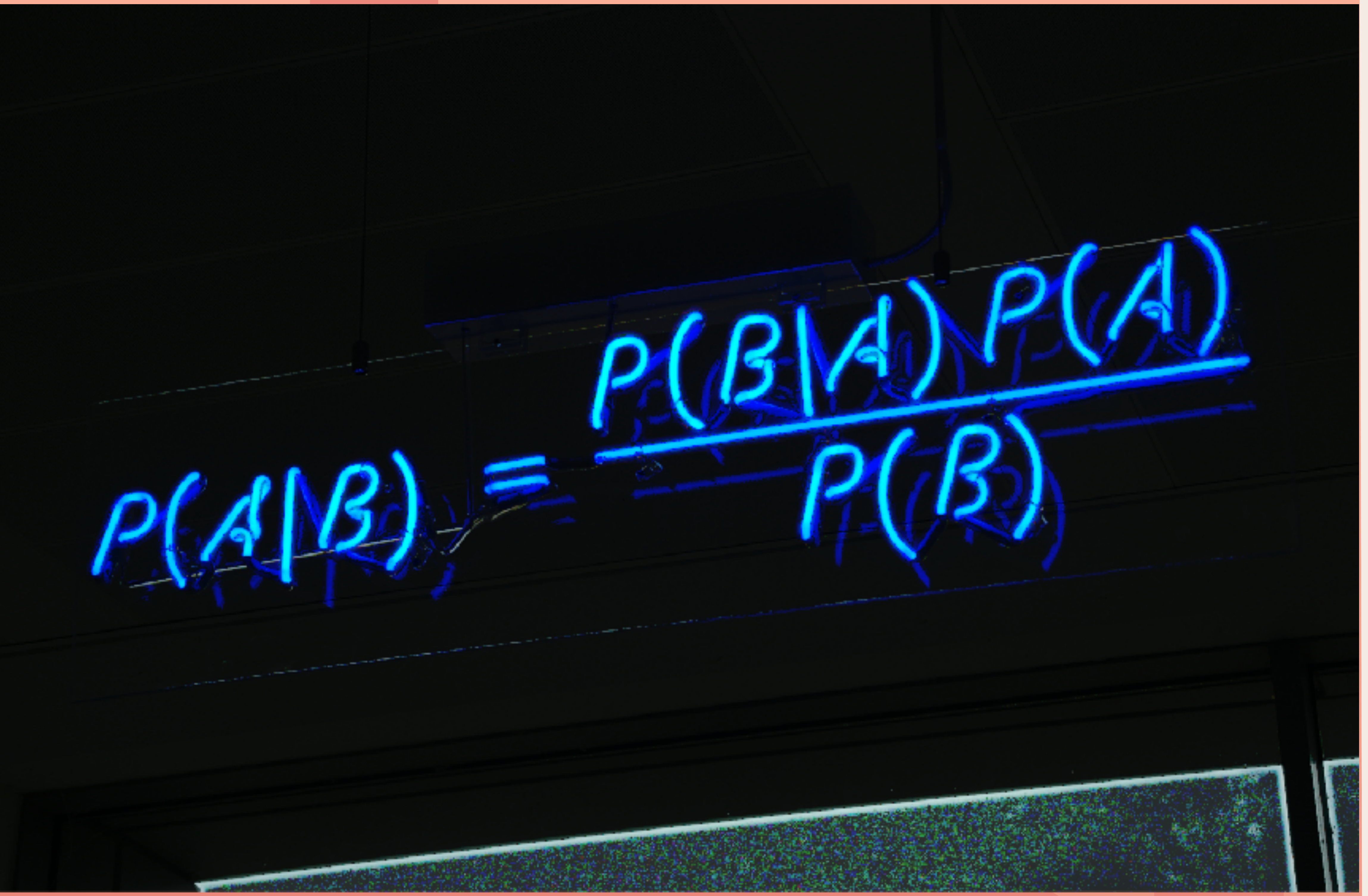
In the case of high dimensionality (d>20), all data are far from each other：

Suppose we want to sample each dimension with 100 unique samples.
In the case of 3d space, we need 100*100*`100=1million data to get the same sampling results.

If two points have a distance 0.9 in a [0, 1] space, they are far from each other.
In the case of 100d space, their distance is 0.9^{10}=$2.6*10^{-5}$

Reduction of Dimensionality

Principal Component Analysis(PCA)
Space filling curve
I-Distance
LDA (latent dirichlet allocation)s
LSH (Locality Sensitive Hashing)



Naive Bayes Classification

Low Cost and Simple

High Precision

Feasible for Big Data

Bag of Words Model

$\gamma($ I **love** this movie! It's **sweet**, but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun**… It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to $)=c$

Bag of Words Model

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$)=c$

The Basic Naive Bayes Rule

- For a document d and a class c

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

Continue

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(c|d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}}\, P(d|c)P(c)$$

Dropping the denominator

$$= \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n | c)P(c)$$

Document d represented as features x1..xn

Independent Assumption

- **Bag of Words assumption:** Assume position doesn't matter
- **Conditional Independence:** Assume the feature probabilities $P(x_i | c_j)$ are independent given the class c.

$$P(x_1, x_2, \ldots, x_n | c)$$

$$P(x_1, \ldots, x_n | c) = P(x_1 | c) \bullet P(x_2 | c) \bullet P(x_3 | c) \bullet \ldots \bullet P(x_n | c)$$

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n | c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c_j) \prod_{x \in X} P(x | c)$$

So, How it applies?

- simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Exception: Words Never Appear

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive (*thumbs-up*)**?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Laplacian Smoothing

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1\right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Training a Naive Bayes

- First, define your Vocabulary

- Calculate $P(c_j)$ terms
  - For each c_j in C do
    $docs_j \leftarrow$ all docs with class $= c_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in $Vocabulary$
    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha \, |Vocabulary|}$$

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|*c*) = (0+1) / (8+6) = 1/14

P(Japan|*c*) = (0+1) / (8+6) = 1/14

P(Chinese|*j*) = (1+1) / (3+6) = 2/9

P(Tokyo|*j*) = (1+1) / (3+6) = 2/9

P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:

P(c|d5) $\propto$ 3/4 * $(3/7)^3$ * 1/14 * 1/14 ≈ 0.0003

P(j|d5) $\propto$ 1/4 * $(2/9)^3$ * 2/9 * 2/9 ≈ 0.0001

Note： chinese appears 3 times in d5， so its probability is repeated 3 times

Floating Problem

- Multiplying lots of probabilities can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$
  - Better to sum logs of probabilities instead of multiplying probabilities.
- Class with highest un-normalized log probability score is still most probable.

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} \log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)$$

- Model is now just max of sum of weights

How to Measure the Effectiveness

	correct	not correct
selected	true positive	false positive
not selected	false negative	true negative

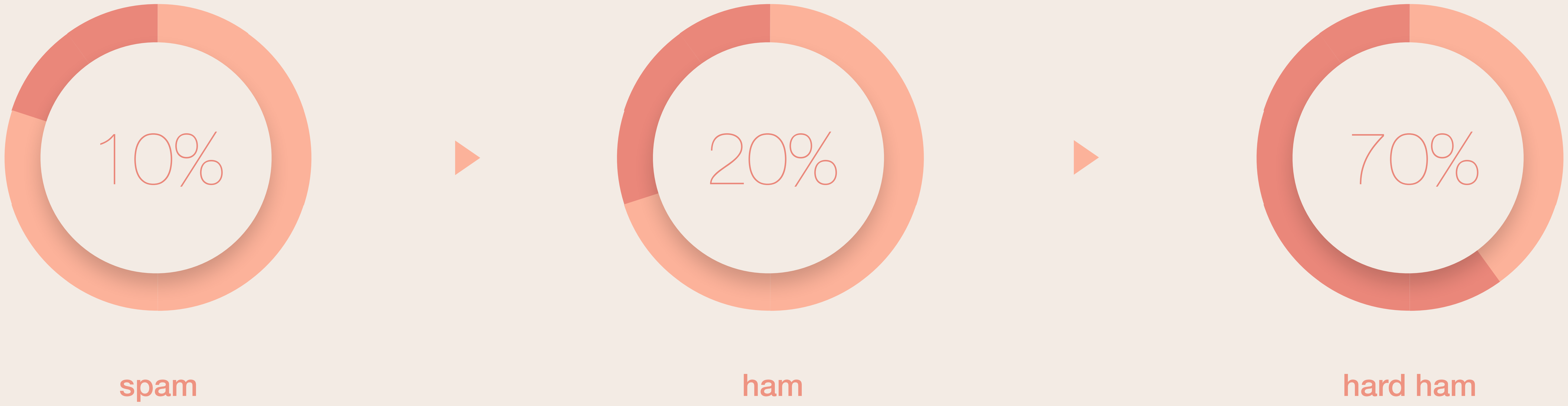
- **Precision:** % of selected items that are correct
  Recall: % of correct items that are selected

F1 Metric

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- The harmonic mean is a very conservative average;
- People usually use balanced F1 measure
  - i.e., with $\beta = 1$ (that is, $\alpha = ½$):
  - $F = 2PR/(P+R)$



Data Skew: Suppose 90% emails are ham and 10% are spam.
How about the training results?

Let us Implement Naive Bayes on Hadoop

We need three MapReduce Jobs for training and one for prediction.

Job1: Compute the Priors (the ratio of each class)
 *input: class id+document id + words
 *output: class id + size of class+number of words in document

Let us Implement Naive Bayes on Hadoop

We need three MapReduce Jobs for training and one for prediction.

Job2: similar to wordcount, compute the probability of a word in each class
*input: class id + document id + words
*output: class id + frequency of the word in the class + frequency of the word in all classes

Let us Implement Naive Bayes on Hadoop

We need three MapReduce Jobs for training and one for prediction.

Job3: Compute the Probability
 * input:class id + word + total number of words
 * output: word, “log probability for a class”
option: maintain the results in Hbase

Let us Implement Naive Bayes on Hadoop

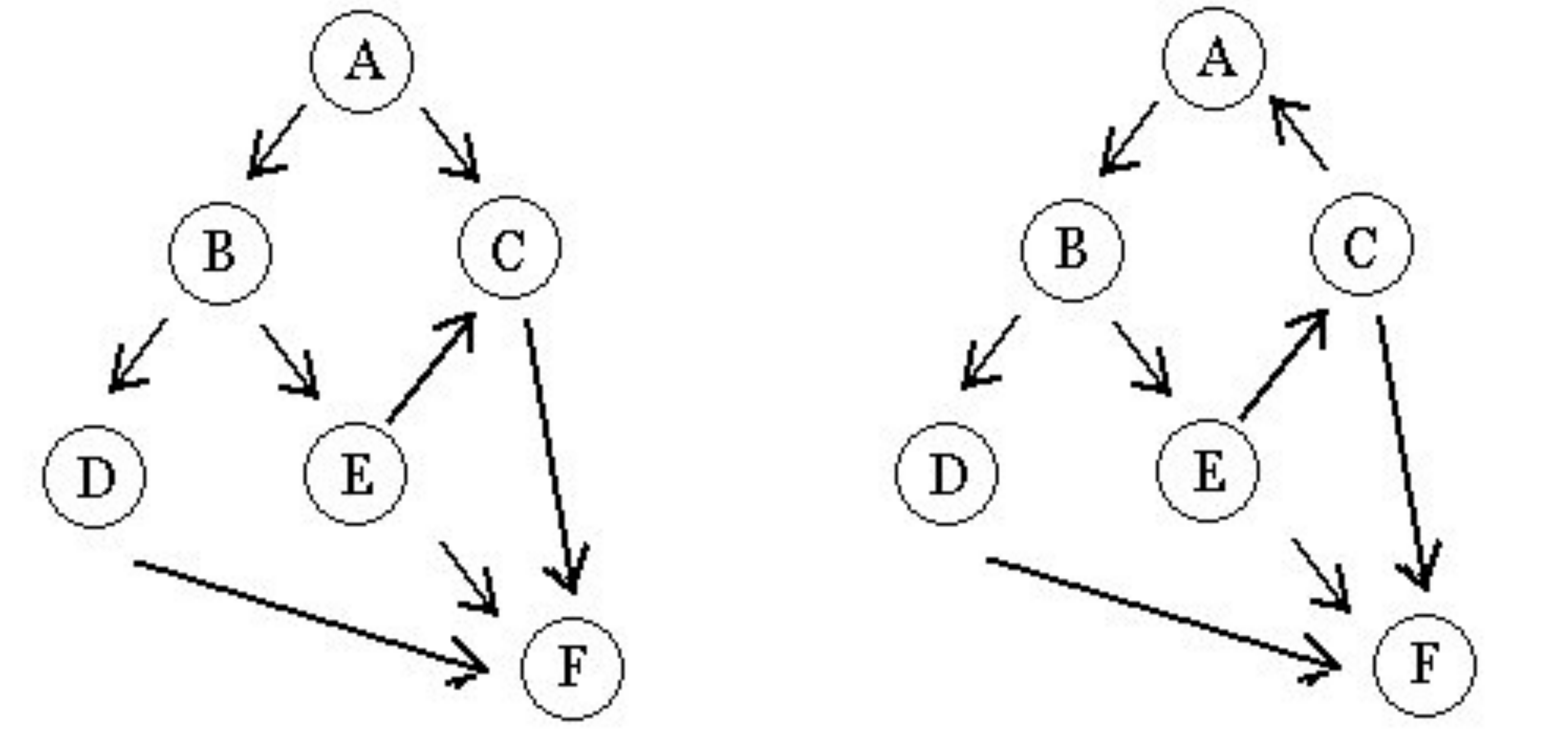
We need three MapReduce Jobs for training and one for prediction.

Job4: for a new document, predict its class id
*input: document id+words
*output: document id + class id

You can access HBase from hadoop, but it is very slow. Any solution?

Last, but not the least: The general Bayes Network

- A set of variables and a set of direct edges between variables
- Each variables has a finite set of mutually exclusive states
- The variable and direct edge form a DAG (directed acyclic graph)

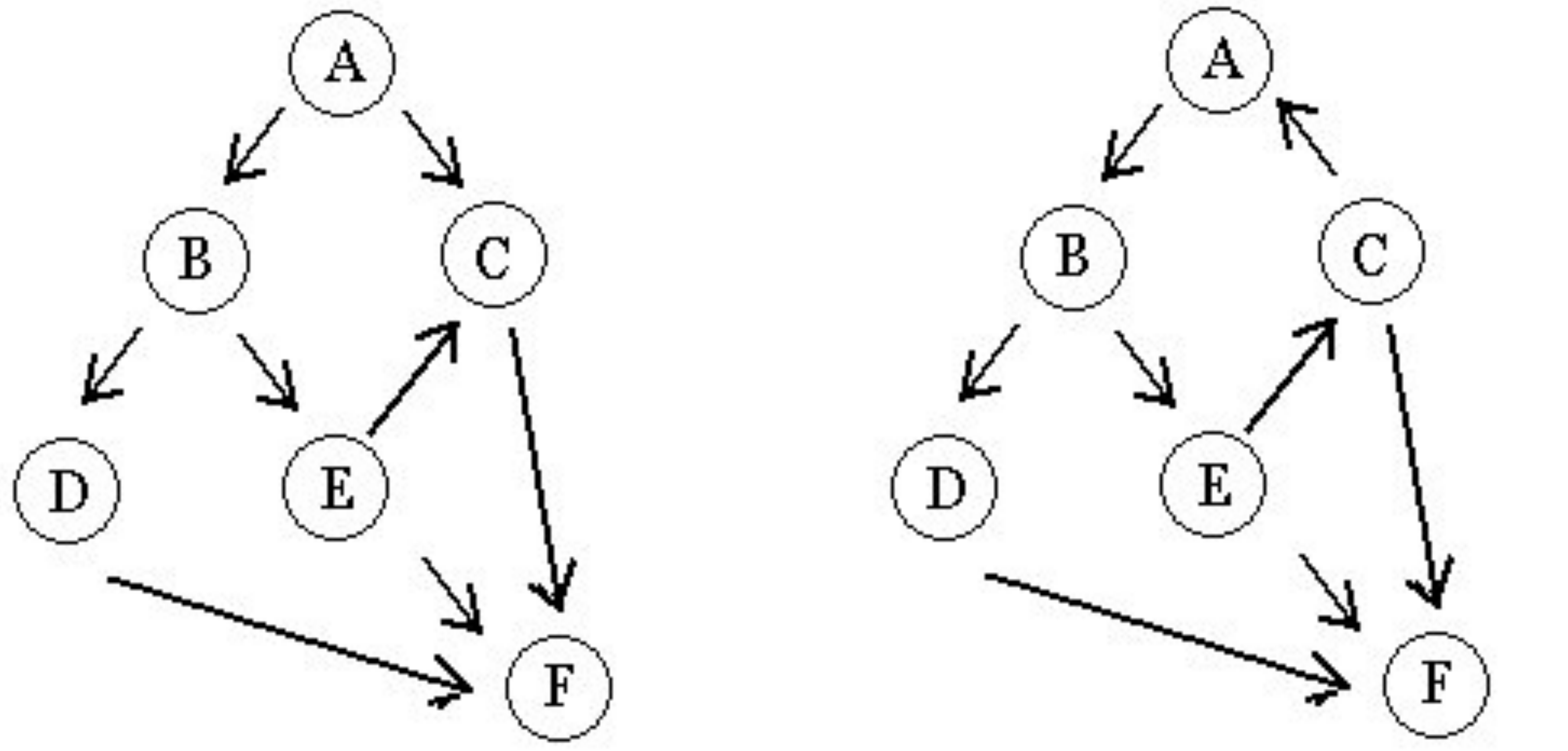


The graph on the left is a valid Bayesian netwrok. The probabilities to specifty are P(A), P(B|A), P(C|A,E),P(D|B), P(E|B) and P(F|C,D,E).
The one on the right is not a valid Bayesian network as the cycle ABEC exists.

Bayes Network

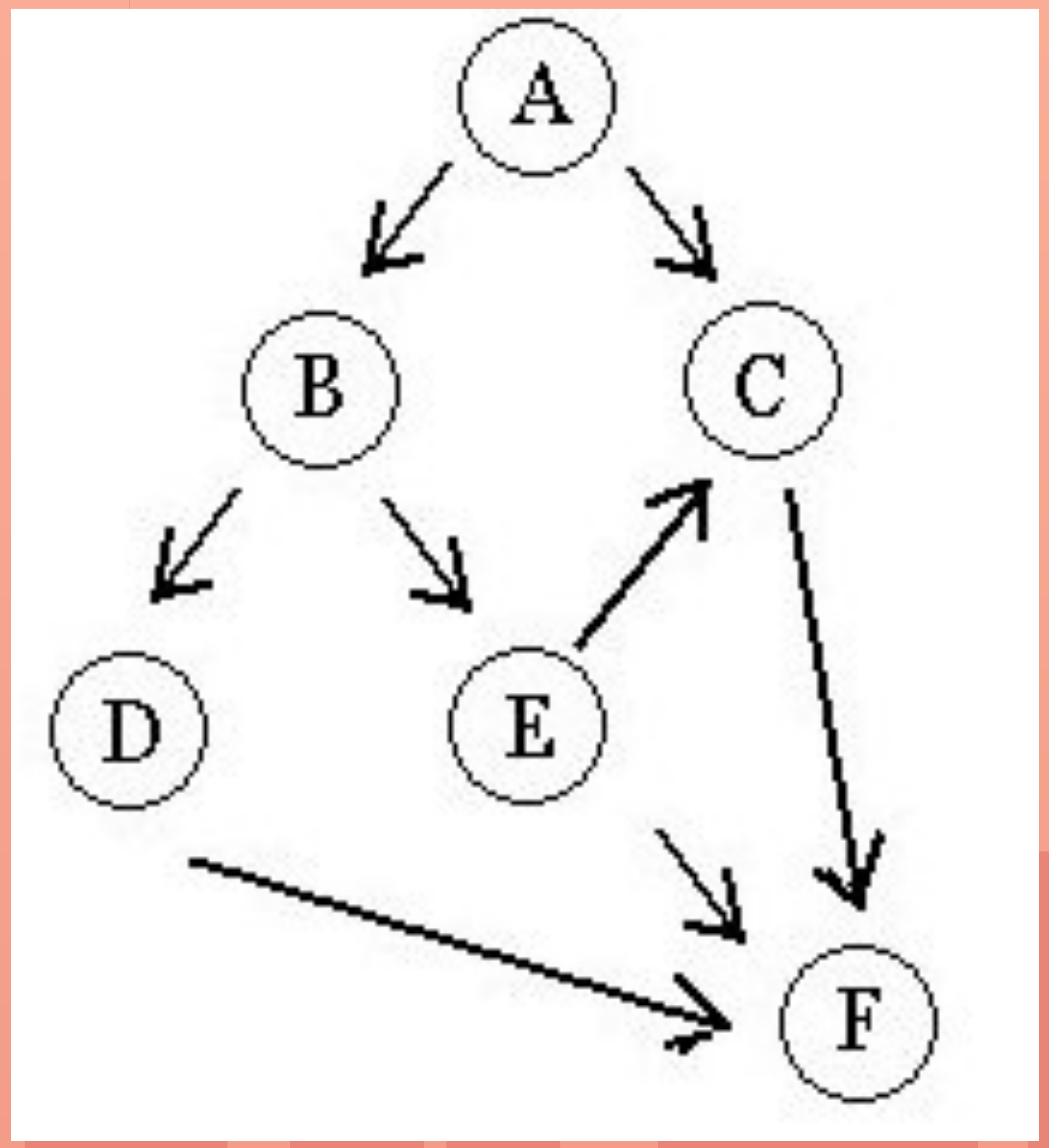
$$P(V_1, V_2, \ldots V_n) = \prod_{i=1}^{n} P(V_i \mid par(V_i))$$

$$\begin{aligned} P(A,B,C,D,E,F) &= P(F|C,D,E)P(A,B,C,D,E) \\ &= P(F|C,D,E)P(C|A,E)P(D|B)P(E|B)P(B,A) \\ &= P(F|C,D,E)P(C|A,E)P(D|B)P(E|B)P(B|A)P(A) \end{aligned}$$



The graph on the left is a valid Bayesian netwrok. The probabilities to specifty are P(A), P(B|A), P(C|A,E),P(D|B), P(E|B) and P(F|C,D,E).
The one on the right is not a valid Bayesian network as the cycle ABEC exists.

Inference in Bayesian Networks



$$P(c|e) = P(c,a|e) + P(c,\sim a|e)$$

$$\mathbf{P(A|B,C)\ P(B|C) = P(A,B|C)}$$

$$P(c|e) = P(c|a,e)P(a|e) + P(c|\sim a,e)P(\sim a|e)$$

A has no parents, therefore $p(a|e) = p(a)$ and $p(\sim a|e) = p(\sim a)$

$$P(c|e) = P(c|a,e)P(a) + P(c|\sim a,e)P(\sim a)$$

Project 1: Email Spam Detection

Given the Enron Email dataset, please build a model (Naive Bayes) on top of Hadoop to classifying the emails into hams and spams.

Your online model is required to be able to do the classification in real-time.

Please report the recall/precision and demonstrate face to face.