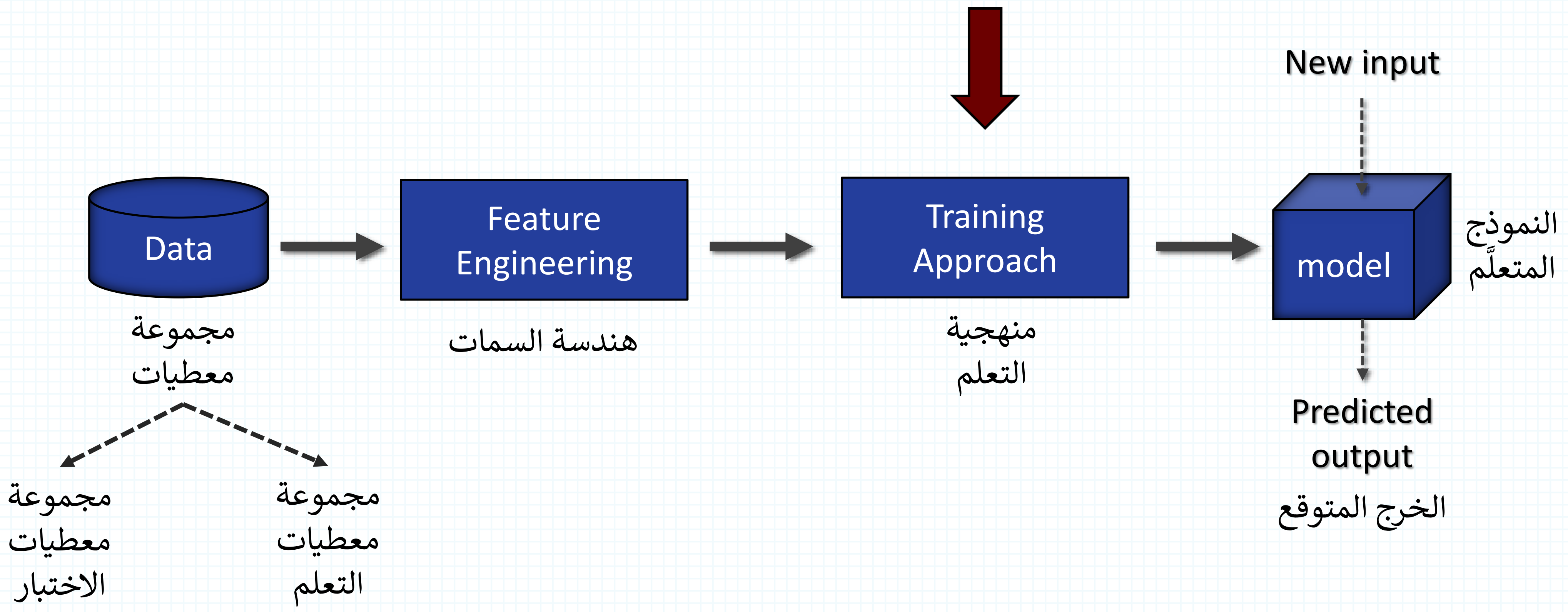
الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

مقرر تعلم الآلة | كلية الهندسة المعلوماتية | المحاضرة الثامنة

Support Vector Machine (SVM) 1

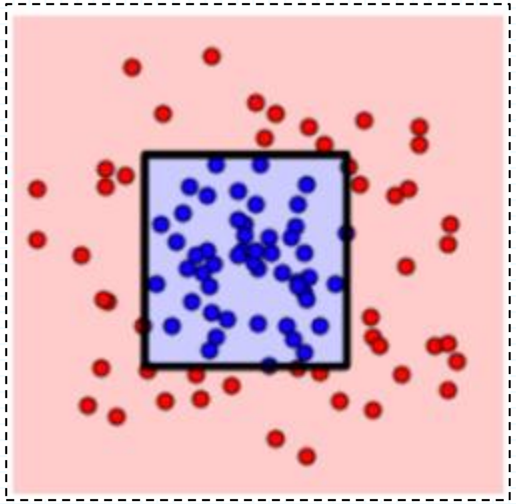
د. رياض سنبل

ML Pipeline


Data
مجموعة معطيات
مجموعة معطيات الاختبار
مجموعة معطيات التعلم
Feature Engineering
هندسة السمات
Training Approach
منهجية التعلم
New input
model
النموذج المتعلَّم
Predicted output
الخرج المتوقع

Why SVM? … Why not decision trees?

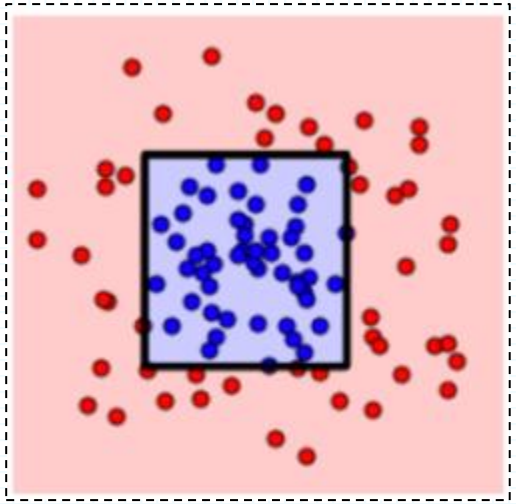
- Can Decision Trees detect non-linear models?
- What type of boundaries can be detected using decision trees in each step?



- What if we have higher-dimensional feature space, more complex relationships between input features and target class?

Why SVM? … Why not decision trees?

- Can Decision Trees detect non-linear models?
  - Yes, Decision trees can detect non-linear relationships
- What type of boundaries can be detected using decision trees in each step?
  - The decision boundary in a Decision Tree is linear and perpendicular to one of the input dimensions, which means that it is limited to finding only axis-parallel splits.



- What if we have higher-dimensional feature space, more complex relationships between input features and target class?
  - In the higher-dimensional feature space, the decision boundary can take on a more complex shape, such as a curved or nonlinear boundary.
  - More problems when the relationship between the input features and the target variable is complex (ex: image classification, sentiment analysis, etc)

So .. what is SVM?

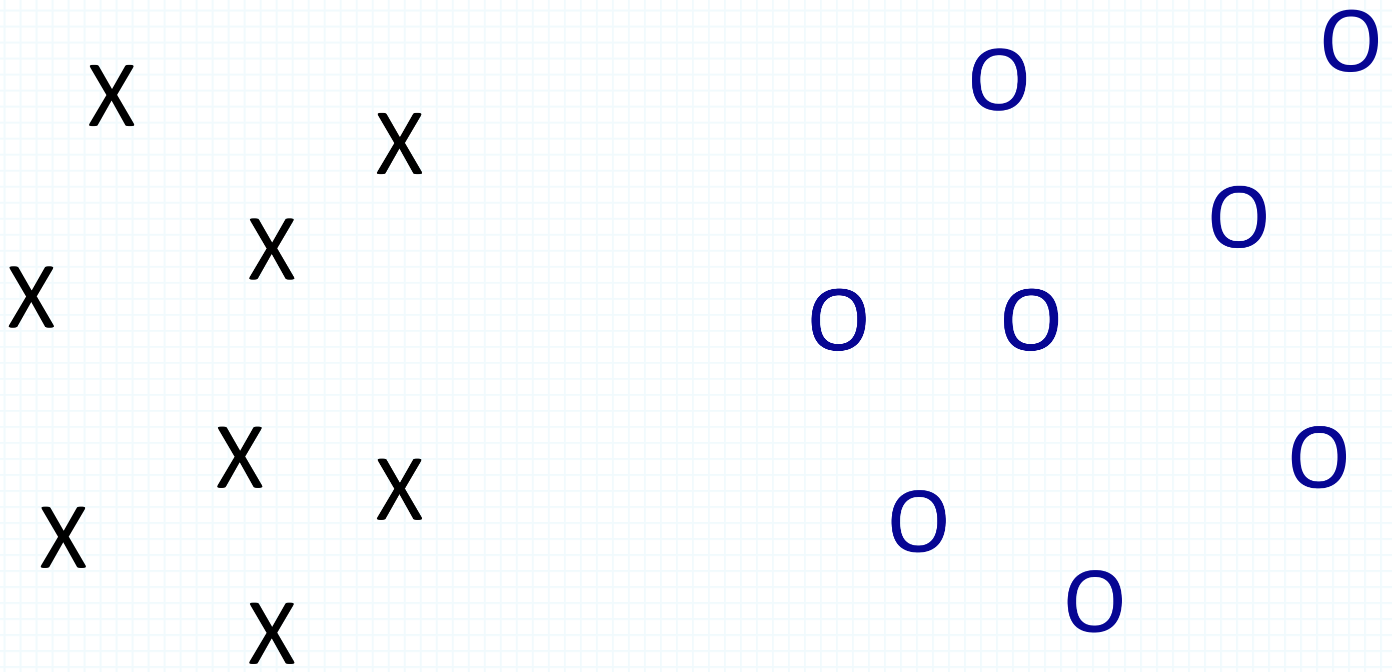
- The short answer!

```python
import pandas as pd
import numpy as np
from sklearn.metrics import classification_report
from sklearn.datasets import load_breast_cancer
from sklearn.svm import SVC
```

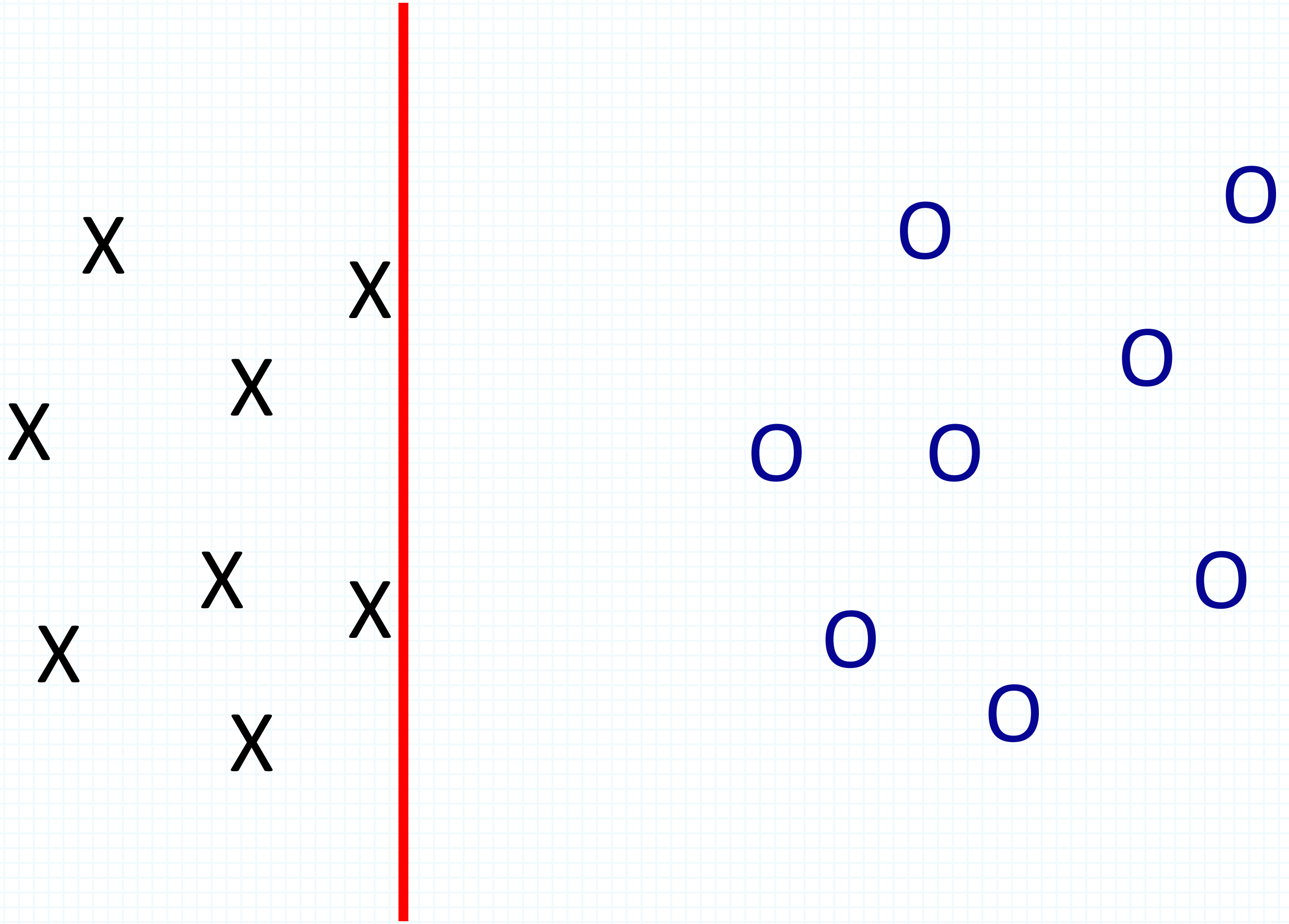
```python
# train the model on train set
model = SVC()
model.fit(X_train, y_train)

# print prediction results
predictions = model.predict(X_test)
print(classification_report(y_test, predictions))
```

```python
from sklearn.model_selection import GridSearchCV

# defining parameter range
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}

grid = GridSearchCV(SVC(), param_grid, refit = True, verbose = 3

# fitting the model for grid search
grid.fit(X_train, y_train)
```

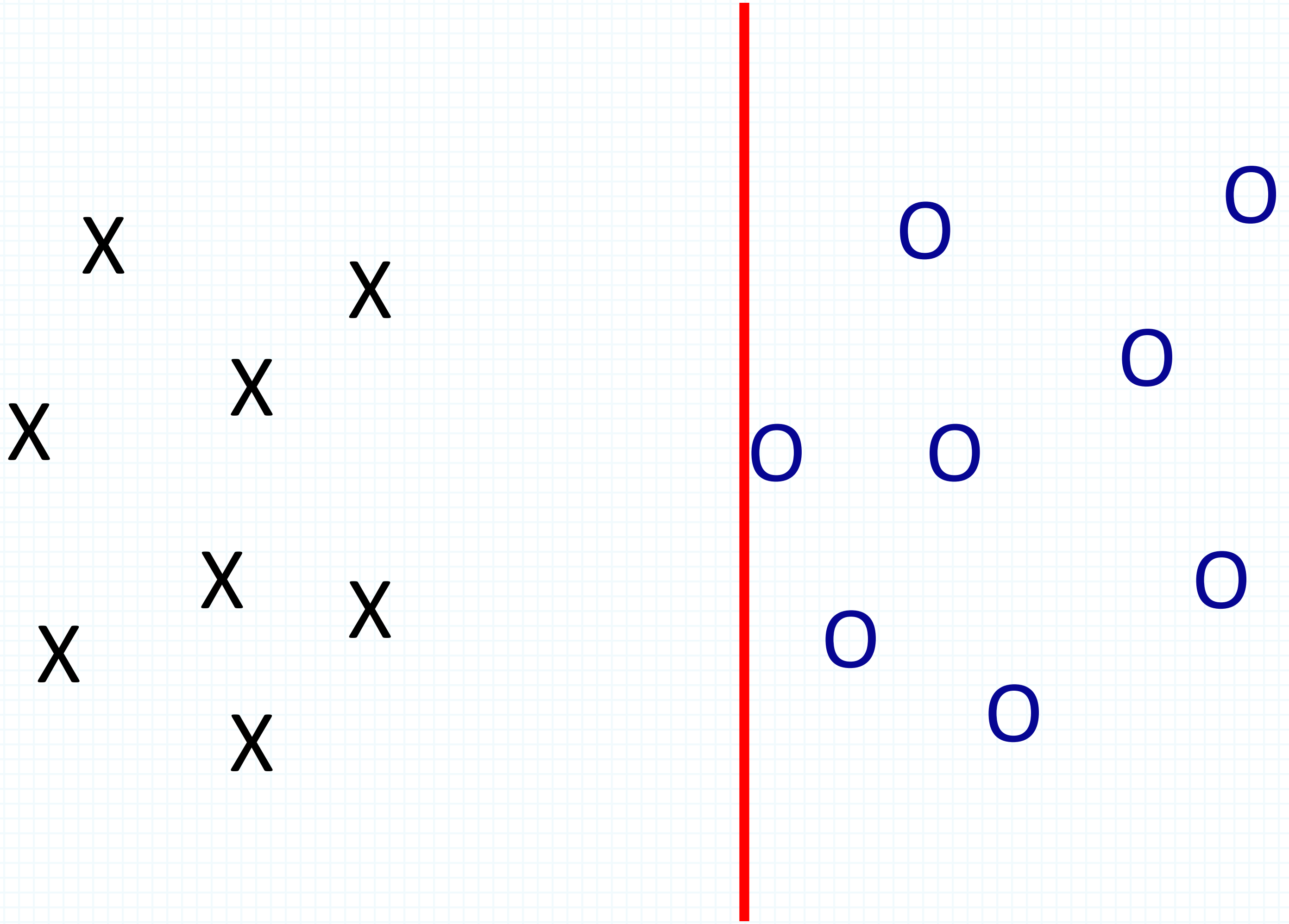
Intuitions



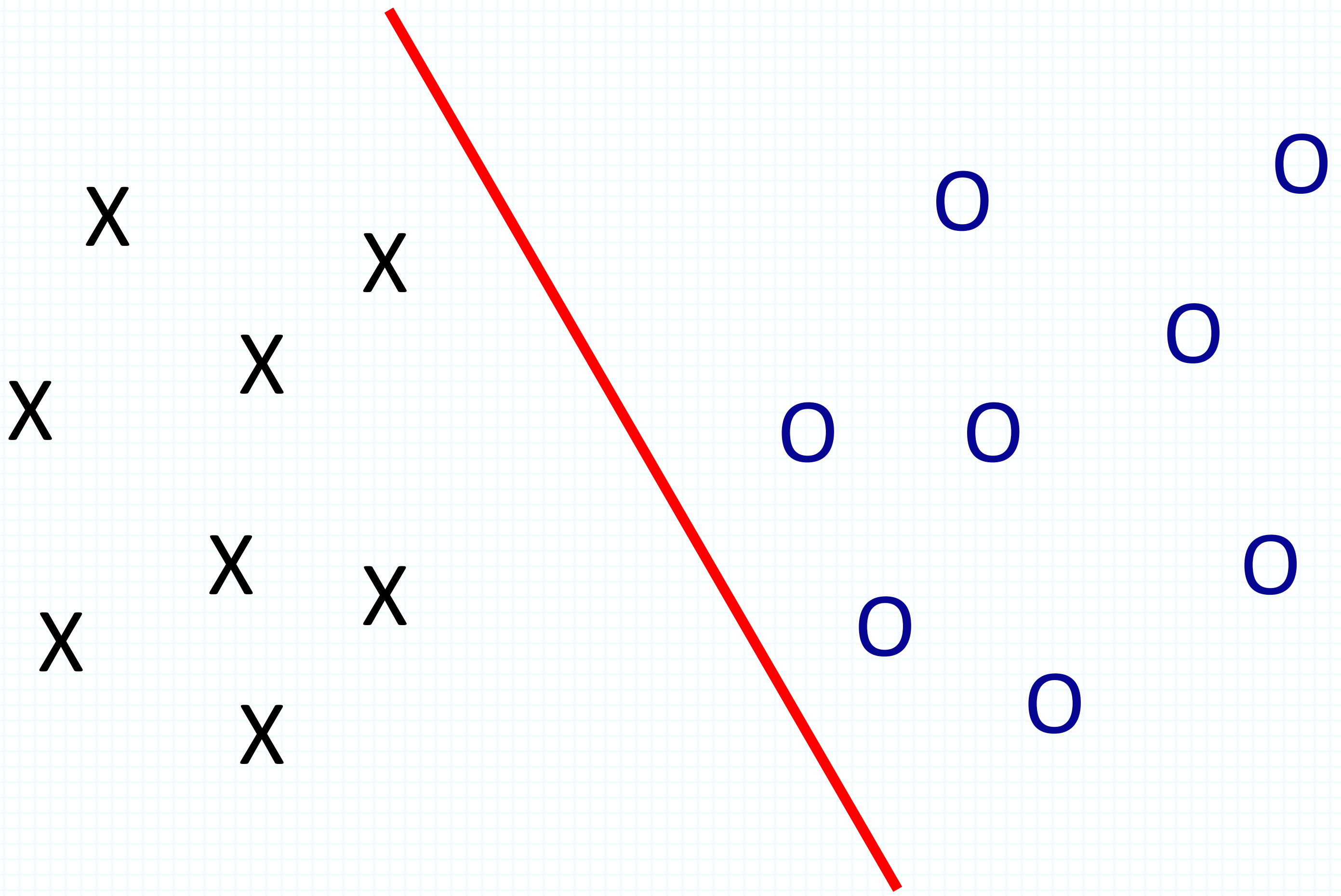
Intuitions



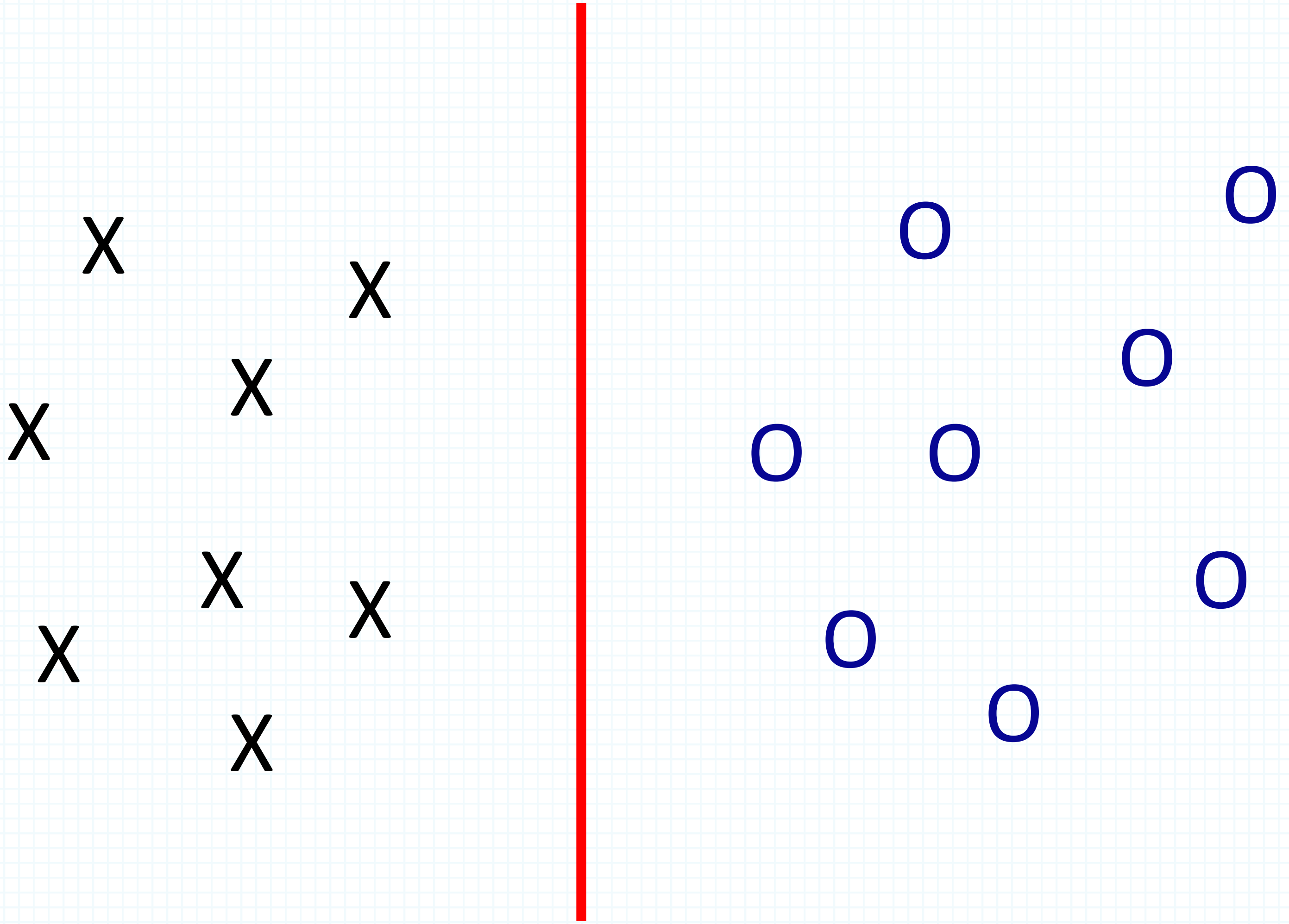
Intuitions



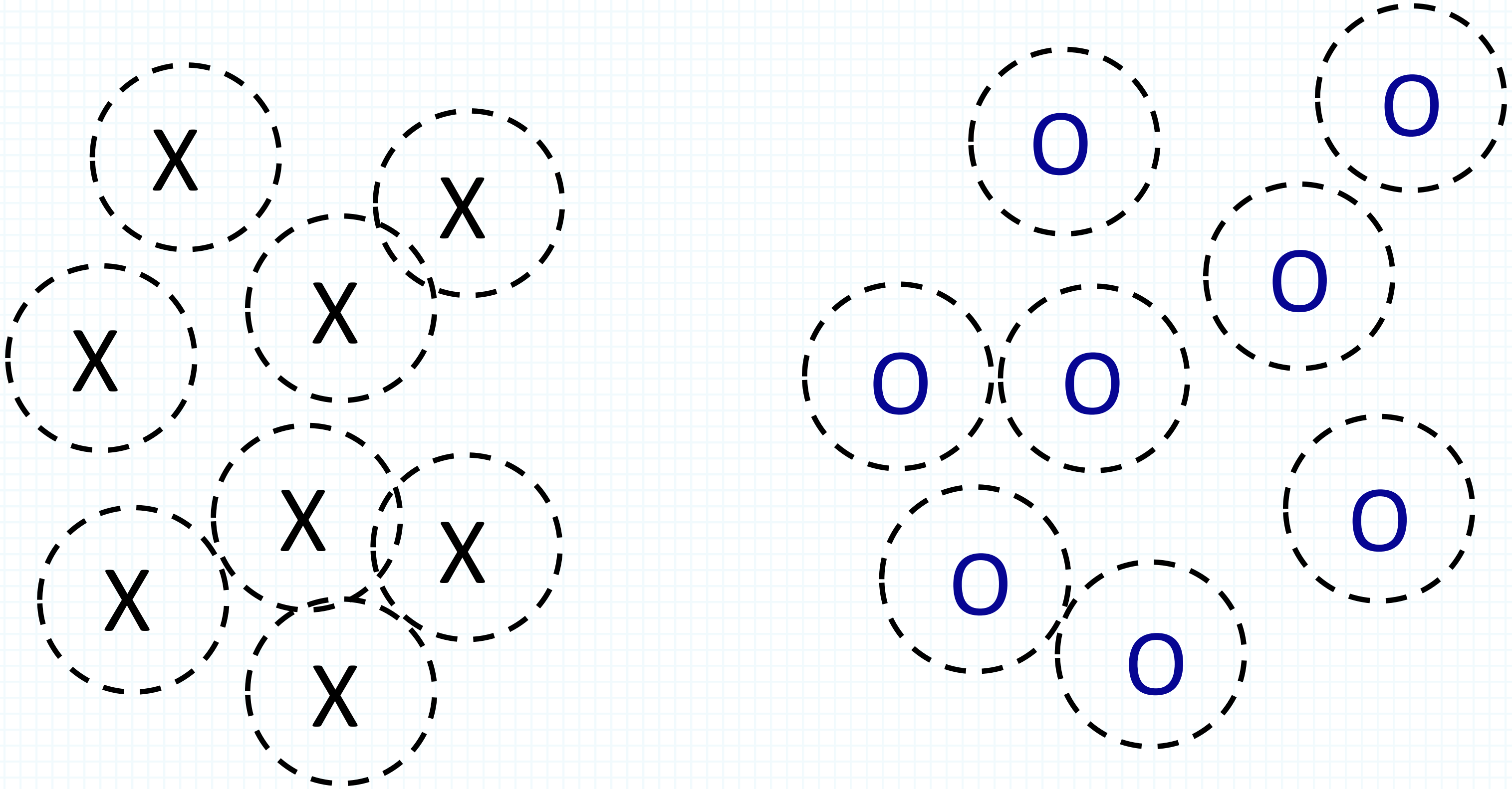
Intuitions



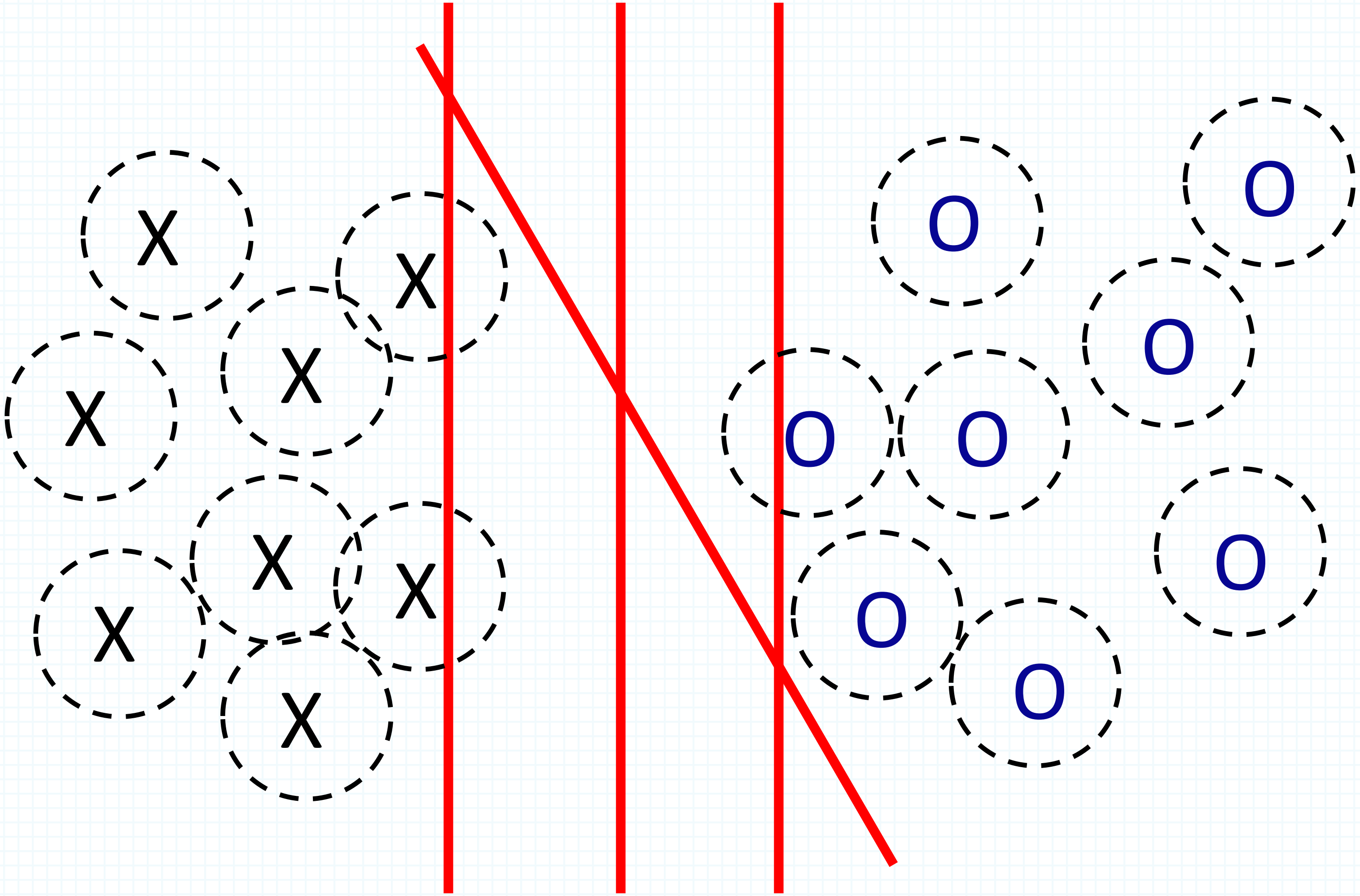
A "Good" Separator



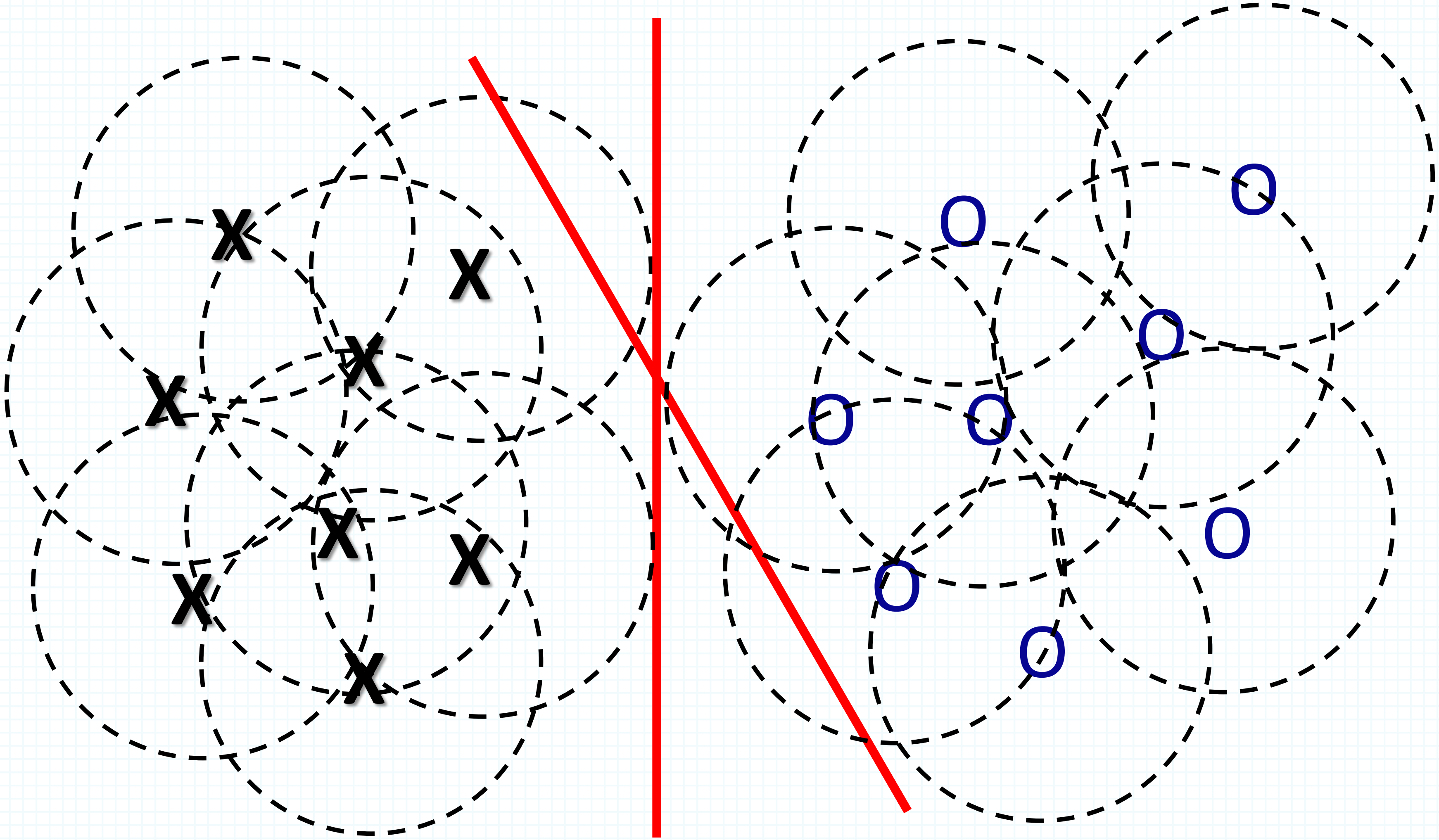
Noise in the Observations



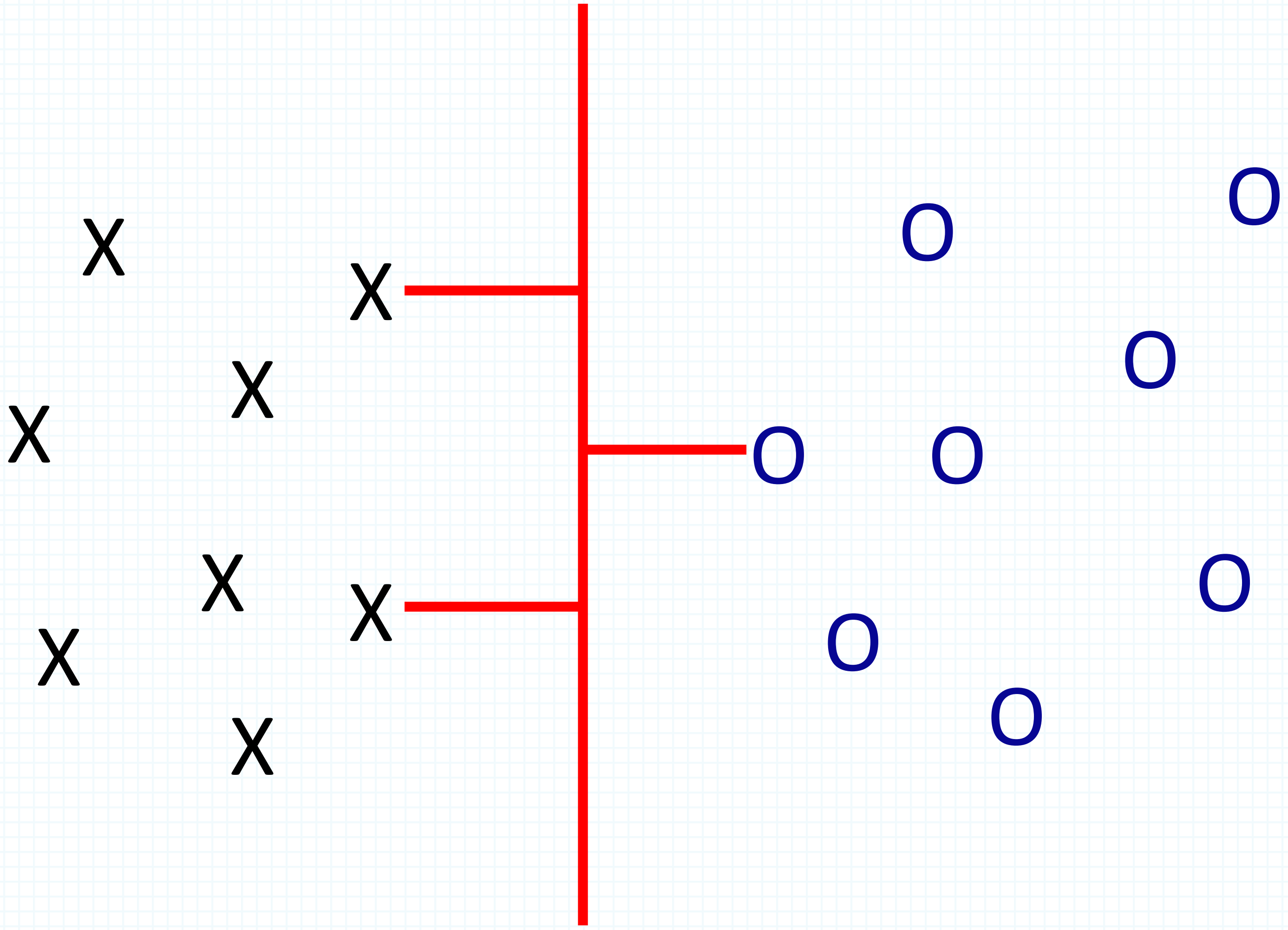
Ruling Out Some Separators



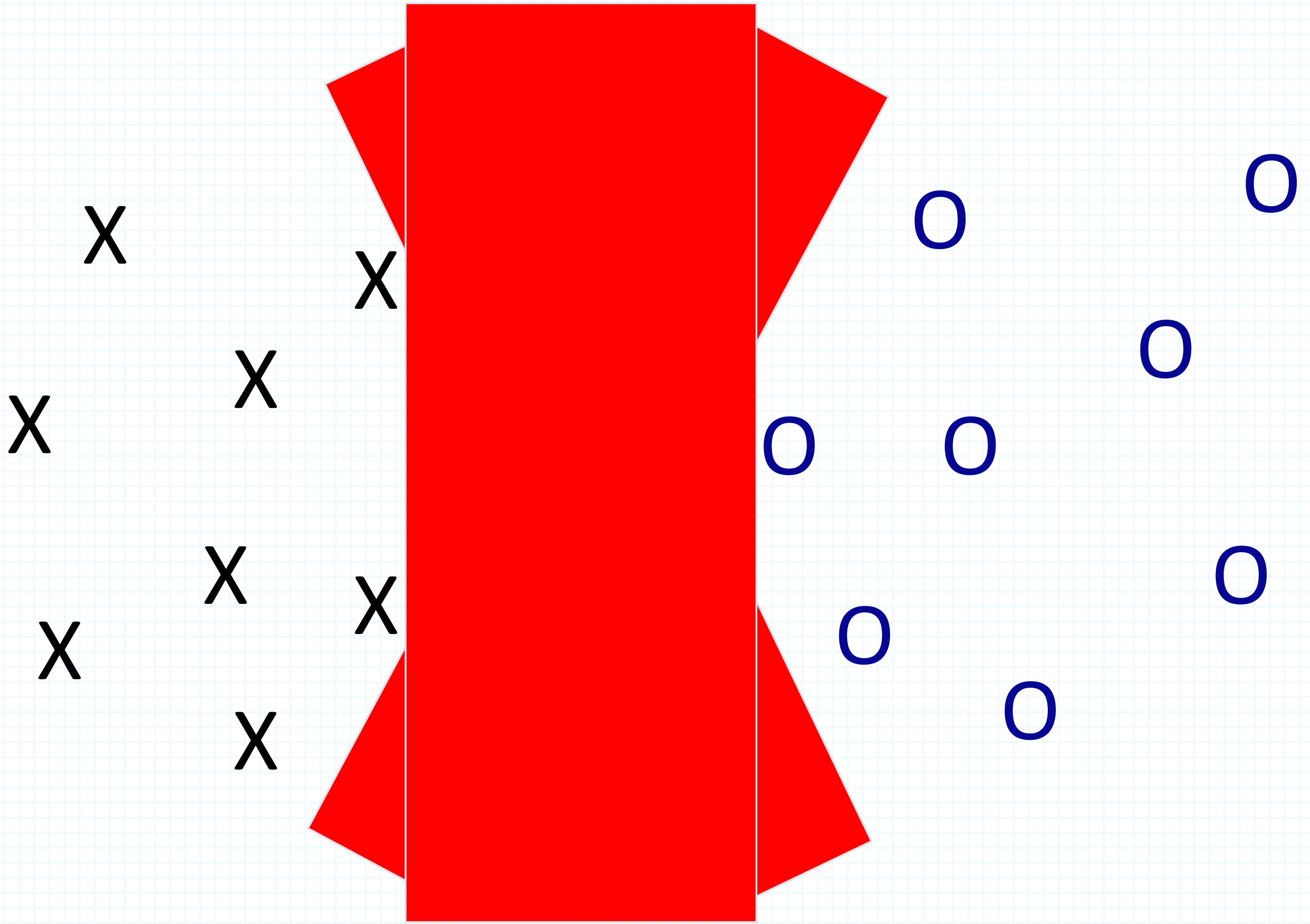
Lots of Noise



Maximizing the Margin

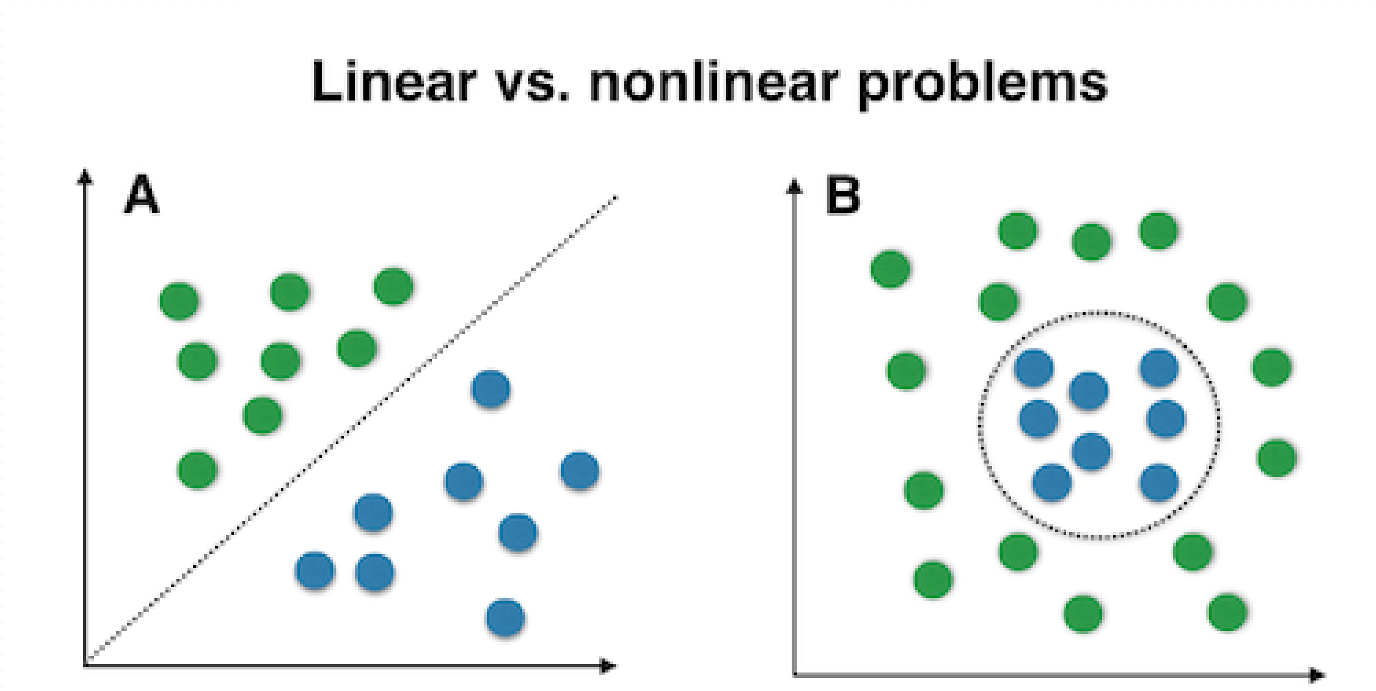


“Fat” Separators



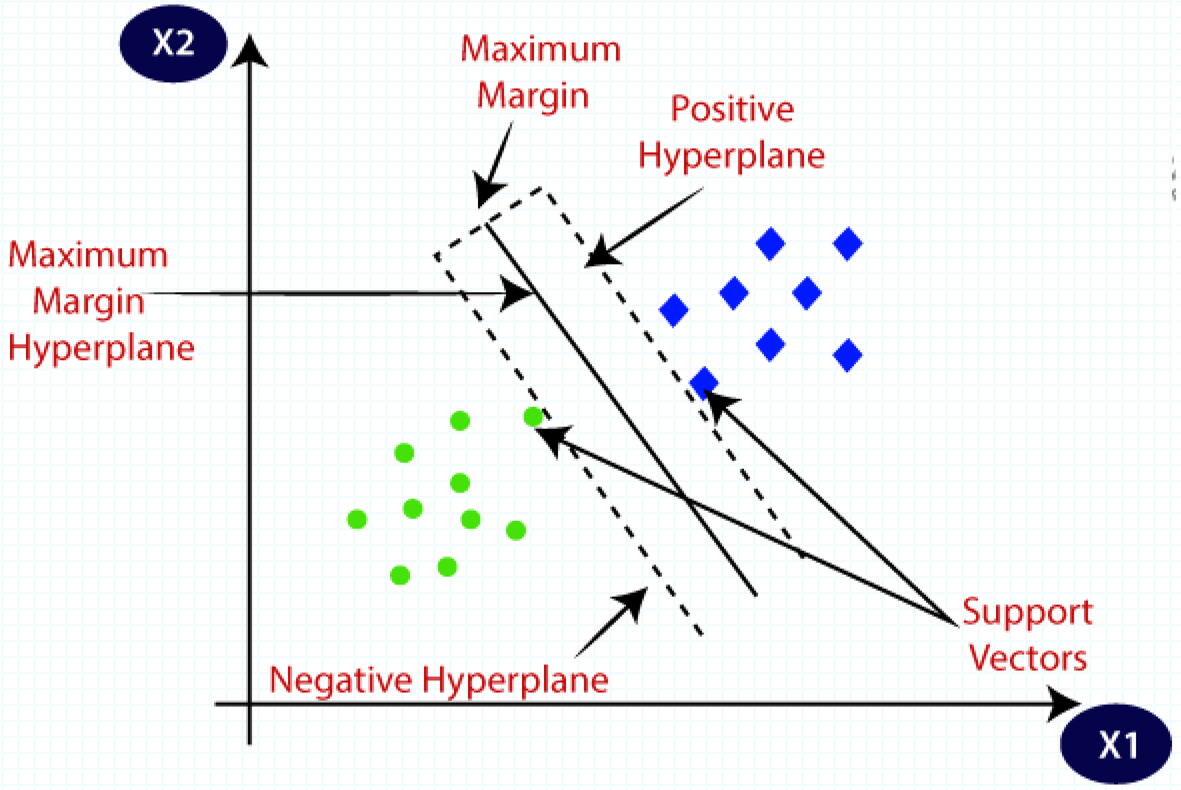
What is a Support Vector Machine?

- It is a supervised machine learning problem where we try to find a hyperplane that best separates the two classes.



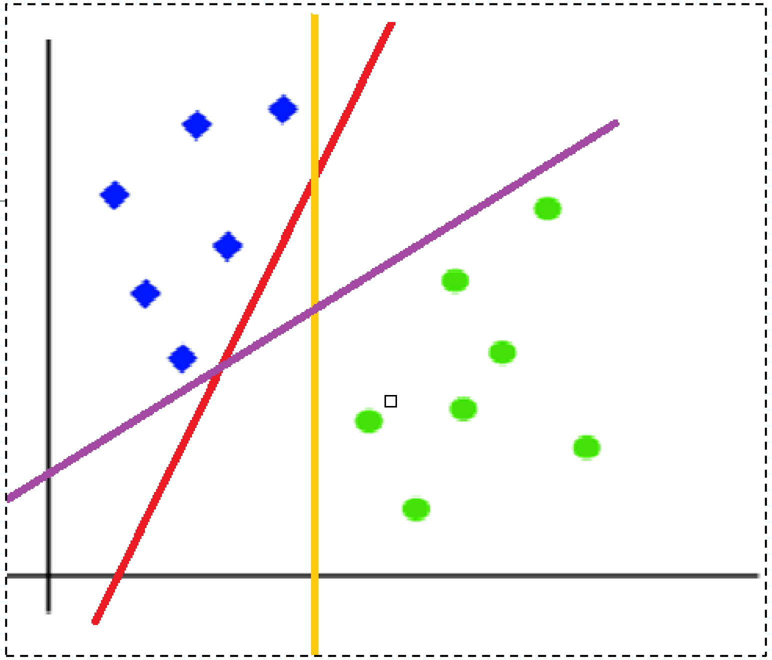
Terms

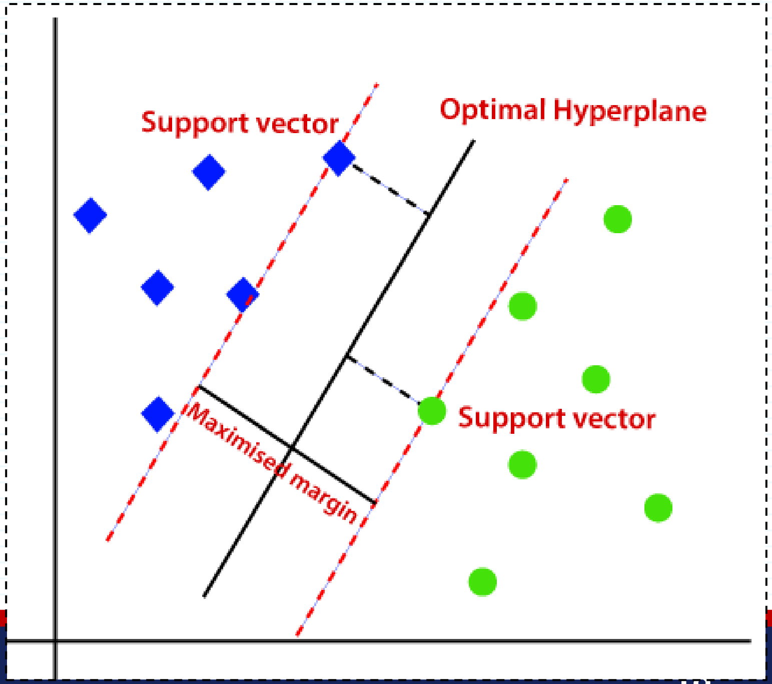
- **Support Vectors:**
  - These are the points that are closest to the hyperplane.
  - A separating line will be defined with the help of these data points.
- **Margin:**
  - It is the distance between the hyperplane and the observations closest to the hyperplane (support vectors).
  - In SVM large margin is considered a good margin.
  - There are two types of margins **hard margin** and **soft margin.**



How does SVM work?

- SVM is defined such that it is defined in terms of the support vectors only.
  - The margin is made using the points which are closest to the hyperplane (support vectors).
  - We don’t have to worry about other observations
  - Hence SVM enjoys some natural speed-ups!
- The best hyperplane is that plane that has the maximum distance from both the classes.



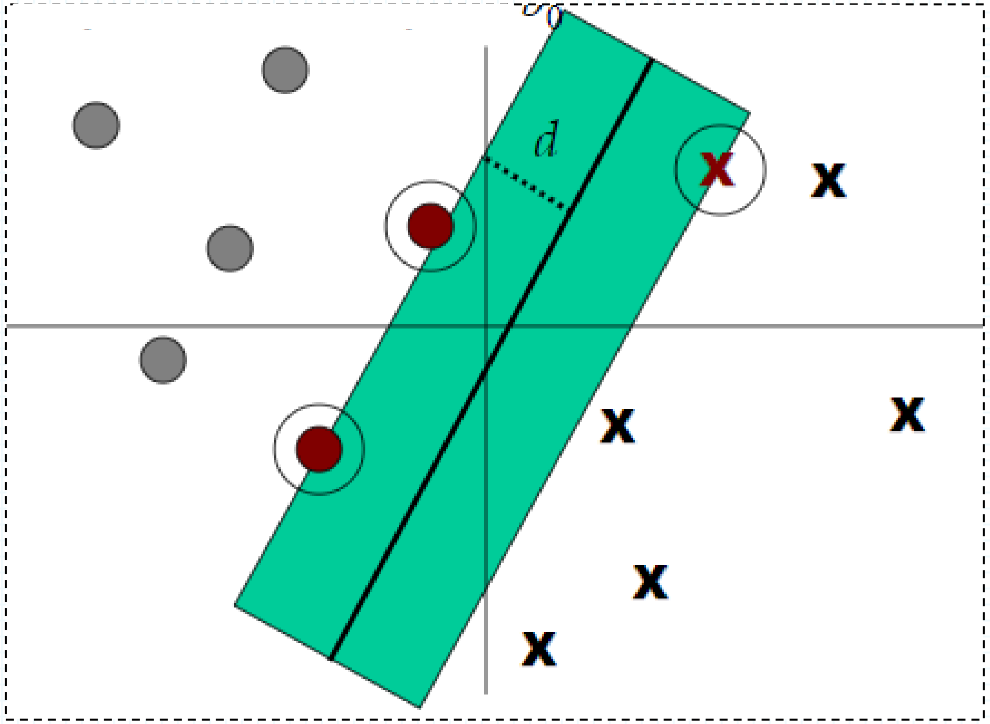


So.. What is our optimization problem?

- **Our problem:**

Maximizing the shortest distance to the closest positive or negative point.

$$w^* = \arg_w \max \left[\min_n d_H(\phi(x_n))\right]$$



Note that W represents all parameters i.e. w and b