Supervised Machine Learning - B

Road Map

- Basic concepts
- Regression
- Naïve Bayesian classification
- **K-nearest neighbor**
- Support vector machines
- Decision tree induction
- Ensemble methods: Bagging and Boosting
- Summary

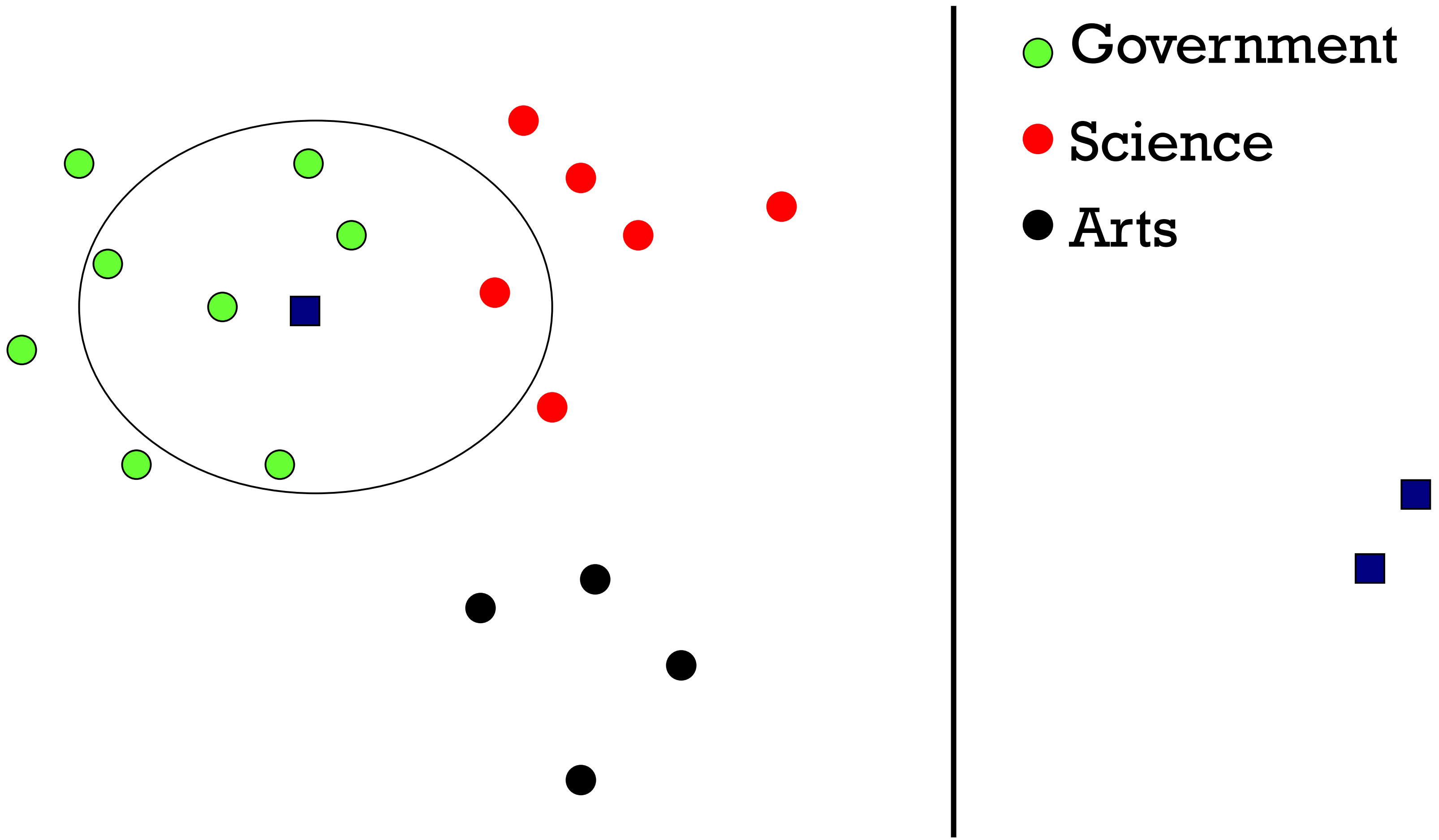
k-Nearest Neighbor Classification (kNN)

- To classify a test instance *d*, define *k*-neighborhood *P* as *k* nearest neighbors of *d*
- Count number *n* of training instances in *P* that belong to class c_j
- Estimate $\Pr(c_j|d)$ as n/k
- Classification time is linear in training set size for each test case.

k-Nearest Neighbor Classification (kNN)

- *k* is usually chosen empirically via a validation set or cross-validation by trying a range of *k* values.
- Distance function is crucial, but depends on applications.

Example: k=6 (6NN)



Discussions

- kNN can deal with complex and arbitrary decision boundaries.
- Despite its simplicity, researchers have shown that the classification accuracy of kNN can be quite strong and in many cases as accurate as those elaborated methods.
- kNN is slow at the classification time
- kNN does not produce an understandable model

Road Map

- Basic concepts
- Regression
- Naïve Bayesian classification
- K-nearest neighbor
- **Support vector machines**
- Decision tree induction
- Ensemble methods: Bagging and Boosting
- Summary

Support vector machines - Introduction

- Support vector machines were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992.
- SVMs are linear classifiers that find a hyperplane to separate two class of data, positive and negative.
- Kernel functions are used for nonlinear separation.
- SVM not only has a rigorous theoretical foundation, but also performs classification more accurately than most other methods in applications, especially for high dimensional data.
- It is perhaps the best classifier for text classification.

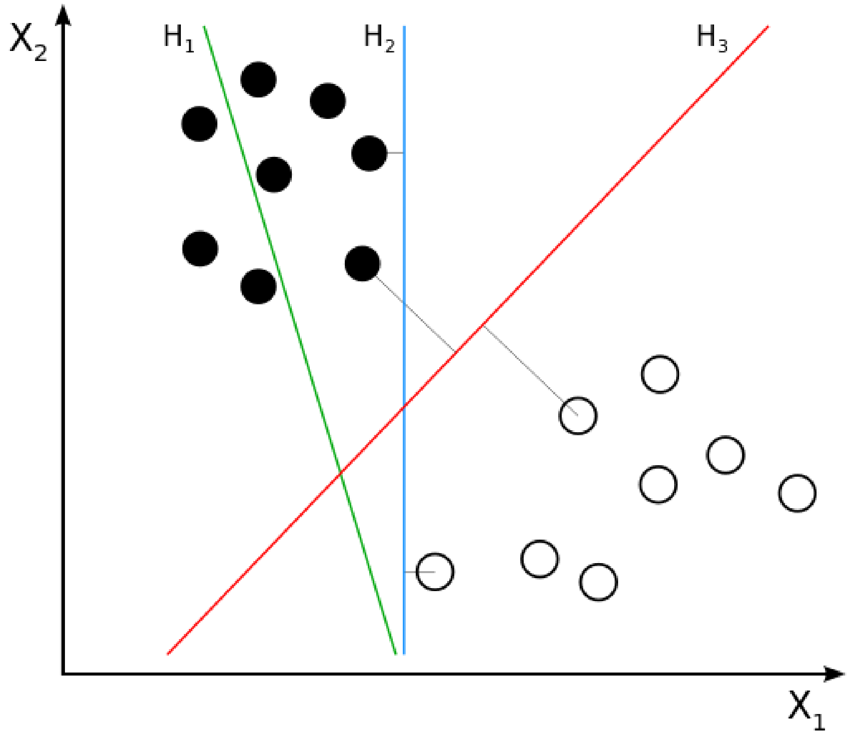
Support vector machines - Introduction

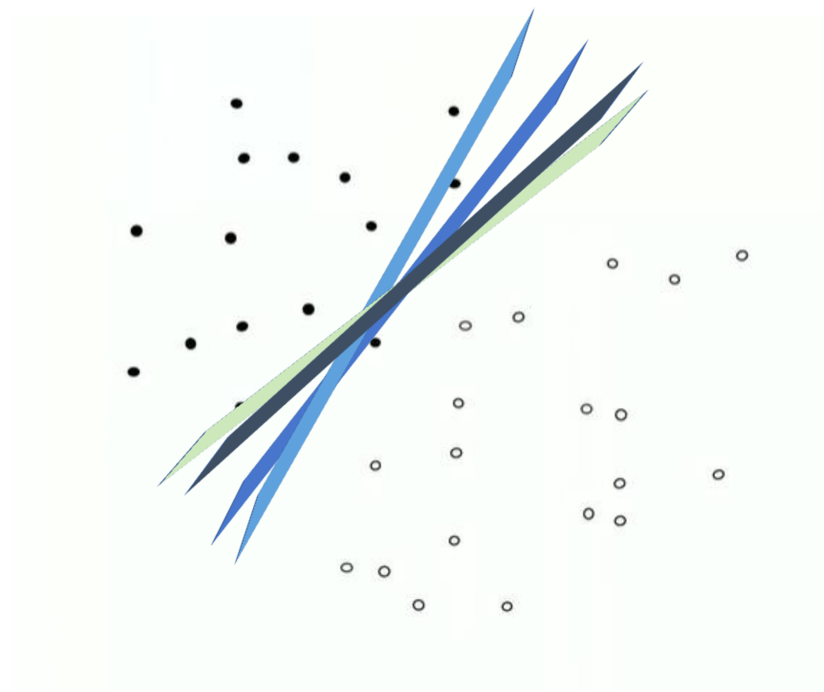
- Robust algorithm for classification problems
  - Medical problems
  - Text and hypertext categorization
  - Image classification

- Advantages:
  - SVM depends on optimization
  - Easy to use
  - Excellent performance on different type of datasets

The hyperplane

- The hyperplane that separates positive and negative training data
- It is also called the decision boundary (surface).
- So many possible hyperplanes, which one to choose?



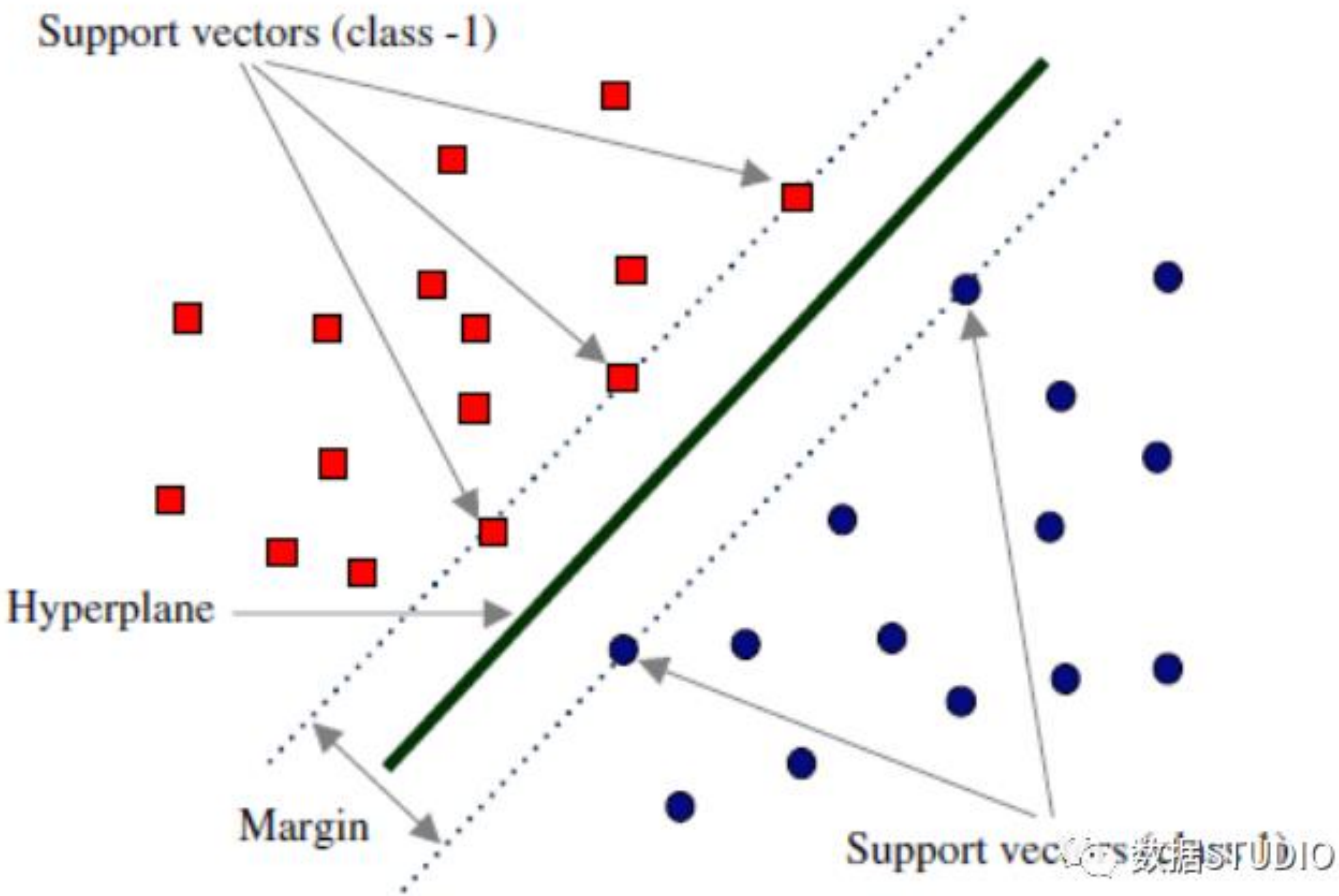


1. We can have different hyperplane.

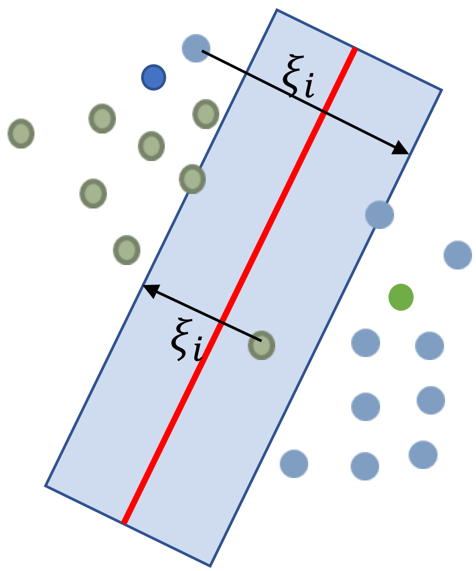
2. How do we choose a classifier?

What we need is a evaluation metrics to determine the a good classifier.

In SVM, we evaluate the margin in which we want to maximize the margin between the points and the hyperplane.


Support vectors (class -1)
Hyperplane
Margin

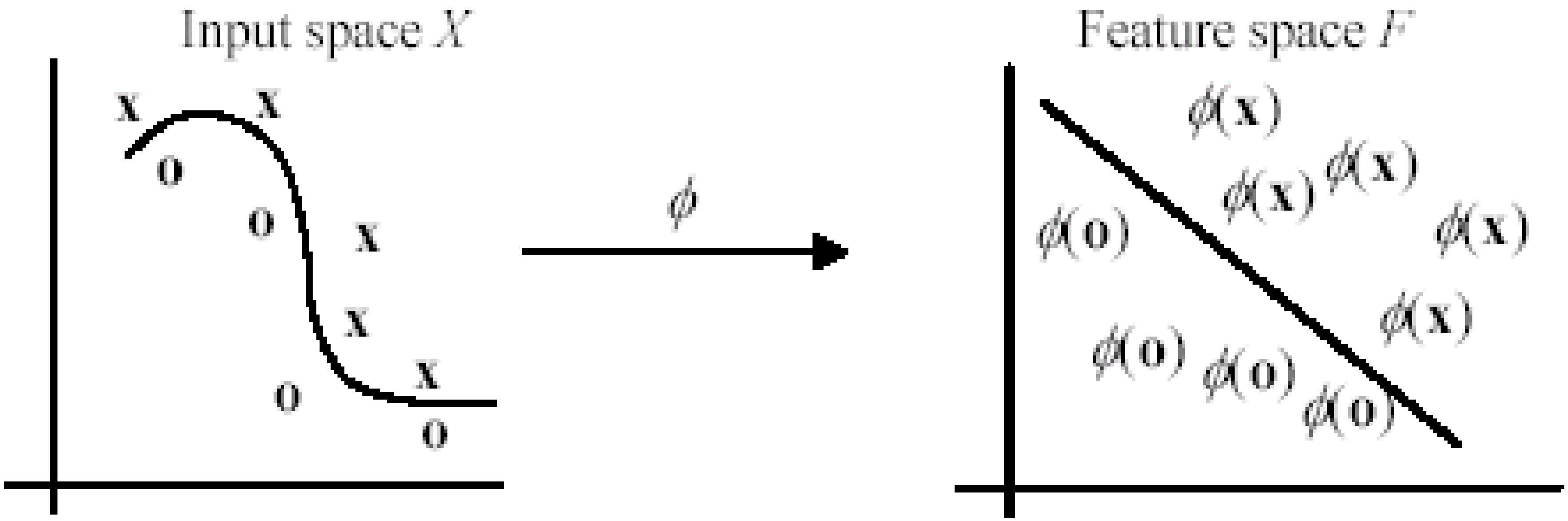
Soft Margin → Give some slack, allow some error



How to deal with nonlinear separation?

- The SVM formulations require linear separation.
- Real-life data sets may need nonlinear separation.
- To deal with nonlinear separation, the same formulation and techniques as for the linear case are still used.
- We only transform the input data into another space (usually of a much higher dimension) so that
  - a linear decision boundary can separate positive and negative examples in the transformed space
- The transformed space is called the **feature space**. The original data space is called the **input space**.

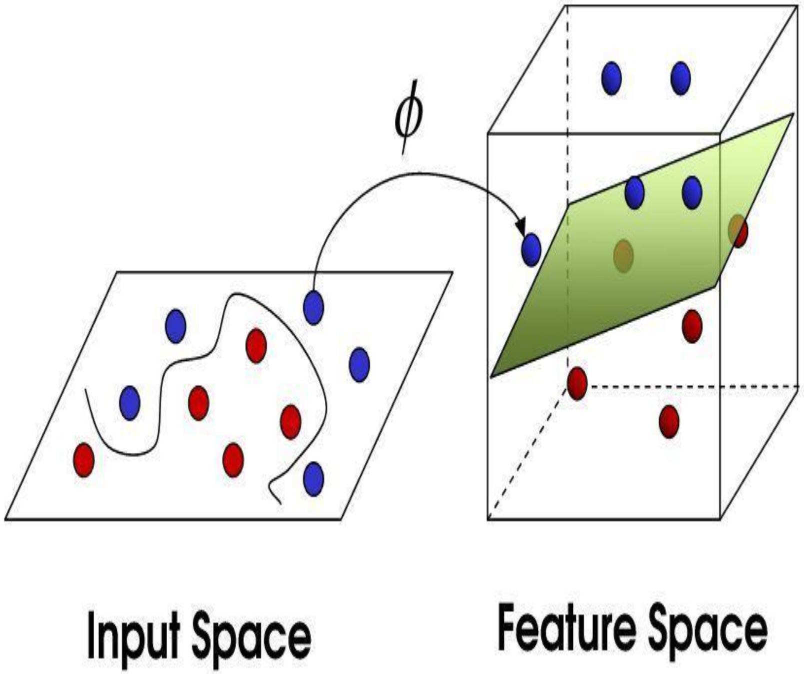
Geometric interpretation



- In this example, the transformed space is also 2-D. But usually, the number of dimensions in the feature space is much higher than that in the input space

Kernel methods

- There are many kernel methods used in the machine learning world.
- Most suited for small to medium datasets
- Kernel methods use kernels to map the input data into a different space.
- Then, simple models can be trained on the new feature space, which can result in an increase in the performance of the models.



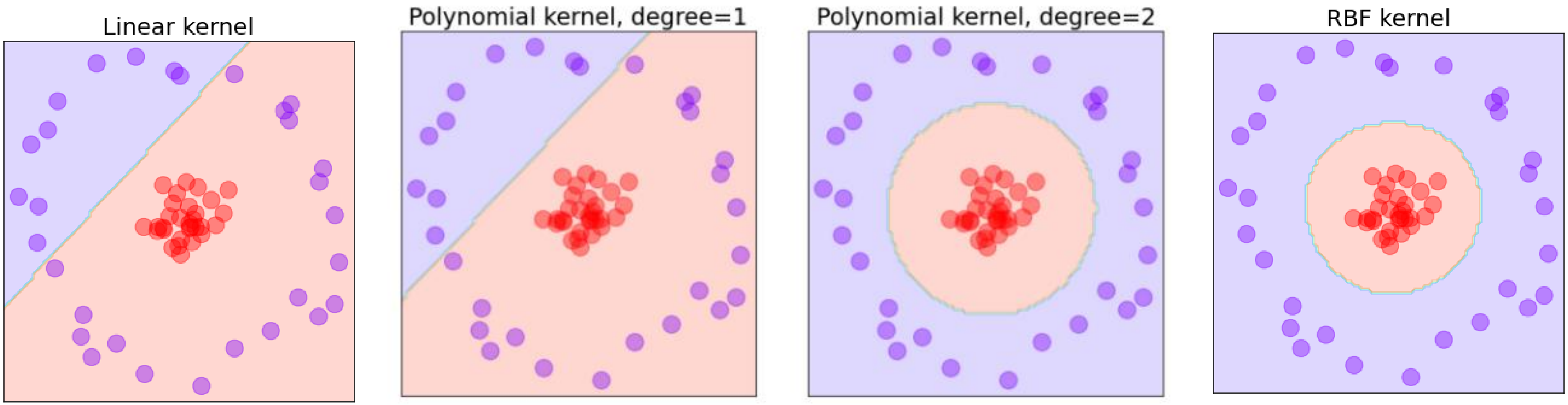
Problem with explicit transformation

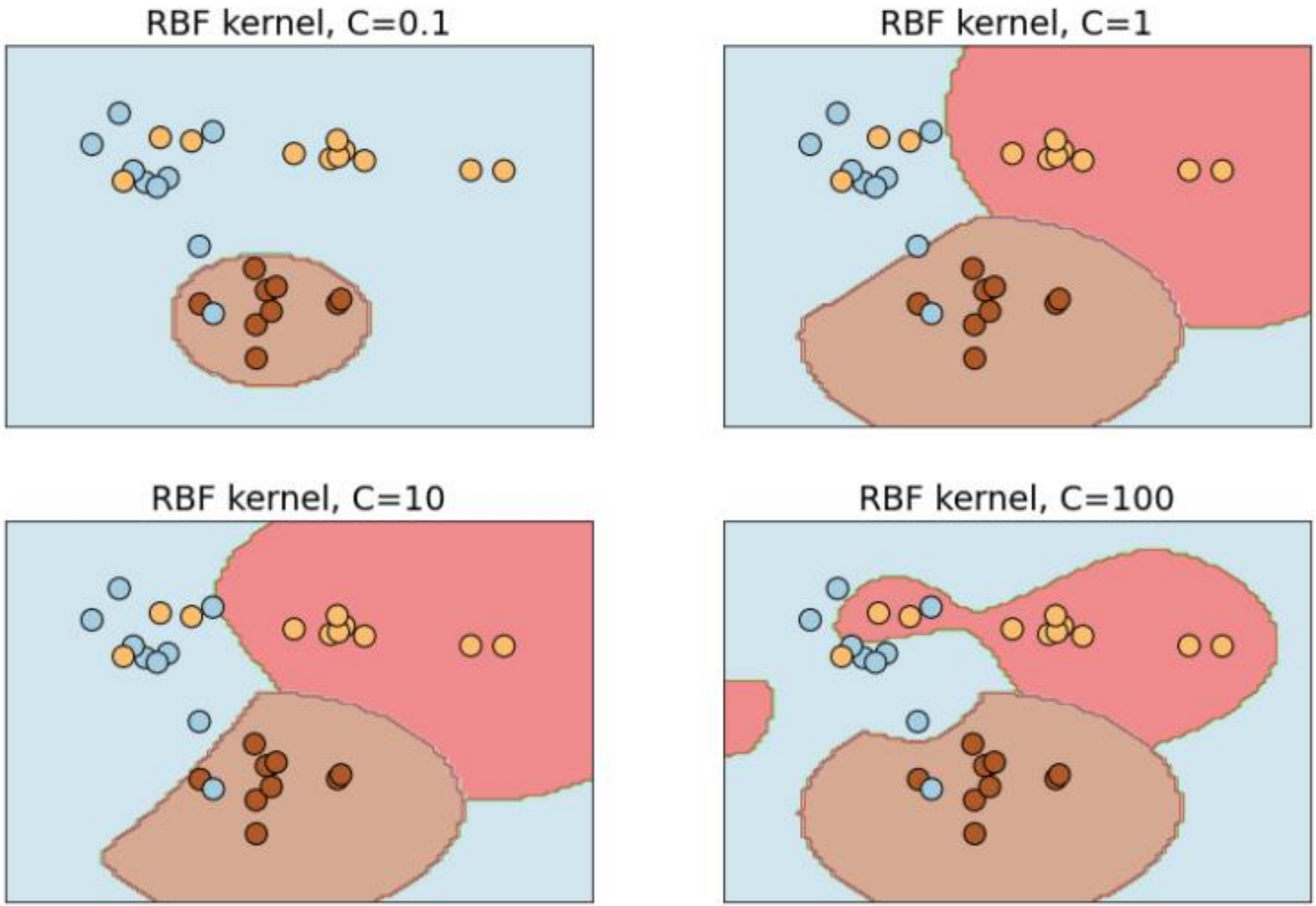
- The potential problem with the explicit data transformation and then applying the linear SVM is that it may suffer from the curse of dimensionality.
- The number of dimensions in the feature space can be huge with some useful transformations even with reasonable numbers of attributes in the input space.
- This makes it computationally infeasible to handle.

`sklearn.svm`.SVC

class sklearn.svm.SVC(**, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, probability=False, tol=0.001, cache_size=200, class_weight=None, verbose=False, max_iter=- 1, decision_function_shape='ovr', break_ties=False, random_state=None*) ¶ [source]

Parameters:	**C : *float, default=1.0*** Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty. **kernel : *{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}, default='rbf'*** Specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape `(n_samples, n_samples)`.


Linear kernel
Polynomial kernel, degree=1
Polynomial kernel, degree=2
RBF kernel


RBF kernel, C=0.1
RBF kernel, C=1
RBF kernel, C=10
RBF kernel, C=100

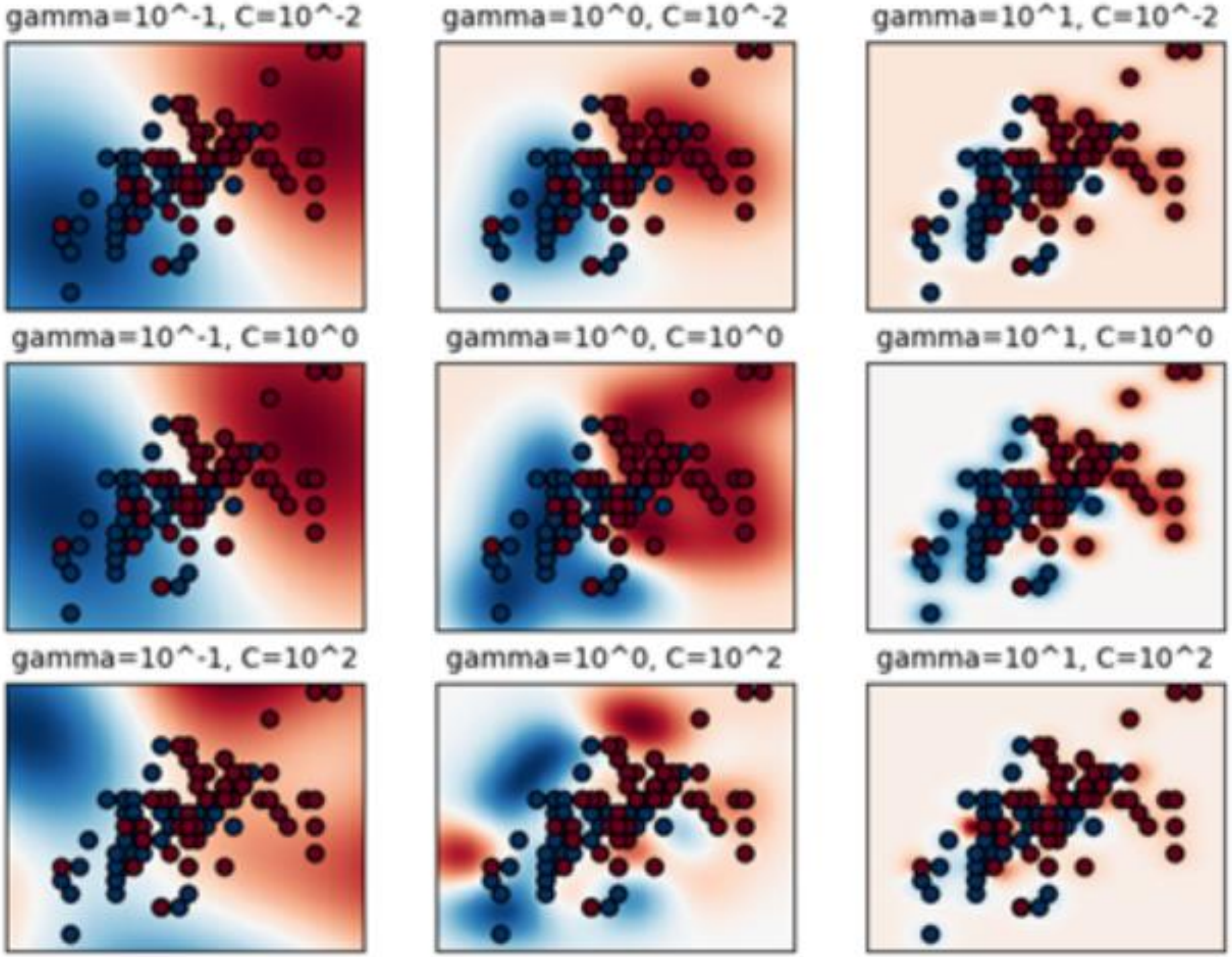


Fig 6: RBF Kernel SVM for Iris Dataset [Image Credits: https://scikit-learn.org/]

1. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Pros	**Cons**
Good linear classifier because it find the best decision boundary	Not suited for large datasets
Easy to transform into non linear model	

Some other issues in SVM

- SVM works only in a real-valued space. For a categorical attribute, we need to convert its categorical values to numeric values.
- The hyperplane produced by SVM is hard to understand by human users. The matter is made worse by kernels. Thus, SVM is commonly used in applications that do not required human understanding.