Support Vector Machines
The basics

Michael Claudius, Associate Professor, Roskilde

05.04.2020

What is Support Vector Machines (SVM) ?

- **A *powerful* versatile algorithm for both classification and regression**
- **Classification predictions are based on a so called margin a street**
  - **Classification with a largest margin, a high way**
  - **Training is based on minimizing number of instances inside margins**

- **Regression predictions are based on a so called margin street**
  - **Regression with smallest margin, a bikers lane**
  - **Training is based on minimizing number of instances outside margins**

- **So its predicting something; lets look at that !**

Evaluation of SVM?

- **Advantages**
  - **Very good for complex small/medium sized data sets**
  - **White box; knows in details how it works**
  - **Easy to use**
  - **Many forms: Linear, non-linear, with without kernel etc….**

- **Disadvantages**
  - **Slow prediction,**
  - **Complex, pipelining with scaling is needed**
  - **Greedy algorithm, (which must be stopped)**
  - **Slow for huge data sets**
  - **No probability outcome for classification**

The Iris flower case

- **Data set with 150 Iris pictures of 3 different species (50 each)**

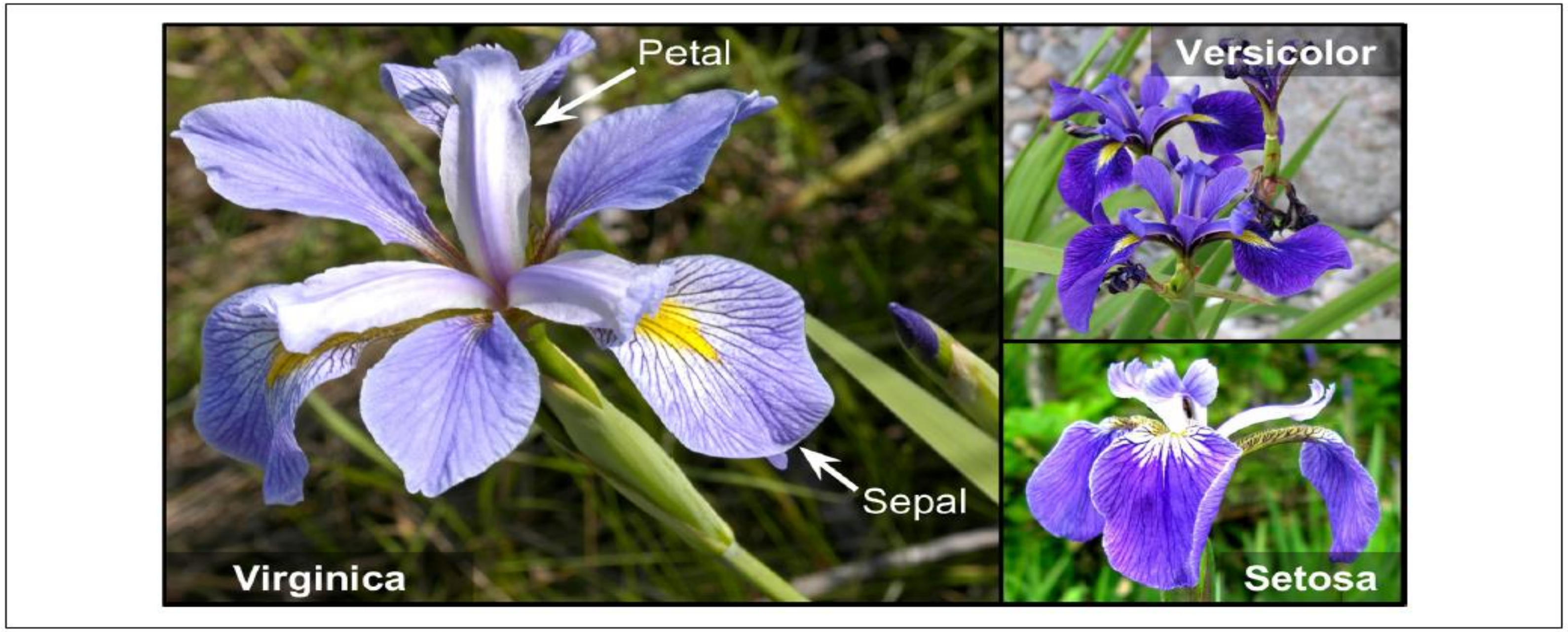

Figure 4-22. Flowers of three iris plant species[14]

Types of SVM classification

- **Linear SVM: straight line**
  - **Choose between approaches: hard-margin or soft margin**
  - **Use LinearSVC, SVC (kernel = linear) or SGDClassifier class**

- **Non-linear: curve**
  - **Choose between techniques polynomial kernel or similarity features**
  - **Use SVC (kernel = poly) or SVC (kernel = rbf)**

- **In practise use Claudius rule: simple ones first**
  - **LinearSVC**
  - **SVC only for polynomial, max degree 3 to avoid overfitting**
  - **SGD (only if out of core problems)**
  - **ANN (Artificial Neural Network) good alternative for complex data sets**

Choose and build classifier(s)

- **Find margin interval (street edge) defined by support vectors**
- **Make a choice on the decision boundaries (Linear/curved)**
- **Apply scaling**
- **The aim is to find variables and values that split the data into groups**
  - **Maximizing the margin interval**

- **The decision boundaries is based on petal length and petal width**
- **Outcome is Iris Virginica OR Iris Setosa, BUT NOT probability**
- **Using several training algorithm to see which one is best…**

Lets see how it looks on next slide!
BUT First watch an easy video introduction SVM Introduction (20 minutes)

Hard margin classification

- **SVM decision boundary on Iris data set**
- **Either it is in or it is out**

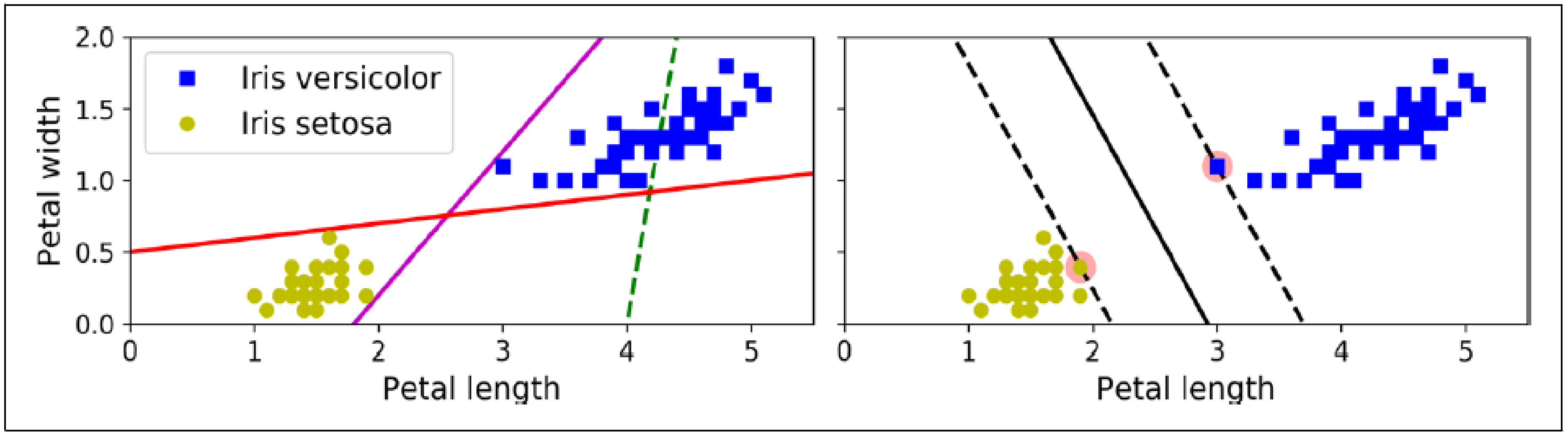

Figure 5-1. Large margin classification

SVM Scaling

- **Utilize scaling to solve the problem of sensitivity to feature scales**

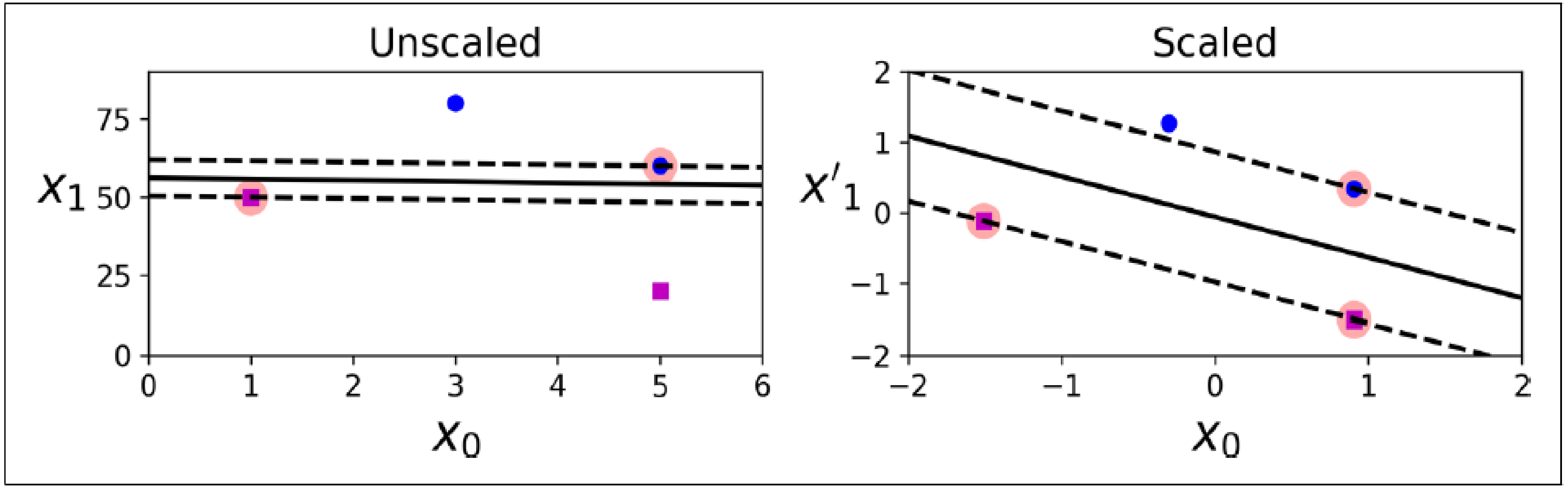

Figure 5-2. Sensitivity to feature scales

Code for Iris data set

- **Import libraries**
- **Set up a pipeline with scaling**

```
iris = datasets.load_iris()
X = iris["data"][:, (2, 3)]  # petal length, petal width
y = (iris["target"] == 2).astype(np.float64)  # Iris virginica

svm_clf = Pipeline([
        ("scaler", StandardScaler()),
        ("linear_svc", LinearSVC(C=1, loss="hinge", random_state=42)),
    ])

svm_clf.fit(X, y)
svm_clf.predict([[5, 2]]) from sklearn.datasets
```

- **What about probability. NO! Cannot predict probability**

SVM Outliers problems

- **Hard margin is sensitive to outliners**
- **And some times impossible to use**

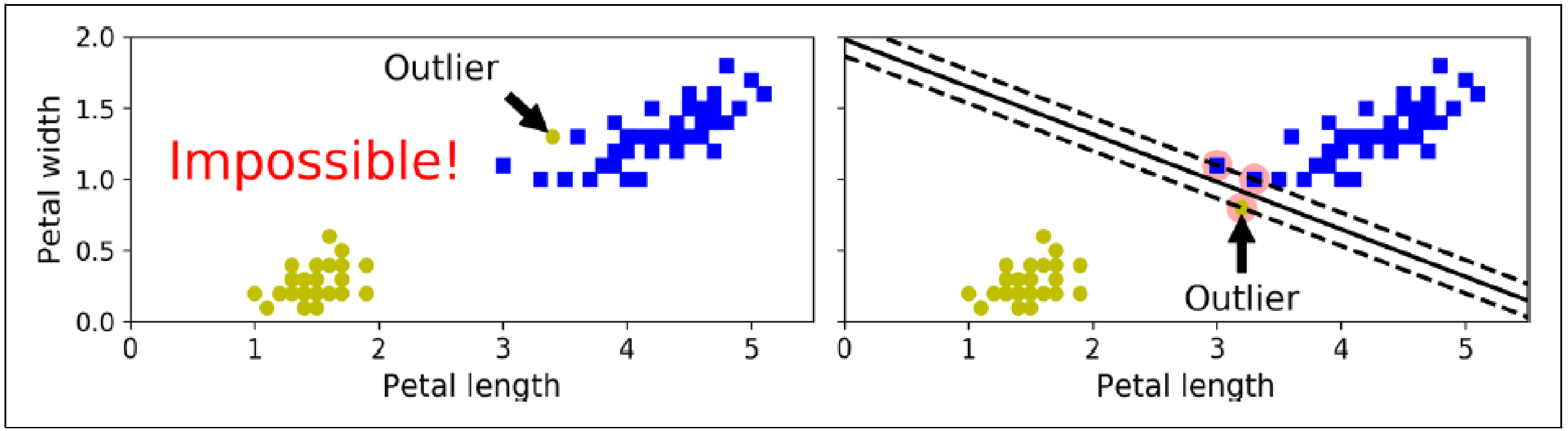

Figure 5-3. Hard margin sensitivity to outliers

- **We are lucky: Soft margin is the answer!**

Soft margin

- **Allows margin violations**
- **C=1: Large margin many violations; C=100 Small margin but few violations**

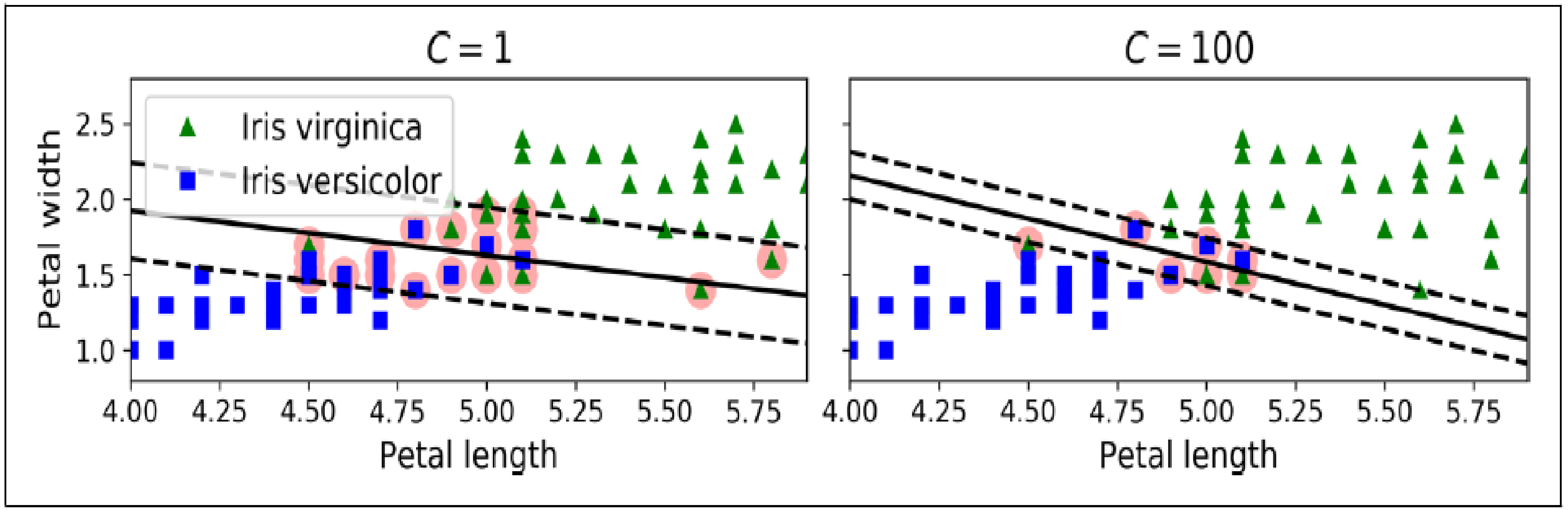

Figure 5-4. Large margin (left) versus fewer margin violations (right)

- **But there are better solutions, but more time greedy Nonlinear SVM Classification**

Nonlinear SVM classification: polynomial features

- **Add extra polynomials e.g. up to degree 3 for each feature: $X_2 = (X_1)^2$ and $X_3 = (X_1)^3$ to the data set.**
- **Remember linear regression: *h(X)* = $\theta_0 + \theta_1 X_1 + \theta_2 X_2 + ….. + \theta_n X_n$ $\theta_0 + \theta_1(X_1)^1 + \theta_2(X_1)^2 + \theta_3(X_1)^3$**
- **Making the data set linear separable as shown for degree 2 below**

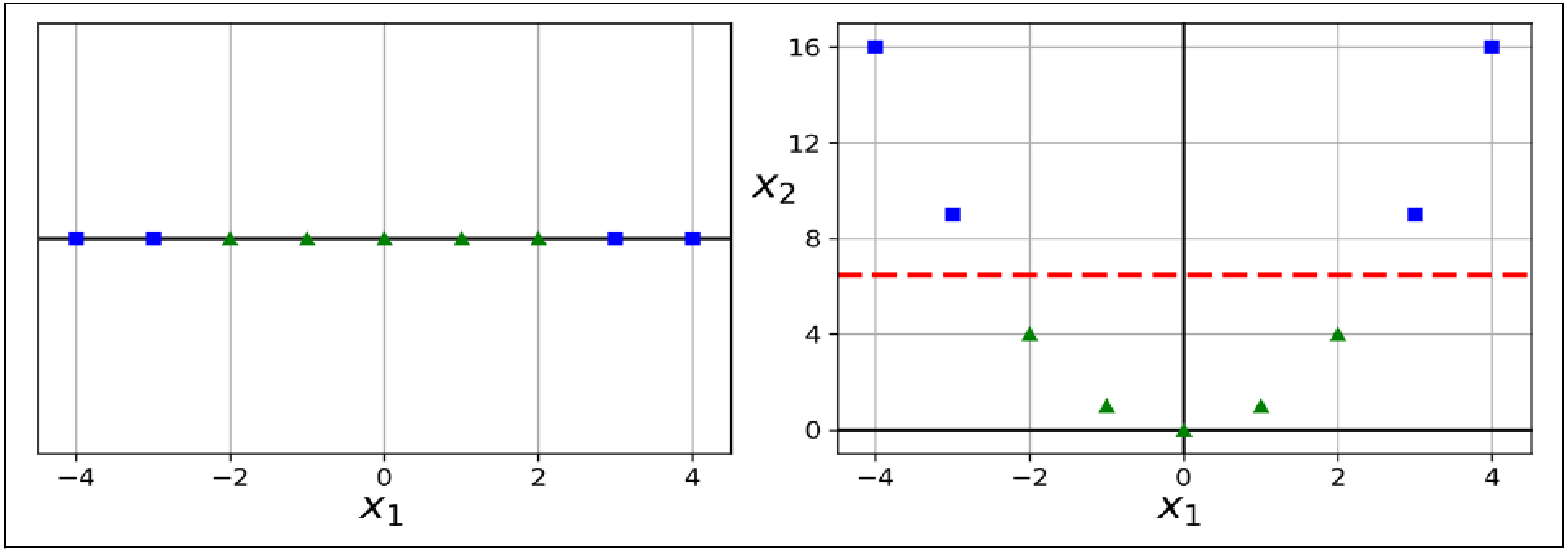

Figure 5-5. Adding features to make a dataset linearly separable

Code for Moon data set

- **Import libraries**
- **Set up a pipeline with polynomial features and scaling**

```
import make_moons
X, y = make_moons(n_samples=100, noise=0.15, random_state=42)

polynomial_svm_clf = Pipeline([
        ("poly_features", PolynomialFeatures(degree=3)),
        ("scaler", StandardScaler()),
        ("svm_clf", LinearSVC(C=10, loss="hinge", random_state=42))
    ])

polynomial_svm_clf.fit(X, y)
```

- **What about probability. NO! Cannot predict probability**

Nonlinear SVM with polynomial features

- **Now we get soft boundary lines**

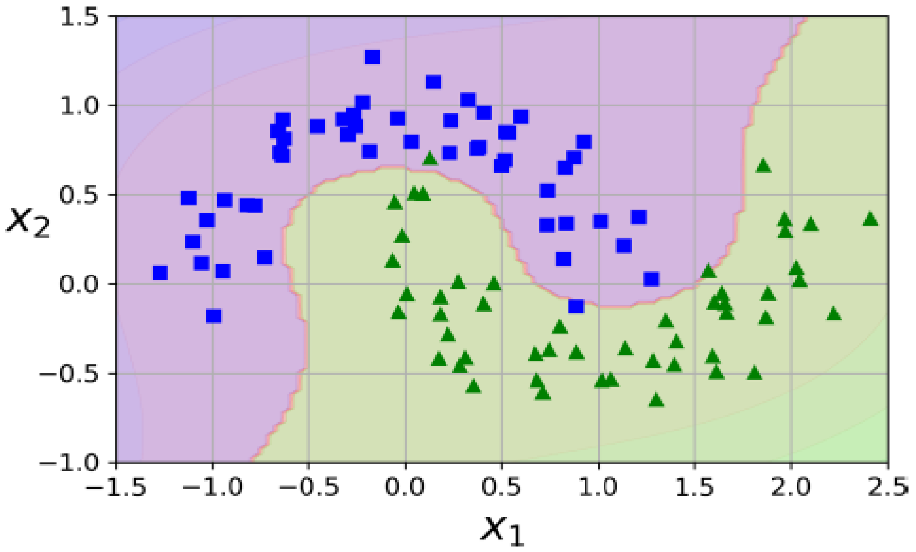

Figure 5-6. Linear SVM classifier using polynomial features

- **But there are problems. Oh no not that again☹**

Problems with Polynomial features

- **Low degree cannot handle complex data set**
- **High degree have too many features => very slow**

- **But we are lucky again.**
- **The kernel trick**
  - **Solution A: Polynomial kernel**
  - **Solution B: Similarities features**

Nonlinear SVM with polynomial kernel

- **Soft boundary lines**
- **Overfitting is an issue**

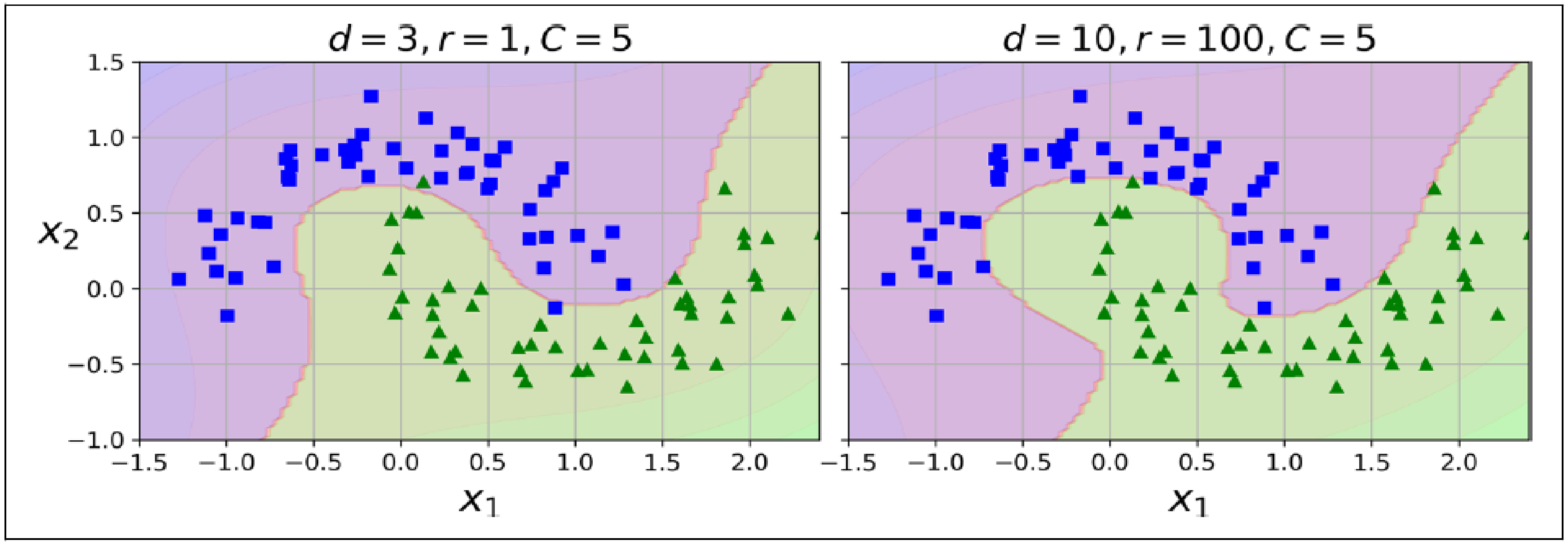

Figure 5-7. SVM classifiers with a polynomial kernel

- **No more problems. But the math behind is complicated.**

Computational complexity

- **Big O notation O(m x n)**
- **m: number of instances**
- **n: number of features**

Table 5-1. Comparison of Scikit-Learn classes for SVM classification

Class	Time complexity	Out-of-core support	Scaling required	Kernel trick
`LinearSVC`	$O(m \times n)$	No	Yes	No
`SGDClassifier`	$O(m \times n)$	Yes	Yes	No
`SVC`	$O(m^2 \times n)$ to $O(m^3 \times n)$	No	Yes	Yes

- **Example: m = 10.000 n= 5**
- **LinearSVC: 50.000 0.1 second**
- **SVC(10.000 x 10.000 x 5 = 500.000.000, 1.000 seconds !!**

Types of SVM regression

- **Idea: Largest possible street with many instances on the street and few instances off the street/margins**
- **(a crowded biker' lane)**

- **Linear SVM: straight line**
  - **Choose between approaches: hard-margin or soft margin**
  - **Use LinearSVR similar to LinearSVC**

- **Non-linear: curve**
  - **Use SVR, SVR (kernel = poly) similar to SVC (kernel = poly)**

Under the hood or Assignments

- **Under the hood or What goes on behind the stage: Complicated mathematics !**
- ***I will skip it.***

- **It is time for discussion and solving a few assignments in groups**

  - [Problems problems enjoy!]()
  - [SVM Iris Exercise]()