Week 8

Machine Learning and Big Data - DATA622

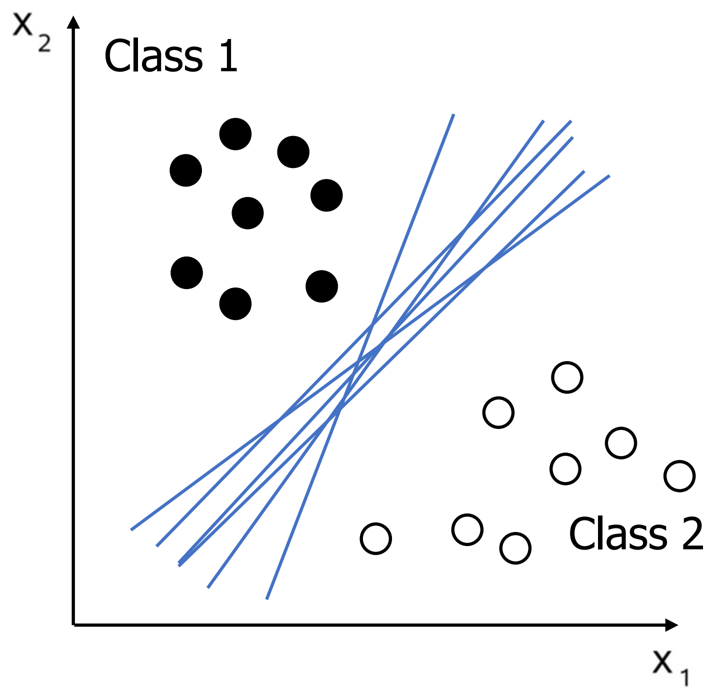
CUNY School of Professional Studies

Support Vector Machines

CUNY School of Professional Studies

Support Vector Machines

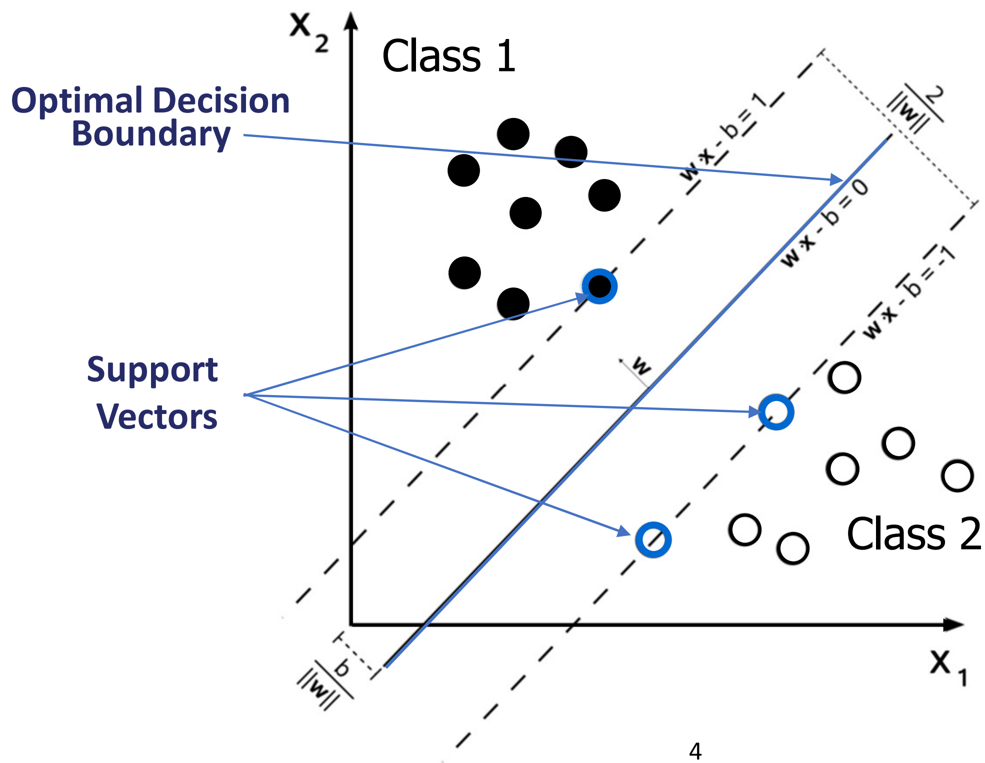
How do you find the right decision boundary?



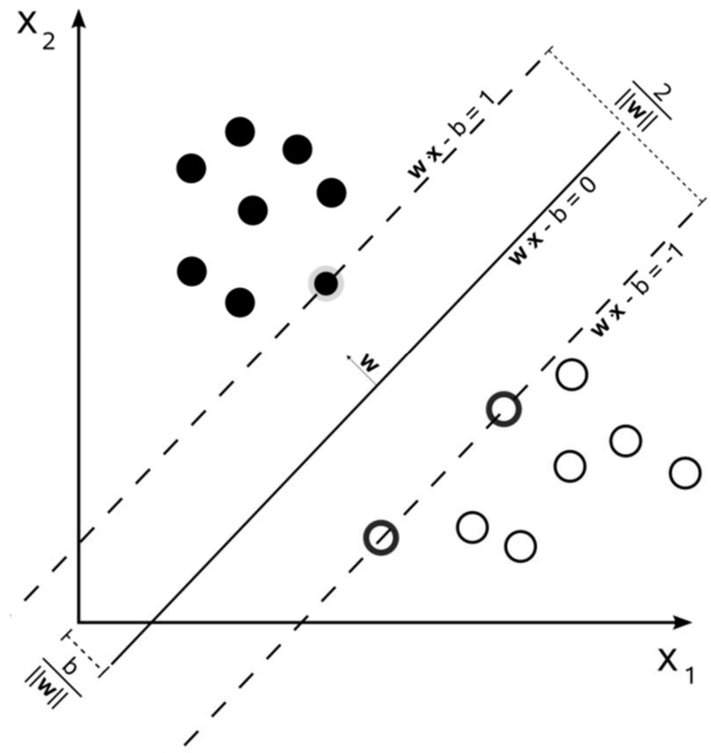
CUNY School of Professional Studies

Support Vector Machines

SVM approach: Maximize the margin, to define the decision boundary



Support Vector Machines



$$\max \frac{2}{\|w\|}$$

subject to

$$y_n \begin{cases} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{cases}$$

which we can write as $y_n(w^T x + b) \geq 1$

$$max\frac{2}{||w||} = max\frac{1}{||w||} = min||w|| = min\frac{1}{2}||w||^2$$

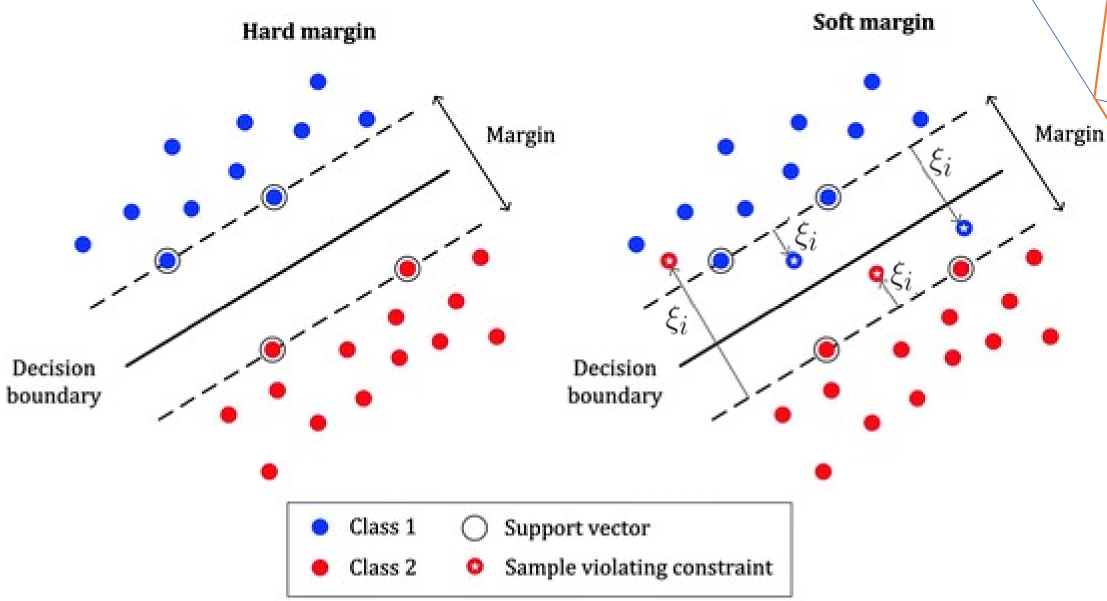
$$b = \frac{1}{S}\sum_{i=1}^{S}(y_i - w \cdot x)$$

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$

CUNY School of Professional Studies

Soft-margin SVM

So far we have considered the case with no errors (hard margin).
How do we handle the case where there is is overlap?

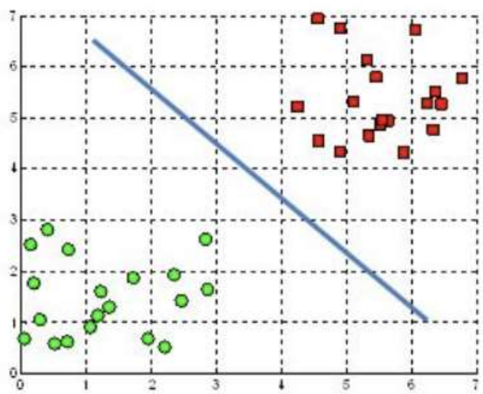


$$\underbrace{\min \frac{1}{2}\|w\|^2}_{\text{regularizer}} + \underbrace{C_i \sum_{i=1}^{n} \xi_i}_{\text{error term}}$$

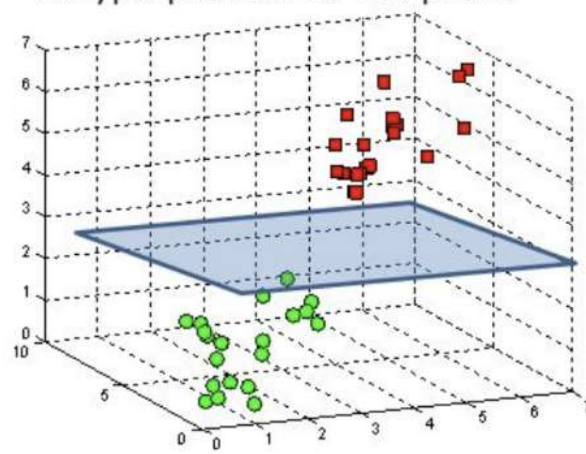
Hyperplanes

A decision boundary is defined by a hyperplane

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane

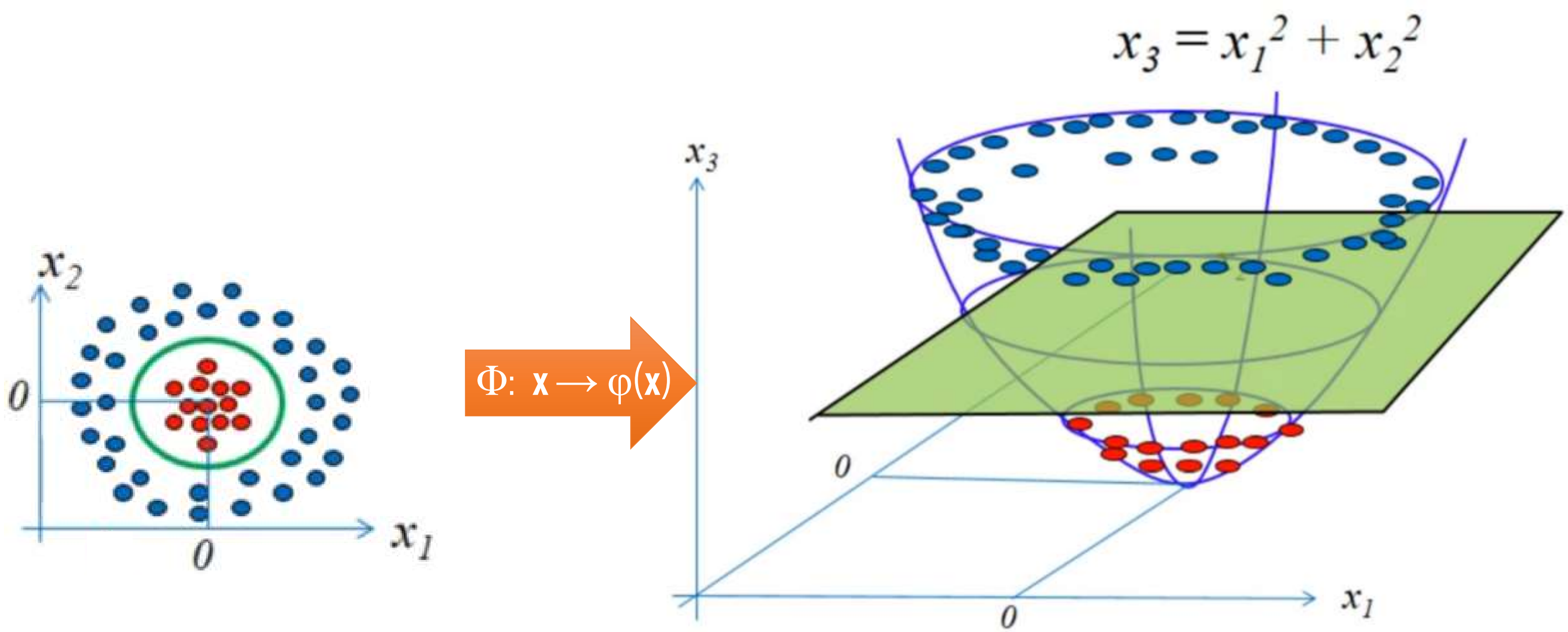


Kernels

What do we do about non-linear decision boundaries?

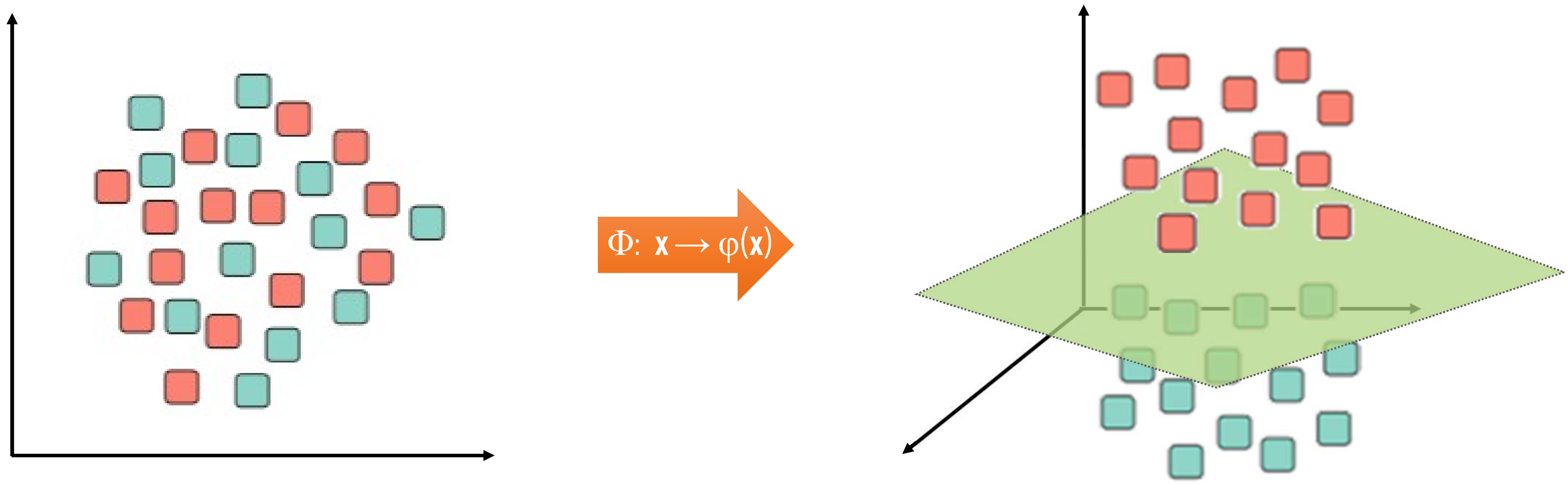
Non-linear decision boundaries

- Map the original input space to a higher-dimensional feature space where the training set is separable.



The "kernel trick"

- Kernels allow us to keep using a linear classifier – because we have embedded the data in a higher-dimensional feature space



CUNY School of Professional Studies

The “kernel trick’

- The linear classifier relies on dot product between vectors:
  $$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$
- If every data point is mapped into high-dimensional space via some transformation
  $$\Phi: \mathbf{x}_i \rightarrow \varphi(\mathbf{x}_i)$$
  the dot product becomes:
  $$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$
- A *kernel function* is similarity function that corresponds to an inner product in some expanded feature space
- *The kernel trick*: instead of explicitly computing the lifting transformation $\varphi(\mathbf{x})$, define a kernel function K such that:
  $$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$
- Goal: Avoid having to directly construct $\varphi()$ at any point in the algorithm

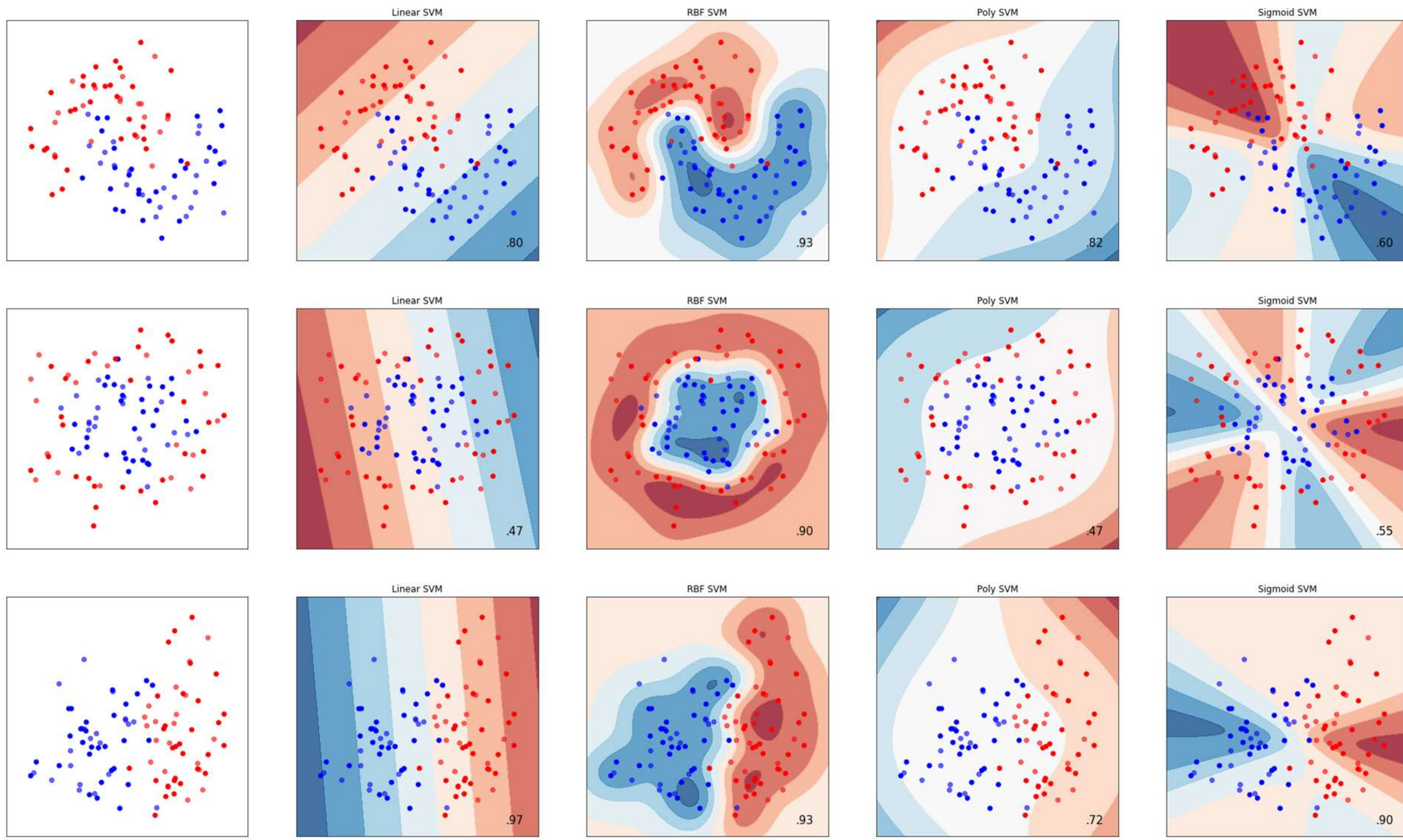
The “kernel trick’

- Kernel trick is allows extremely complex ϕ( ) while keeping K(a,b) simple
- Goal: Avoid having to directly construct ϕ( ) at any point in the algorithm

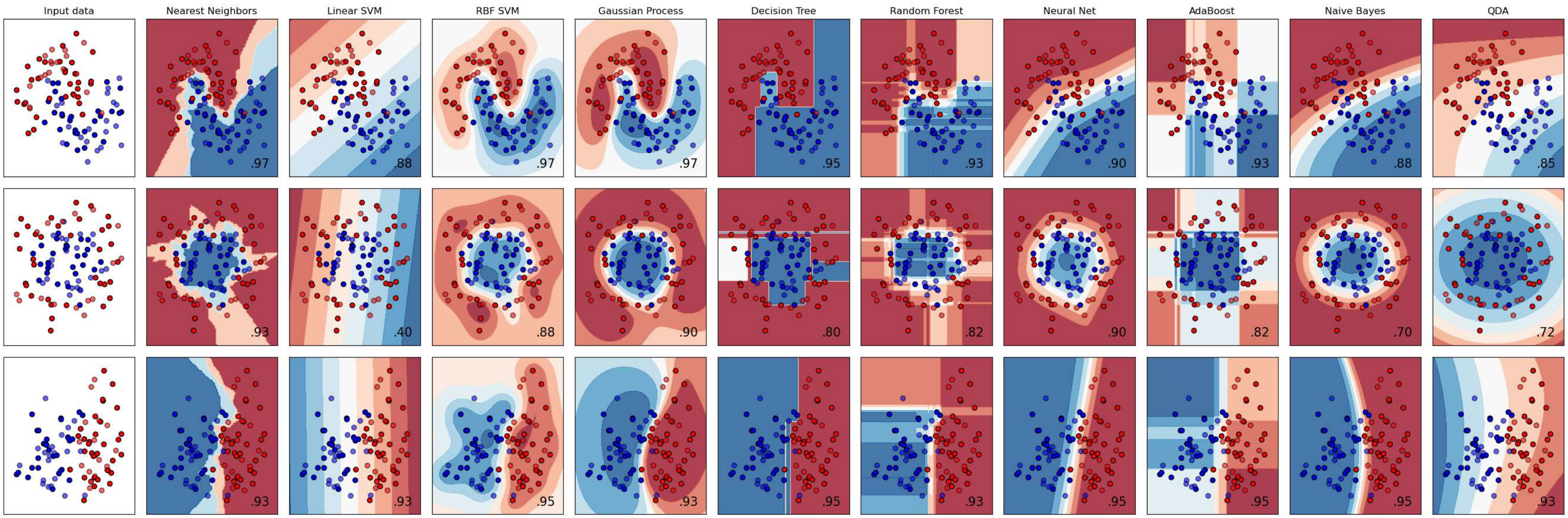
Common kernels

1. Polynomial kernel: $k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$
2. Gaussian kernel: $k(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$
3. Gaussian radial basis function (RBF): $k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma ||\mathbf{x_i} - \mathbf{x_j}||^2)$
4. Laplace RBF kernel: $k(x, y) = \exp\left(-\frac{||x-y||}{\sigma}\right)$
5. Sigmoid kernel: $k(x, y) = \frac{J_{v+1}(\sigma ||x-y||)}{||x-y||^{-n(v+1)}}$
6. ANOVA radial basis kernel: $k(x, y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$
7. Linear splines kernel in one-dimension: $k(x, y) = 1 + xy + xy\ min(x, y) - \frac{x+y}{2}\ min(x, y)^2 + \frac{1}{3}\min(x, y)^3$

Comparison of kernels



Comparison of kernels



Source: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

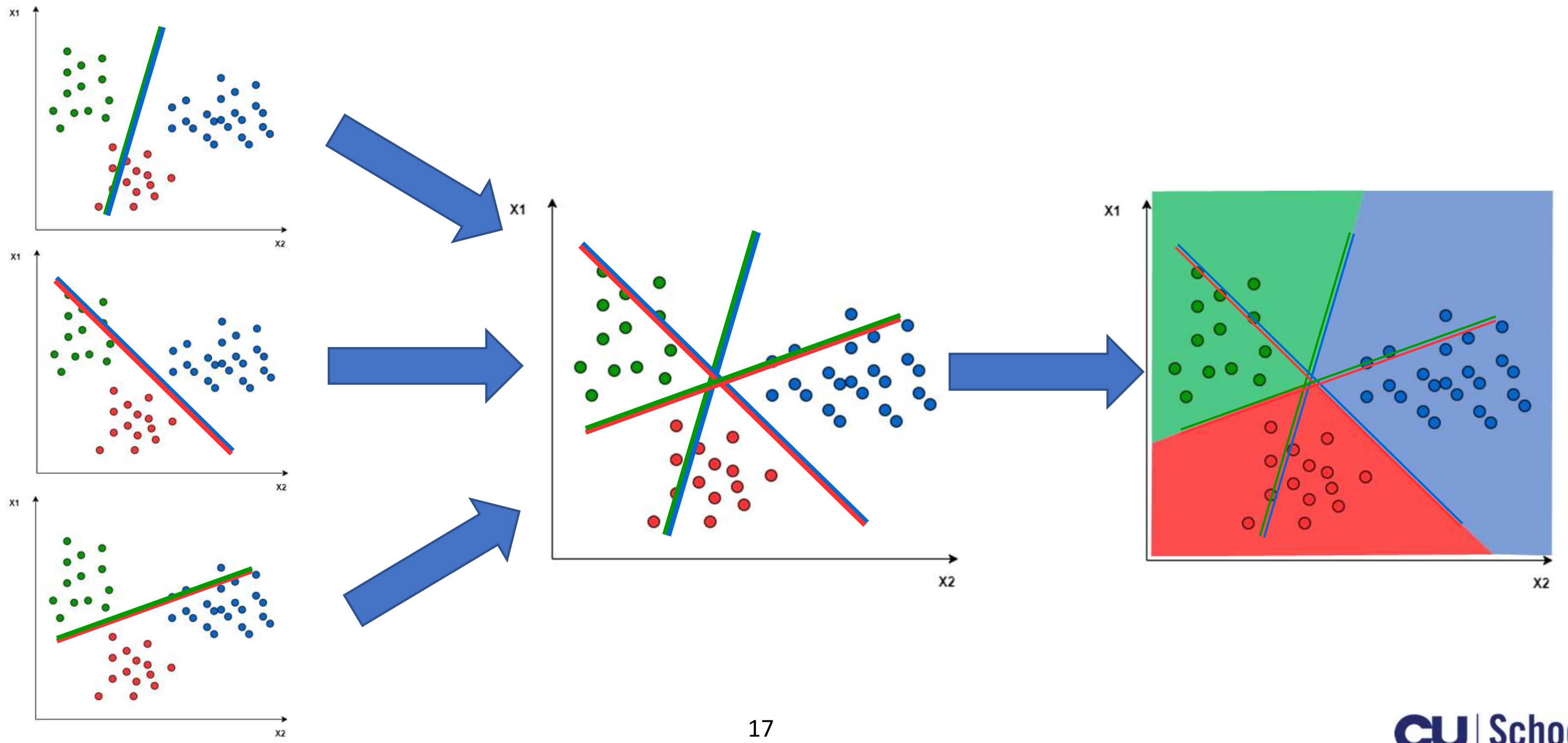
Multi-class Classification

Turning a binary classifier into a multi-class classifier

CUNY School of Professional Studies

Multi-class SVM: One-vs-one

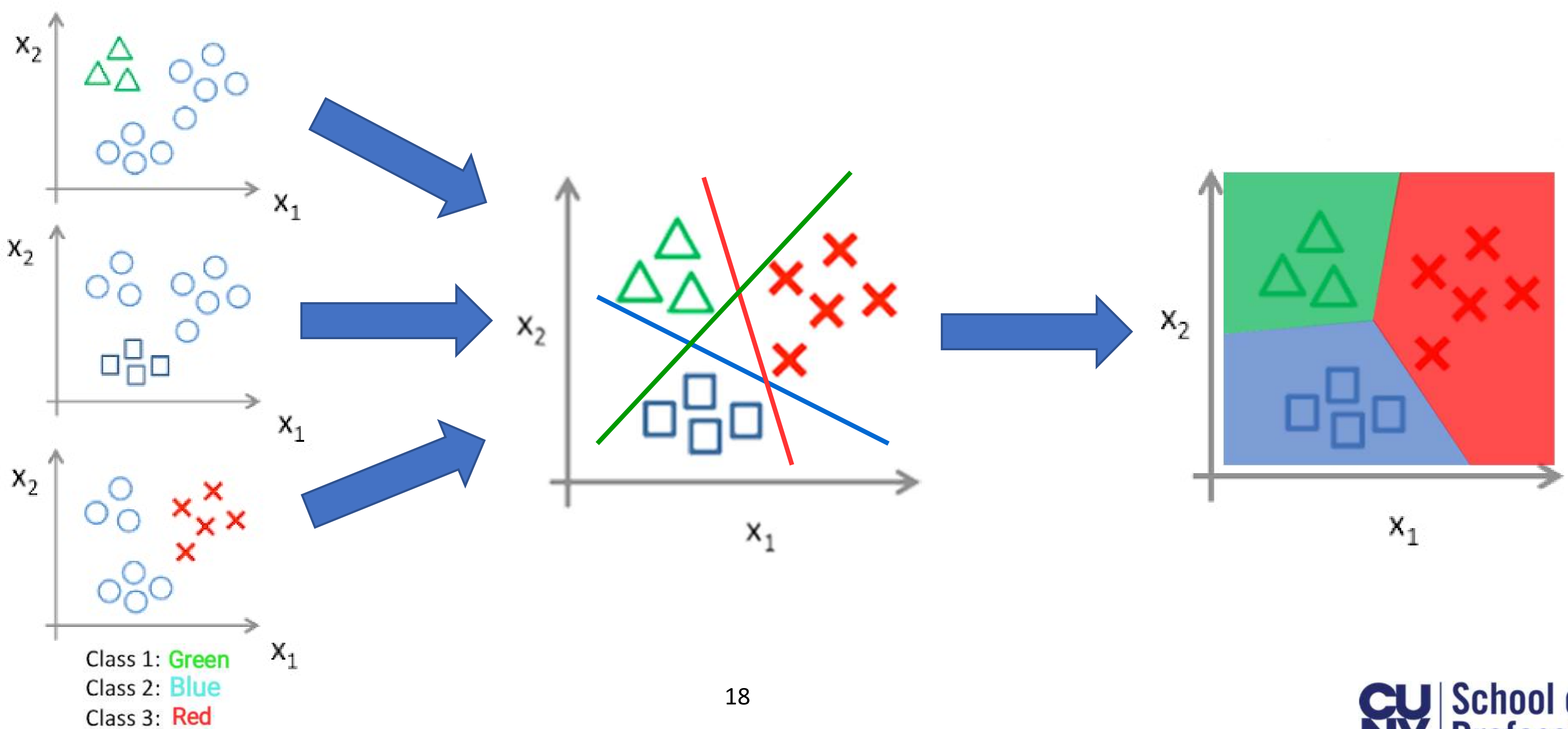
A hyperplane is generated between every two classes, neglecting the points of the third class.



17

Multi-class SVM: One-vs-rest

A hyperplane is generated each class and all other classes at once.



Properties of SVMs

Benefits and weaknesses of SVMs

CUNY School of Professional Studies

Benefits of SVMs

- Works well on smaller cleaner datasets
- It can be more efficient because it uses a subset of training points
- Different kernel functions can be specified for the decision function
- Sparseness of solution when dealing with large data sets
  - Only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - Complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Guaranteed to converge to a single global solution (convex optimization)

20

Weakness of SVMs

- Sensitive to noise
  - A relatively small number of mislabeled examples can dramatically decrease the performance
- Essentially a binary classifier
- Isn’t suited to larger datasets as the training time with SVMs can be high
- Less effective on noisier datasets with overlapping classes
- If the number of features is a lot bigger than the number of data points, avoiding over-fitting when choosing kernel functions and regularization term is crucial.

21