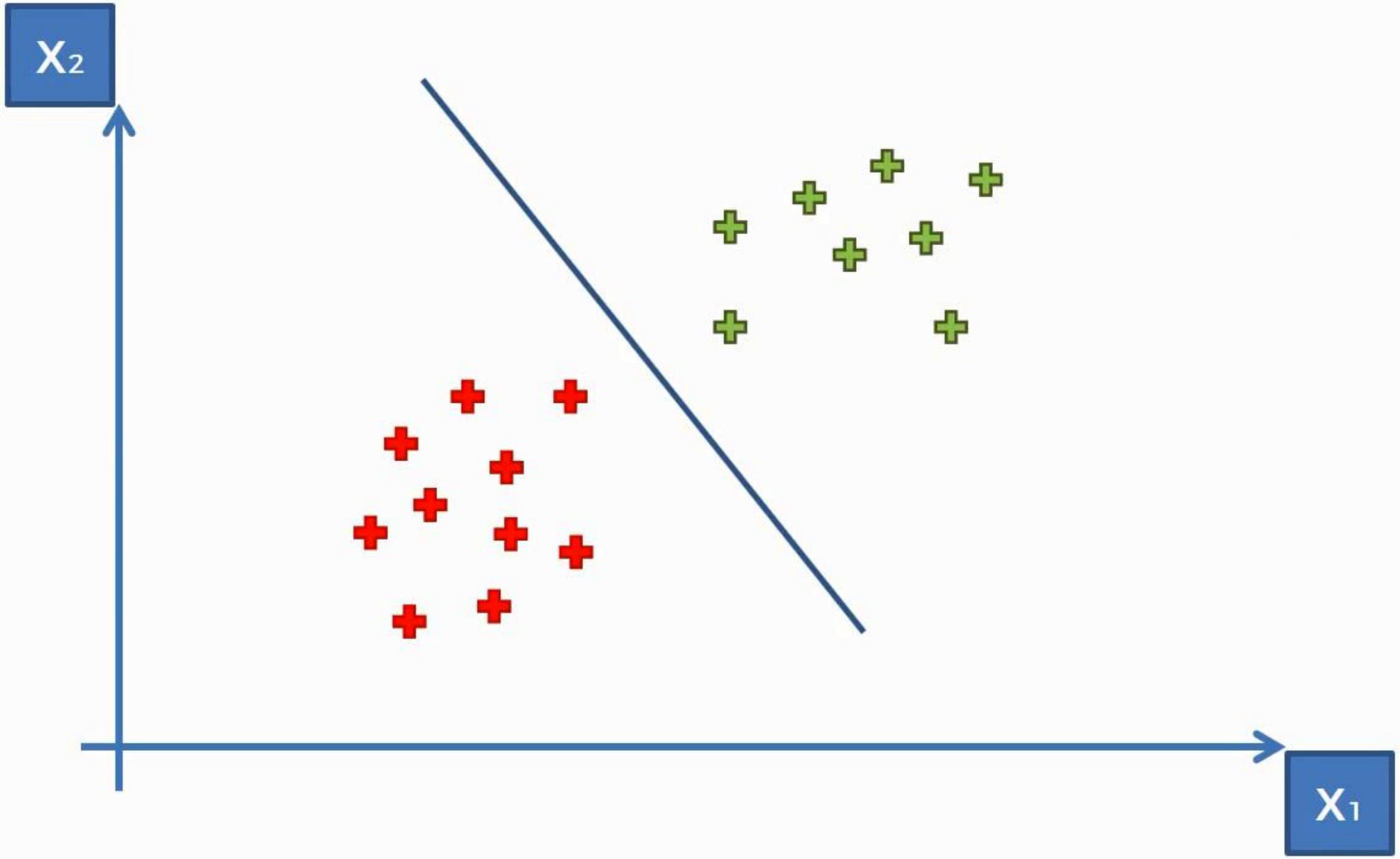
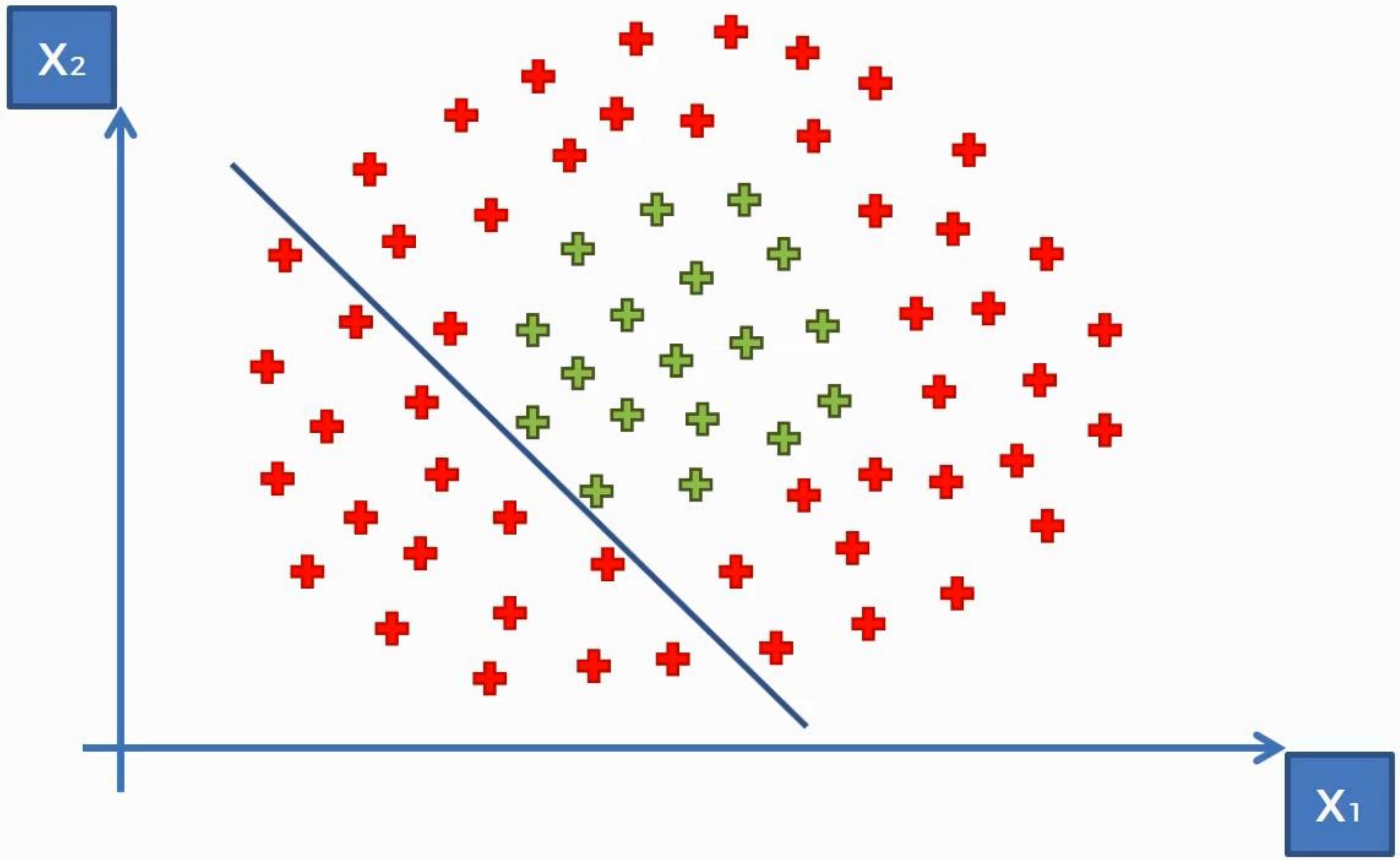
Kernel SVM

Non Parametric, Binary Classification Technique

SVM separates well these points

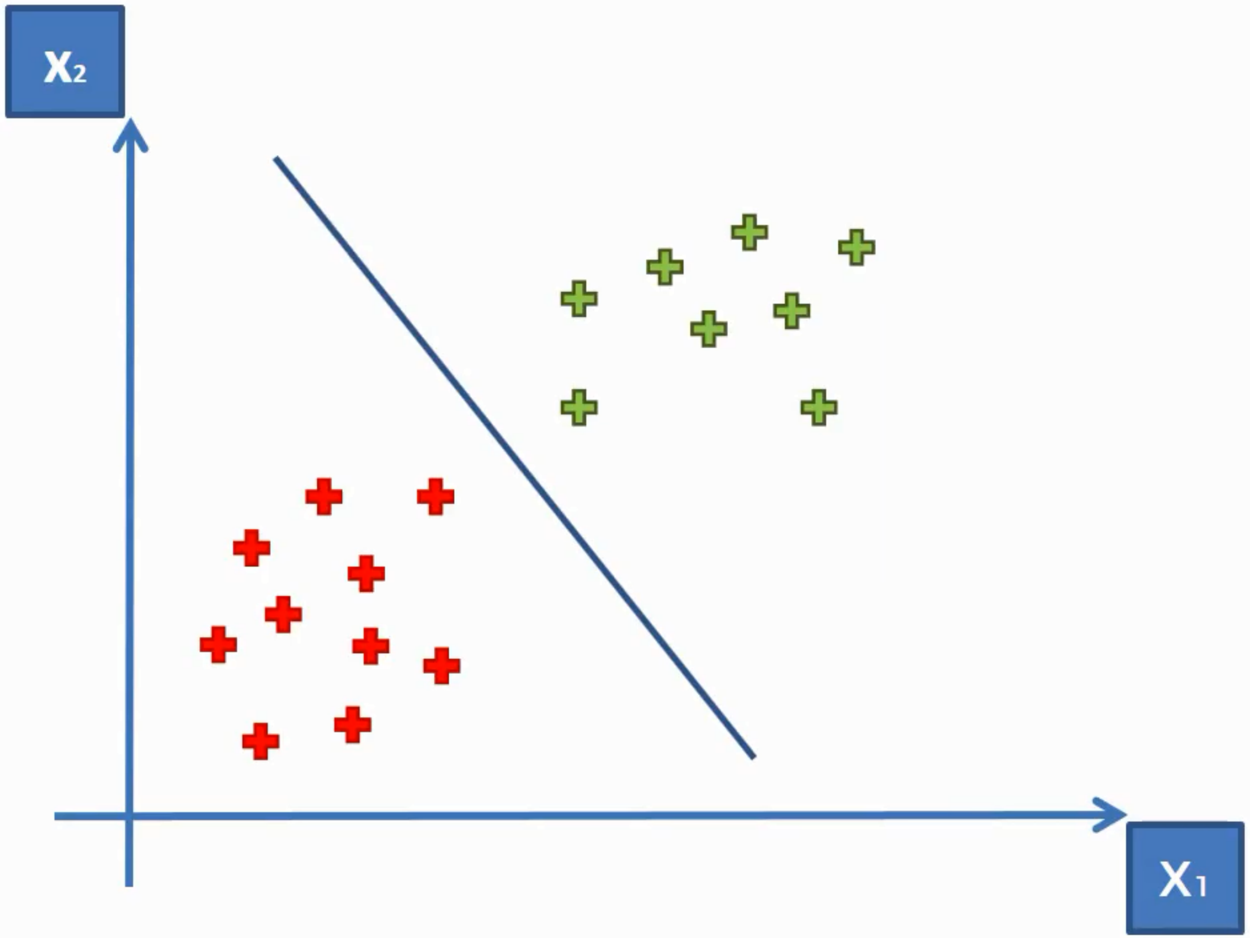


What about these points ?

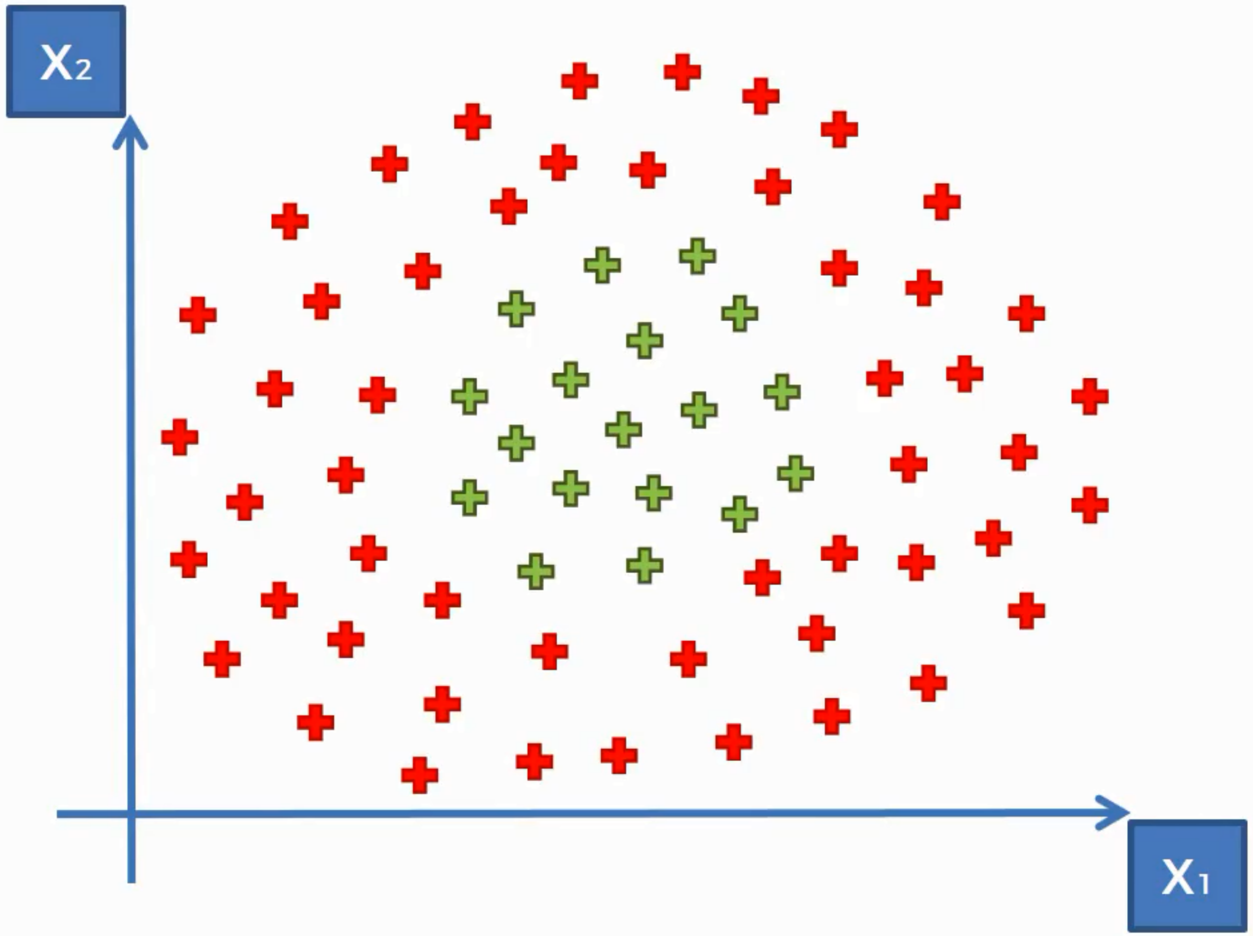


Linear Separability

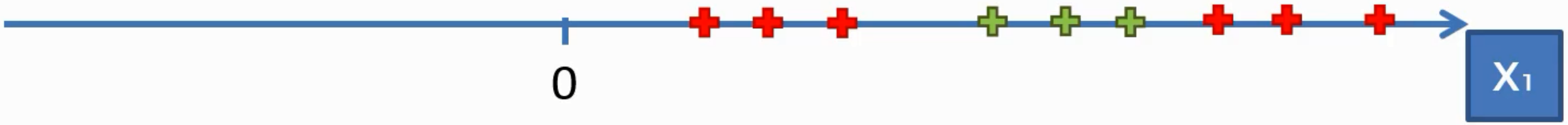
Linearly Separable



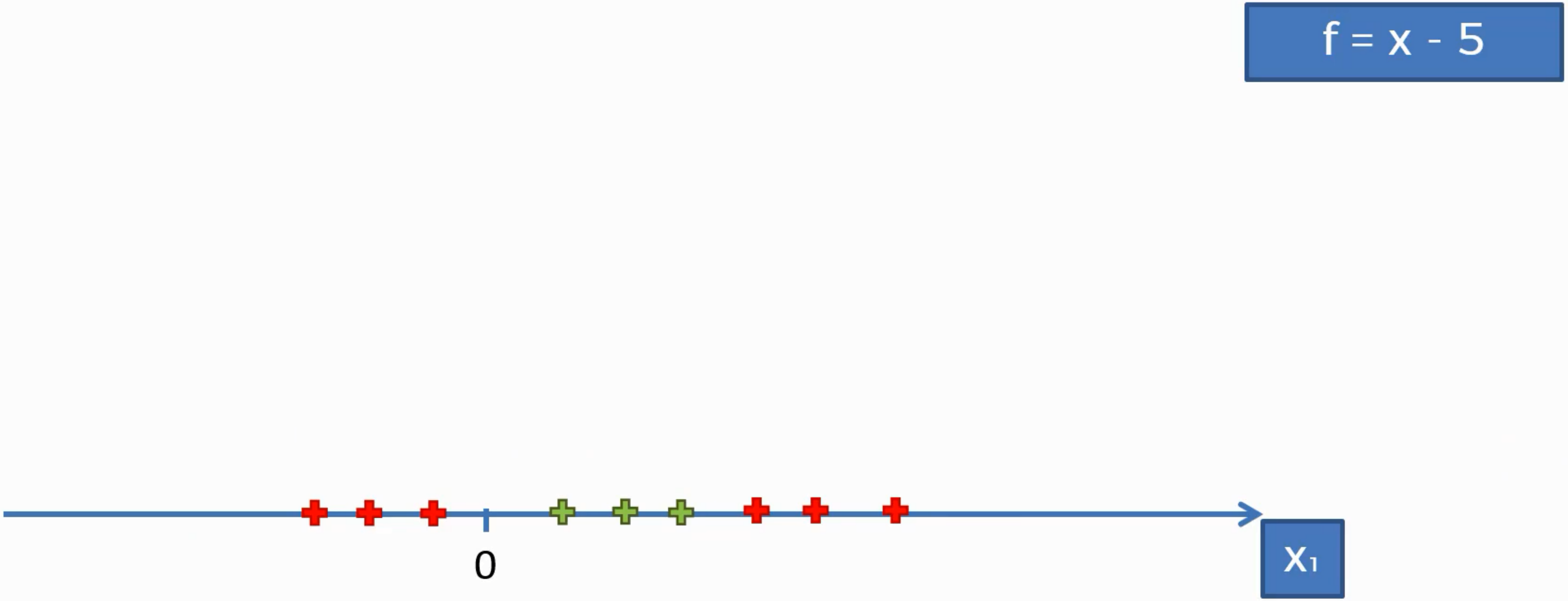
Not Linearly Separable



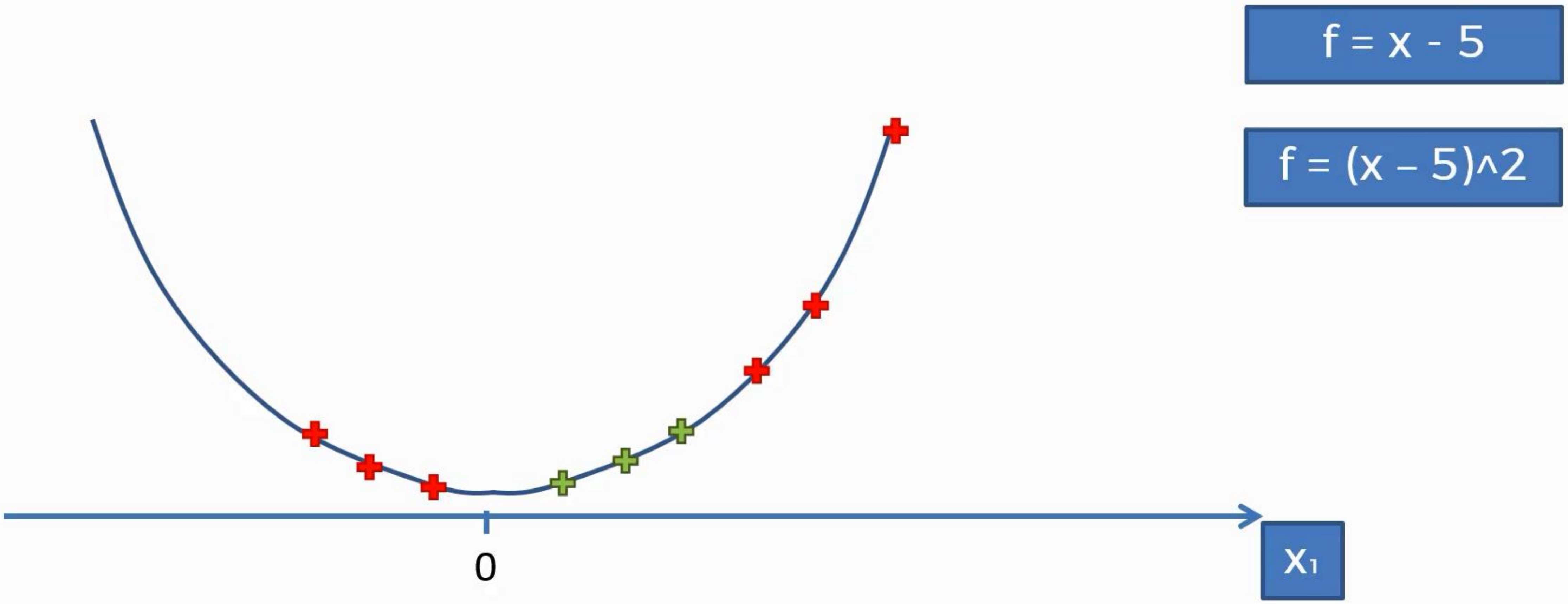
Mapping to a Higher Dimension



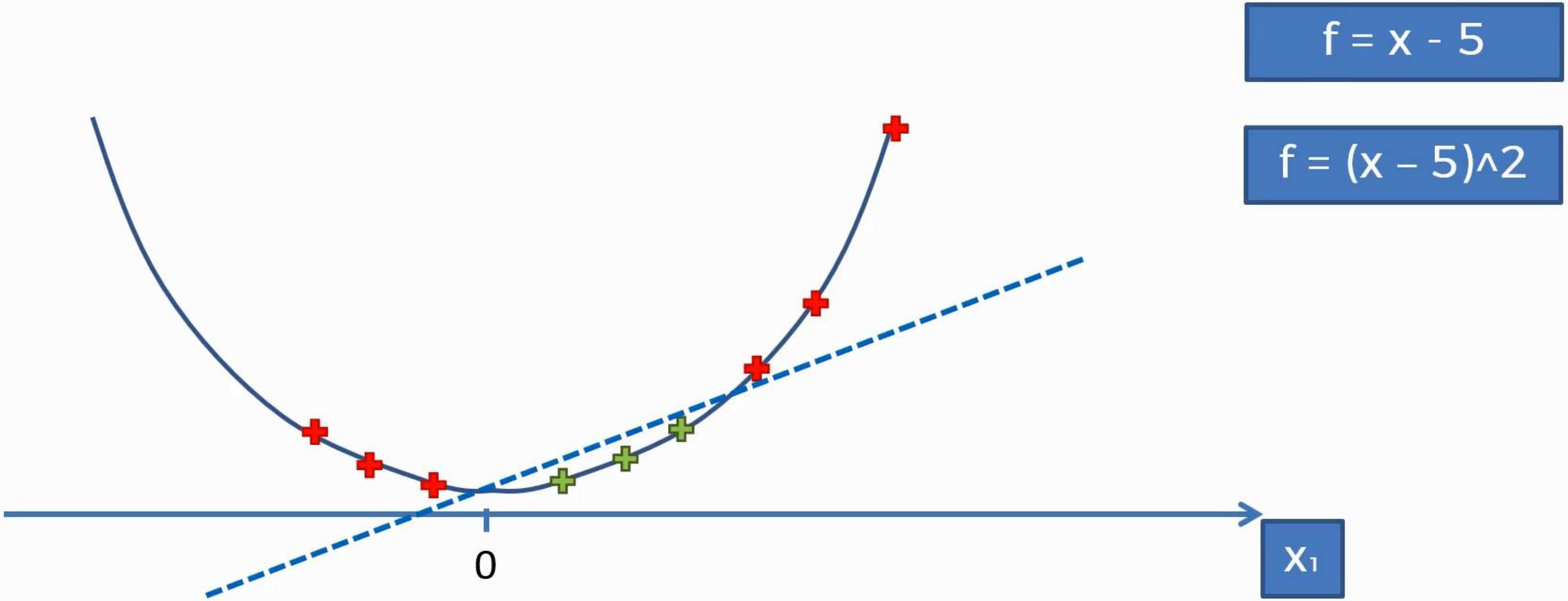
Mapping to a Higher Dimension



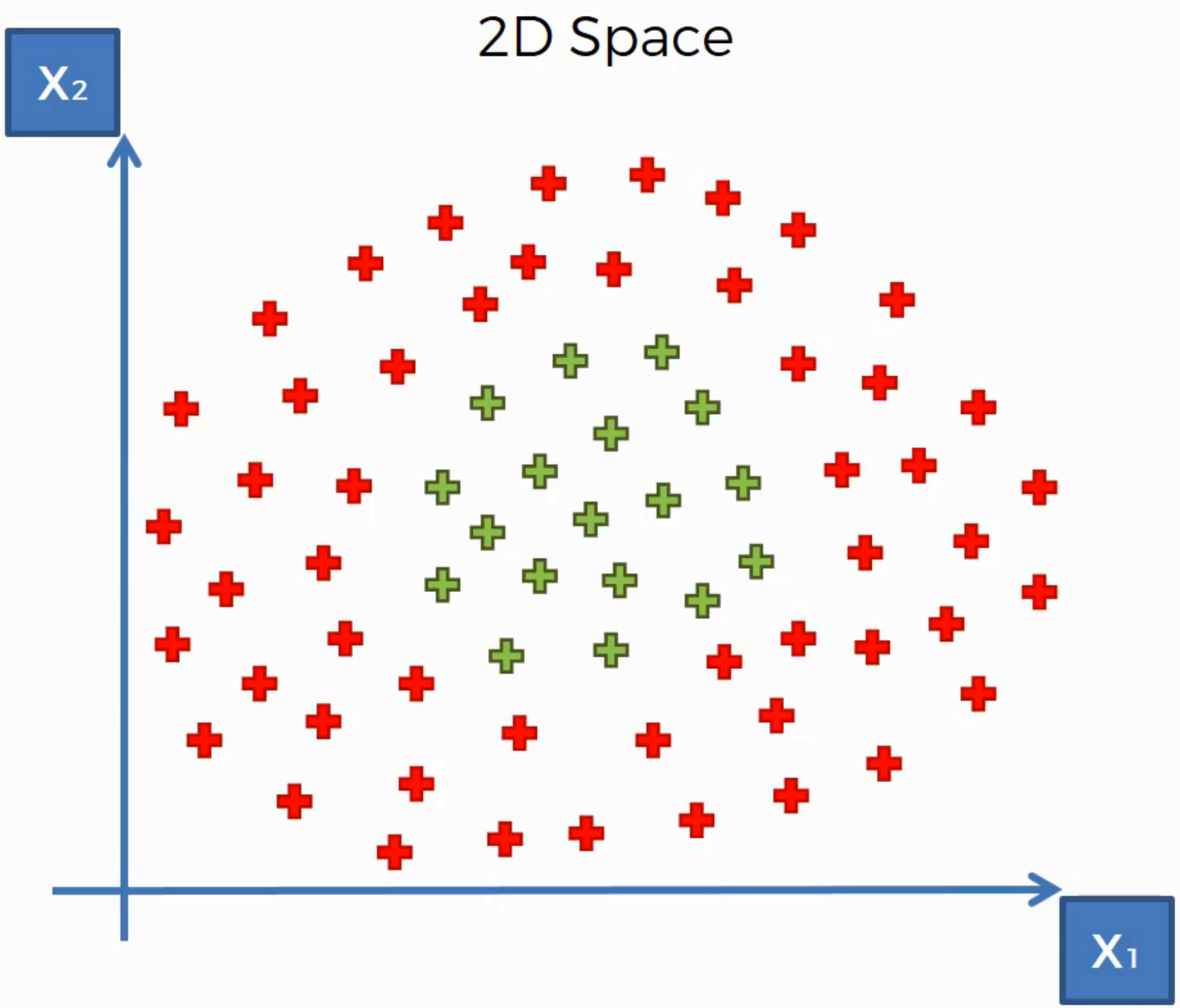
Mapping to a Higher Dimension



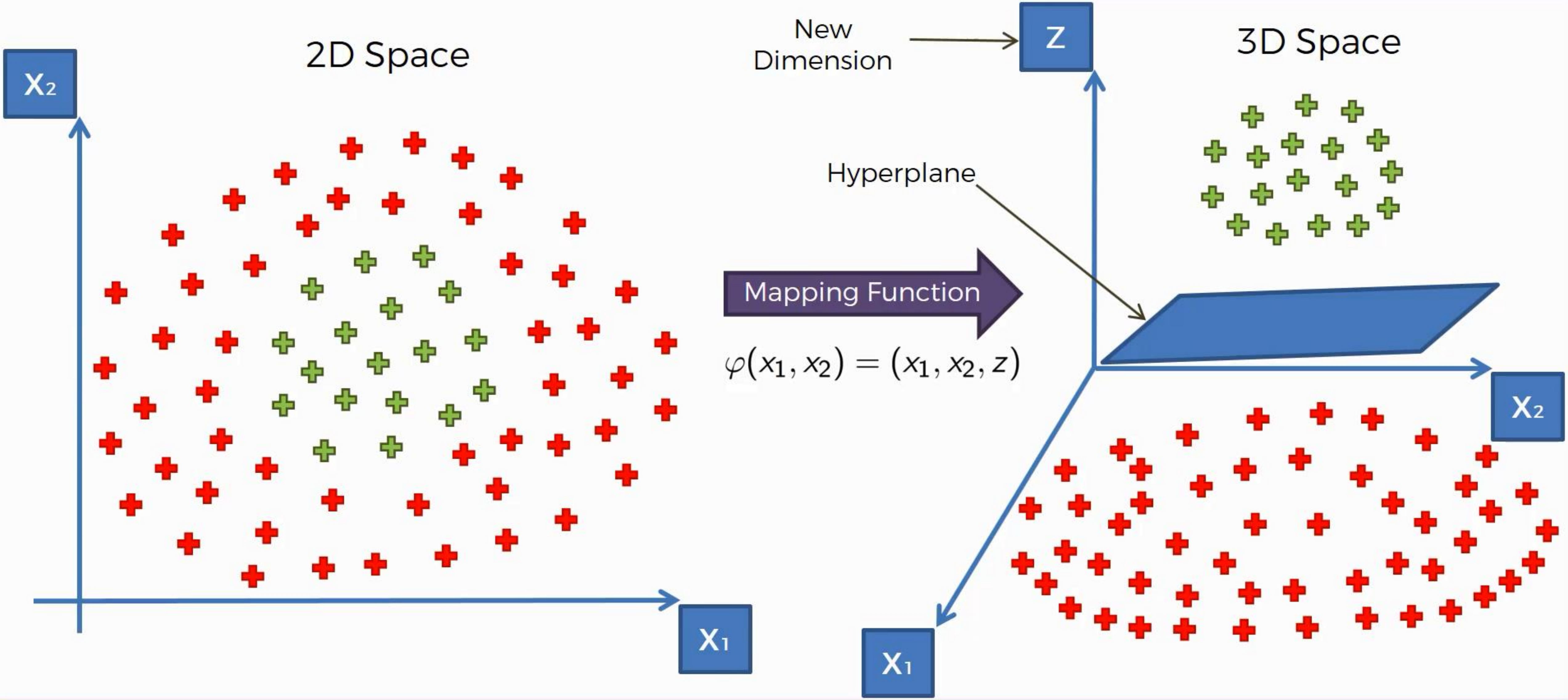
Mapping to a Higher Dimension



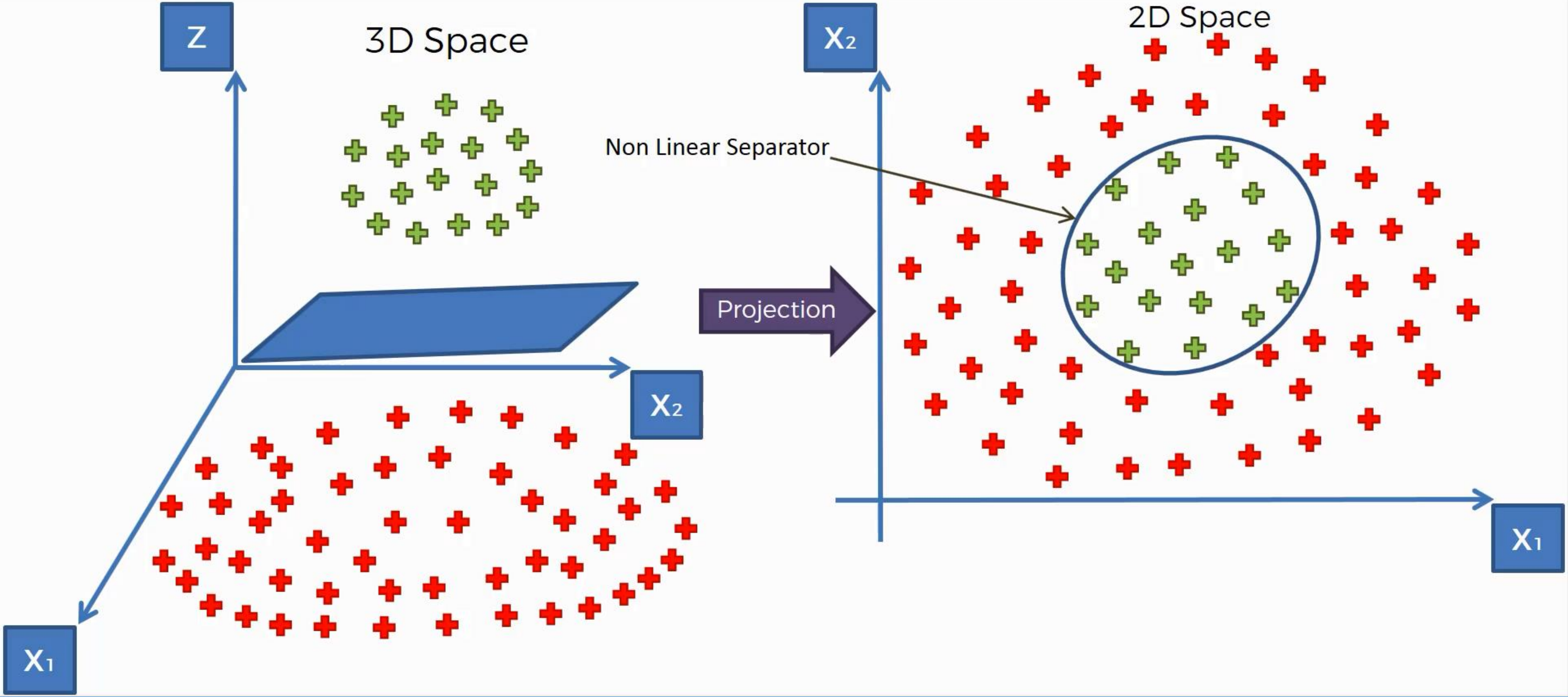
Mapping to a Higher Dimension



Mapping to a Higher Dimension



Projecting back to 2D Space



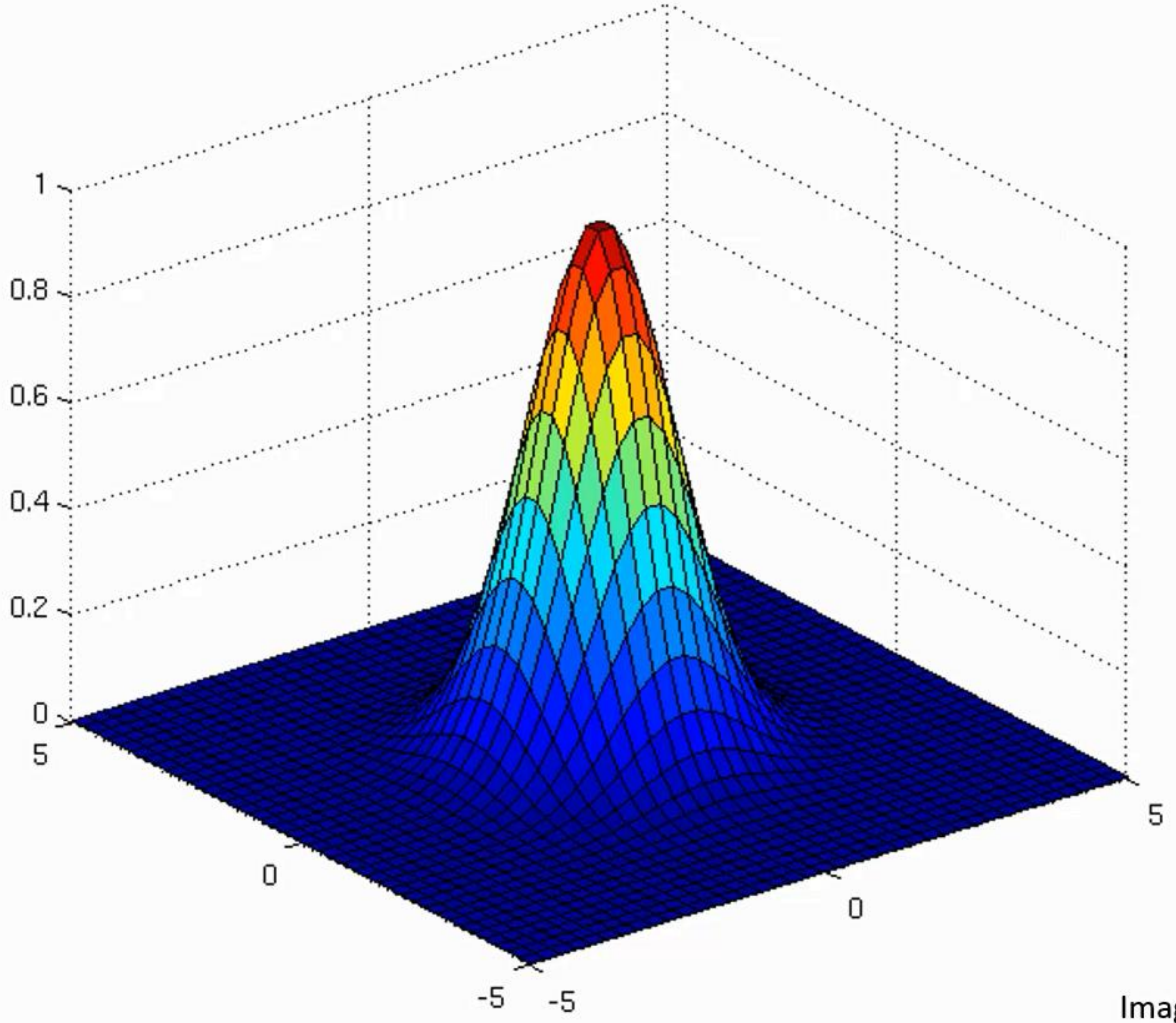
Challenges

- Mapping to a higher dimension can’t always be visualized
- Mapping to a higher dimension is highly computation-intensive

The Gaussian RBF Kernel

$$K\left(\vec{x}, \vec{l}^{i}\right) = e^{-\frac{\left\|\vec{x}-\vec{l}^{i}\right\|^{2}}{2\sigma^{2}}}$$

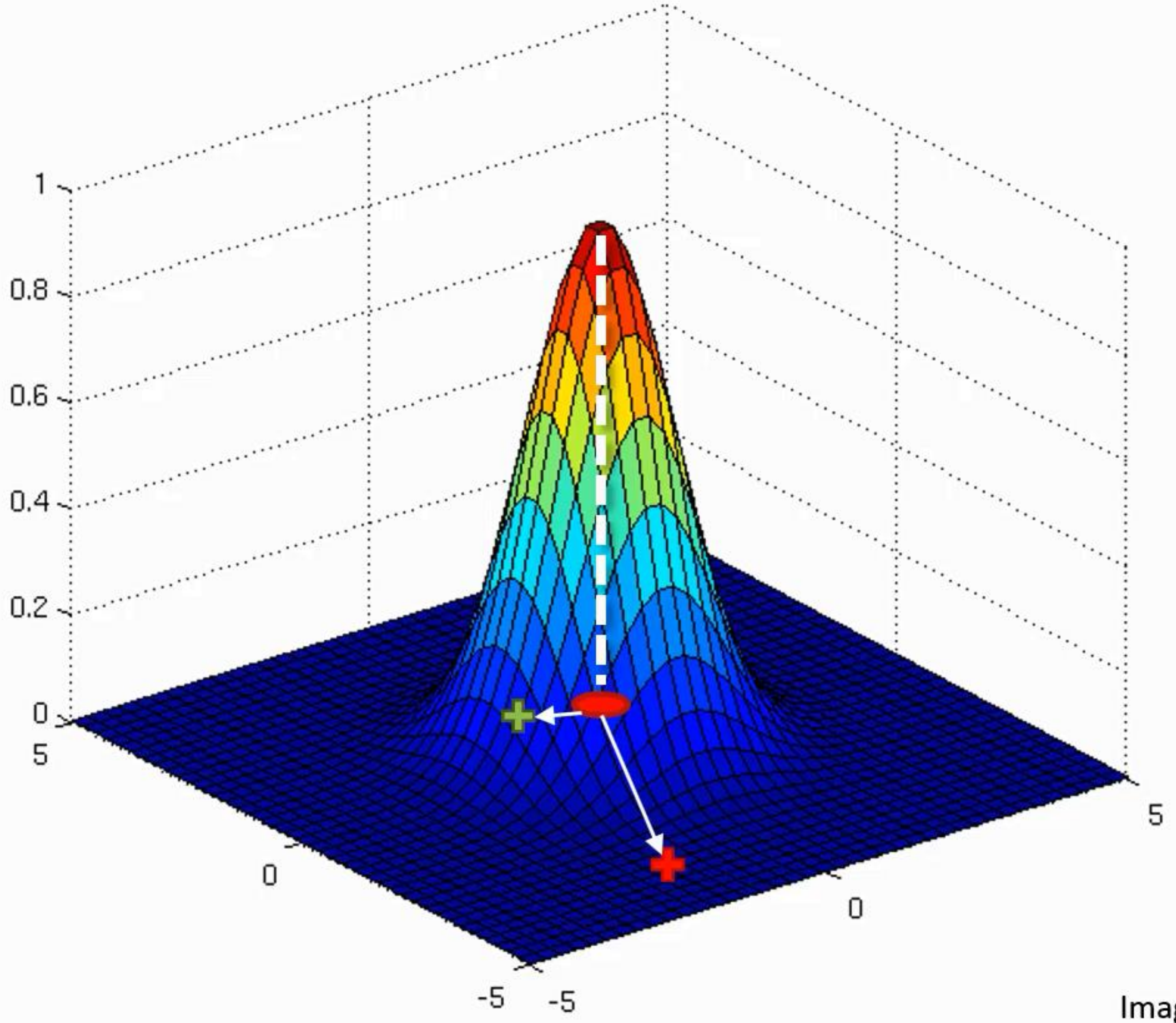
The Gaussian RBF Kernel



$$K\left(\vec{x}, \vec{l}^i\right) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$

Image source: http://www.cs.toronto.edu/~duvenaud/cookbook/index.html

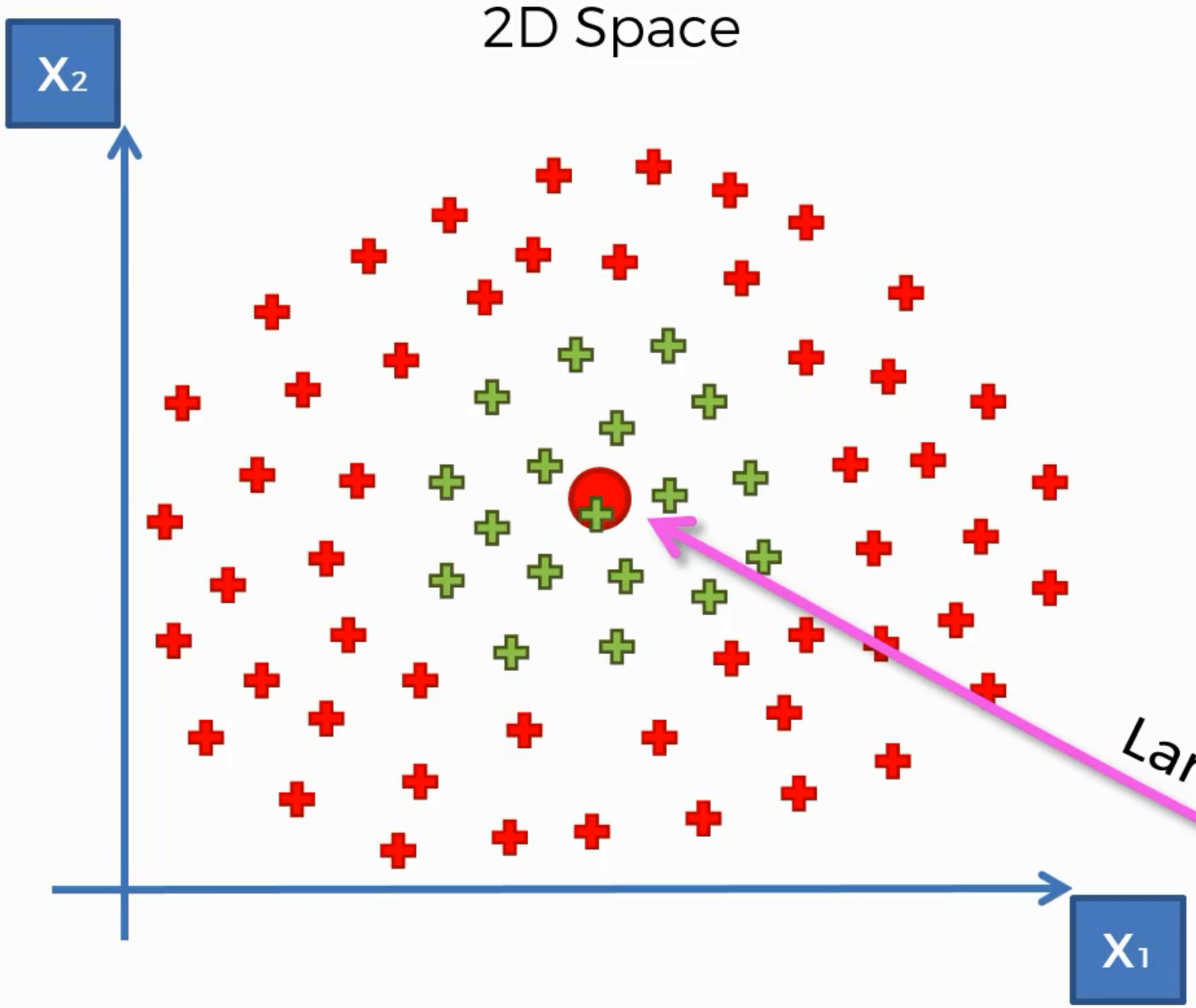
The Gaussian RBF Kernel

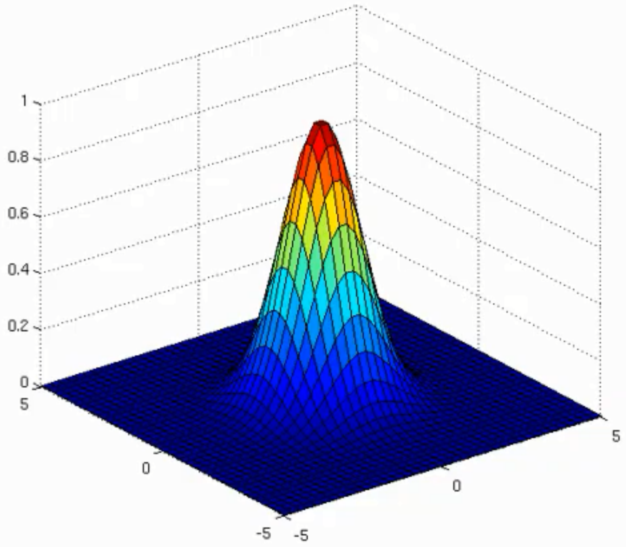


$$K\left(\vec{x}, \vec{l}^i\right) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$

Image source: http://www.cs.toronto.edu/~duvenaud/cookbook/index.html

The Gaussian RBF Kernel

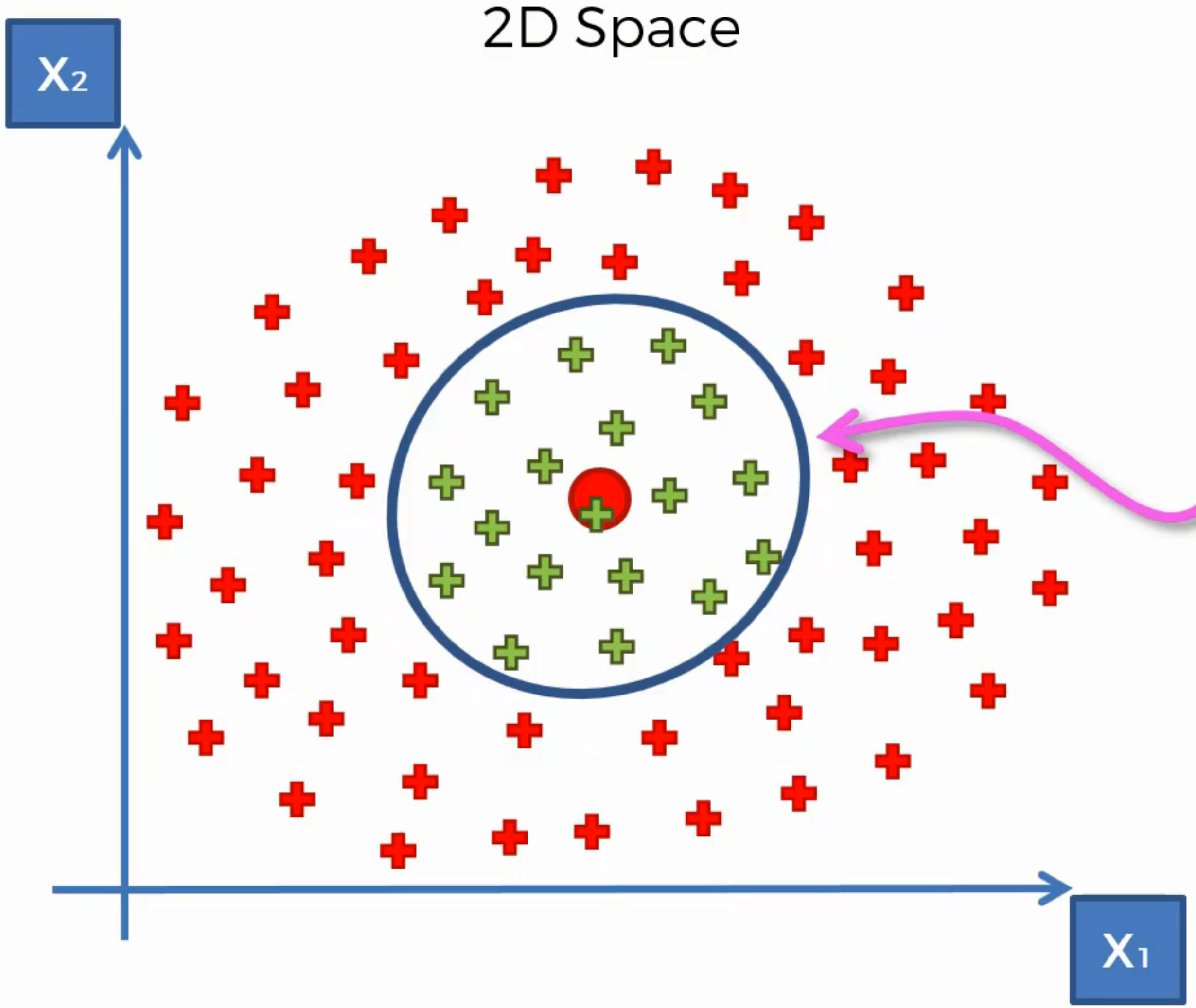


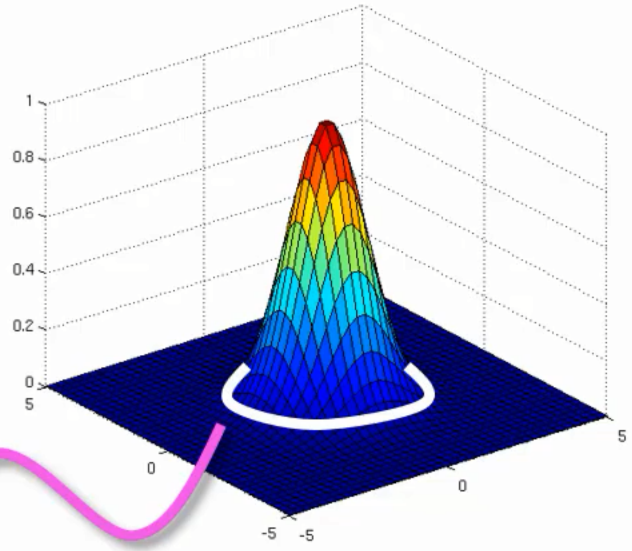


Landmark

$$K(\vec{x},\vec{l}^{i}) = e^{-\frac{\left\|\vec{x}-\vec{l}^{i}\right\|^{2}}{2\sigma^{2}}}$$

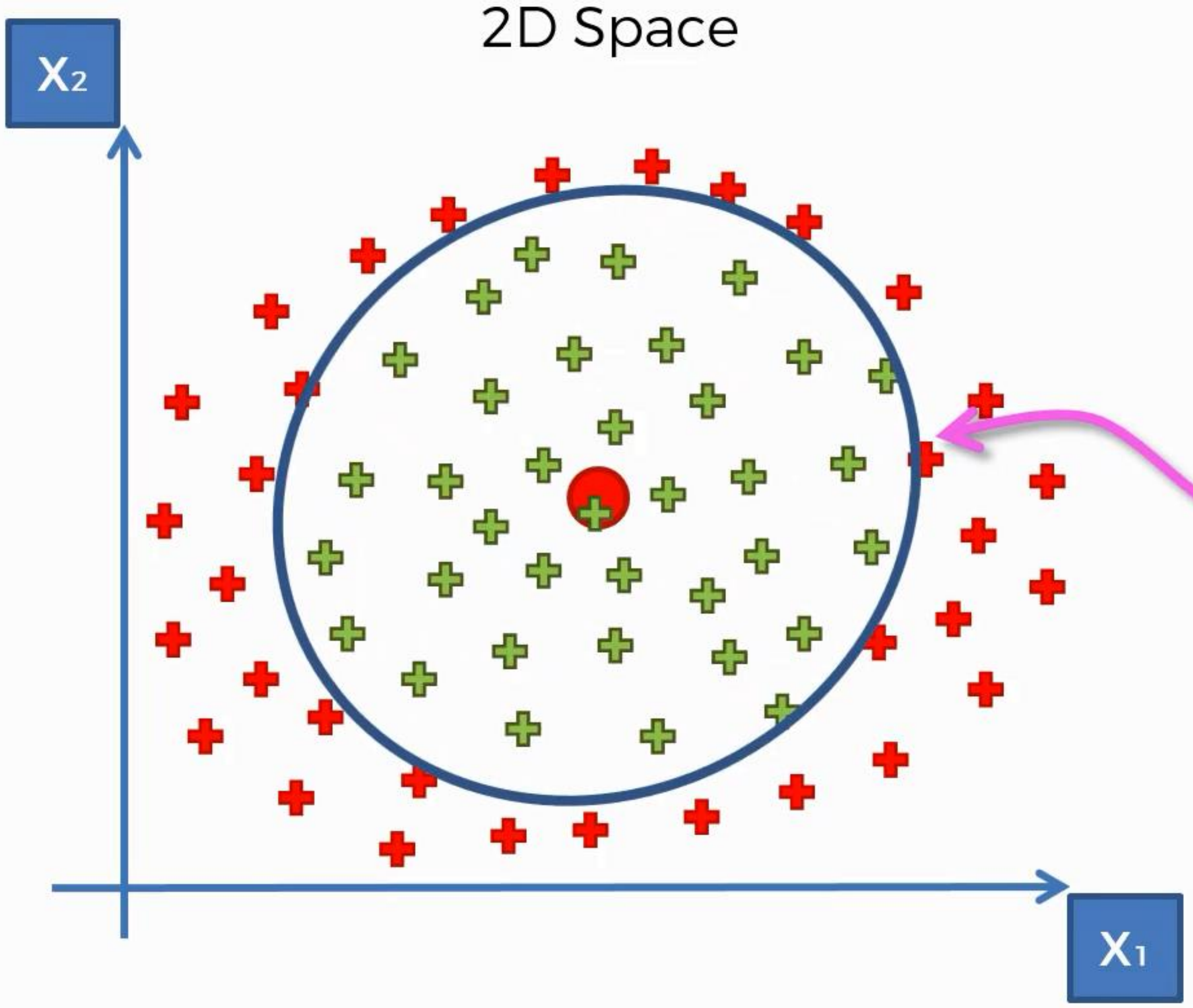
The Gaussian RBF Kernel



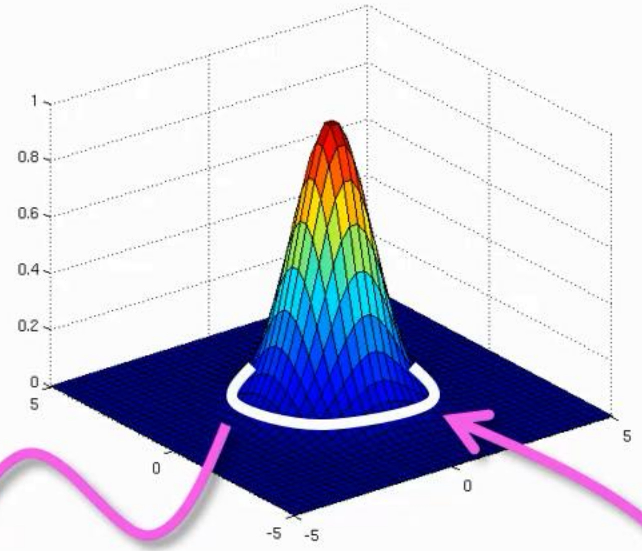


$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

The Gaussian RBF Kernel



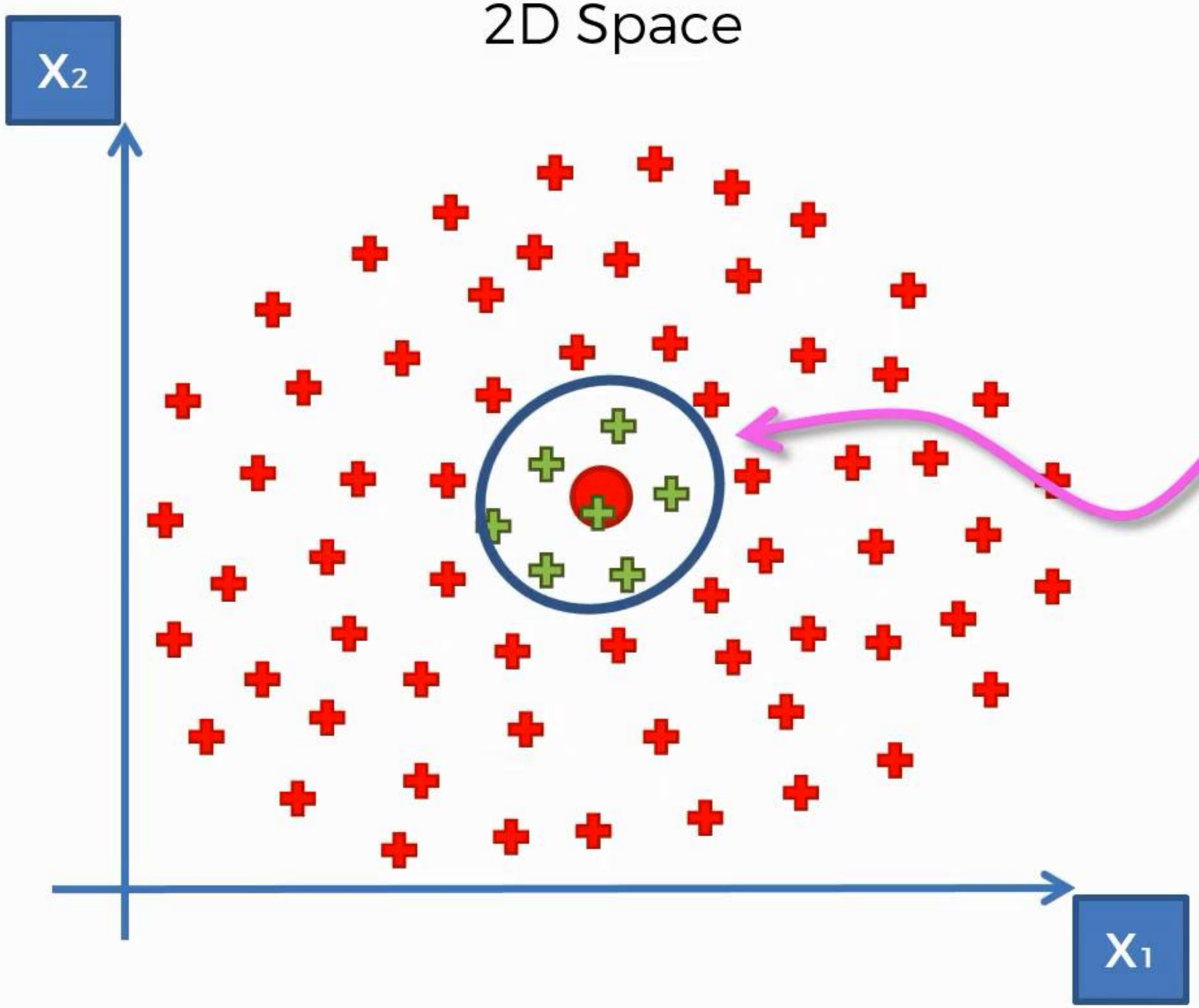
σ



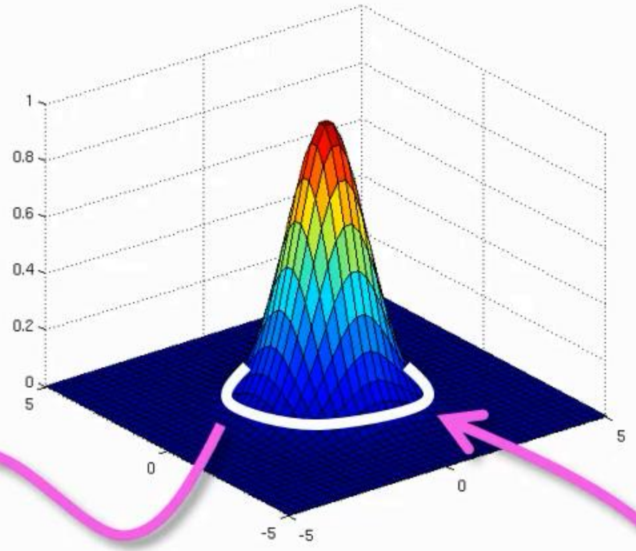
$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$

The Gaussian RBF Kernel

2D Space

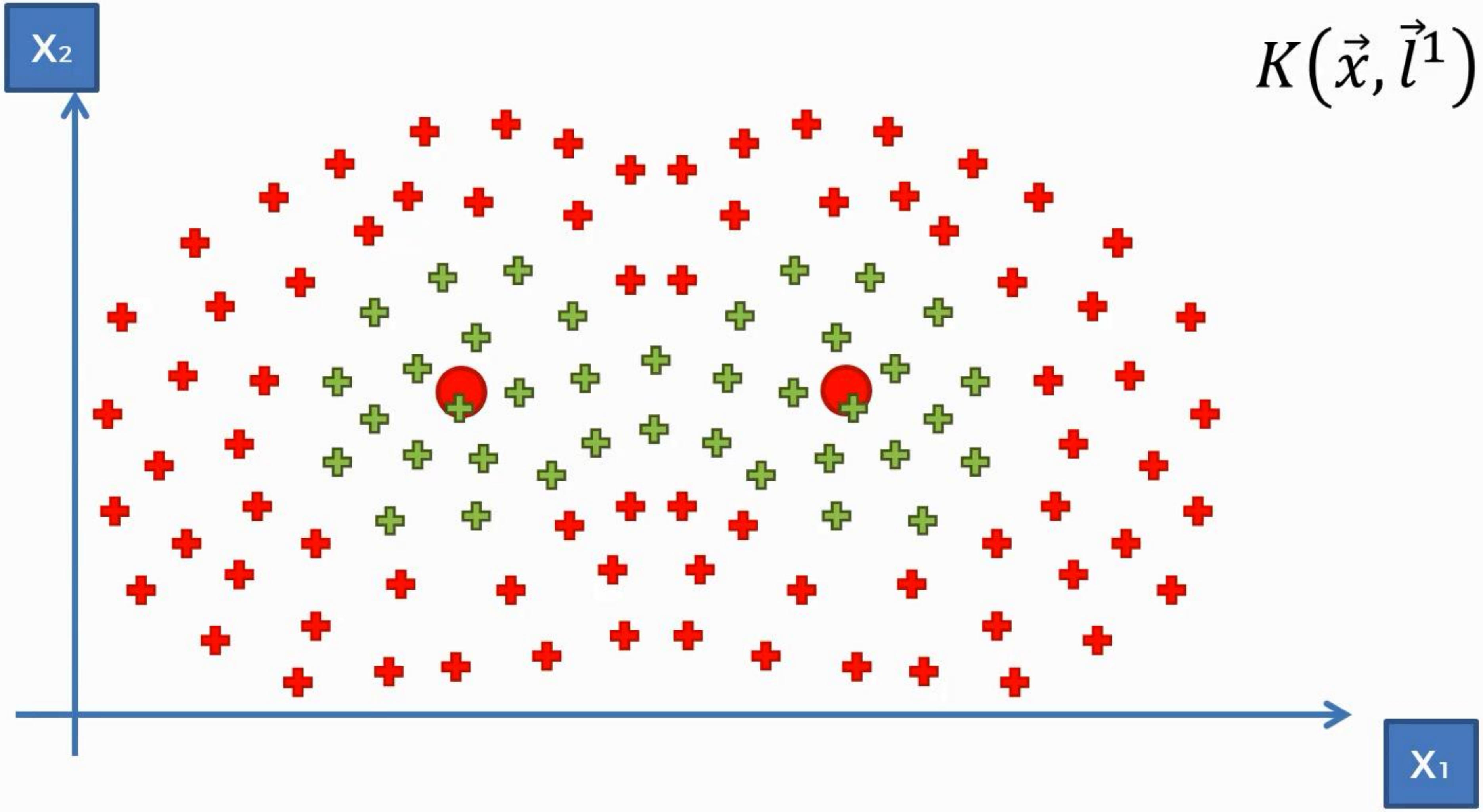


σ



$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

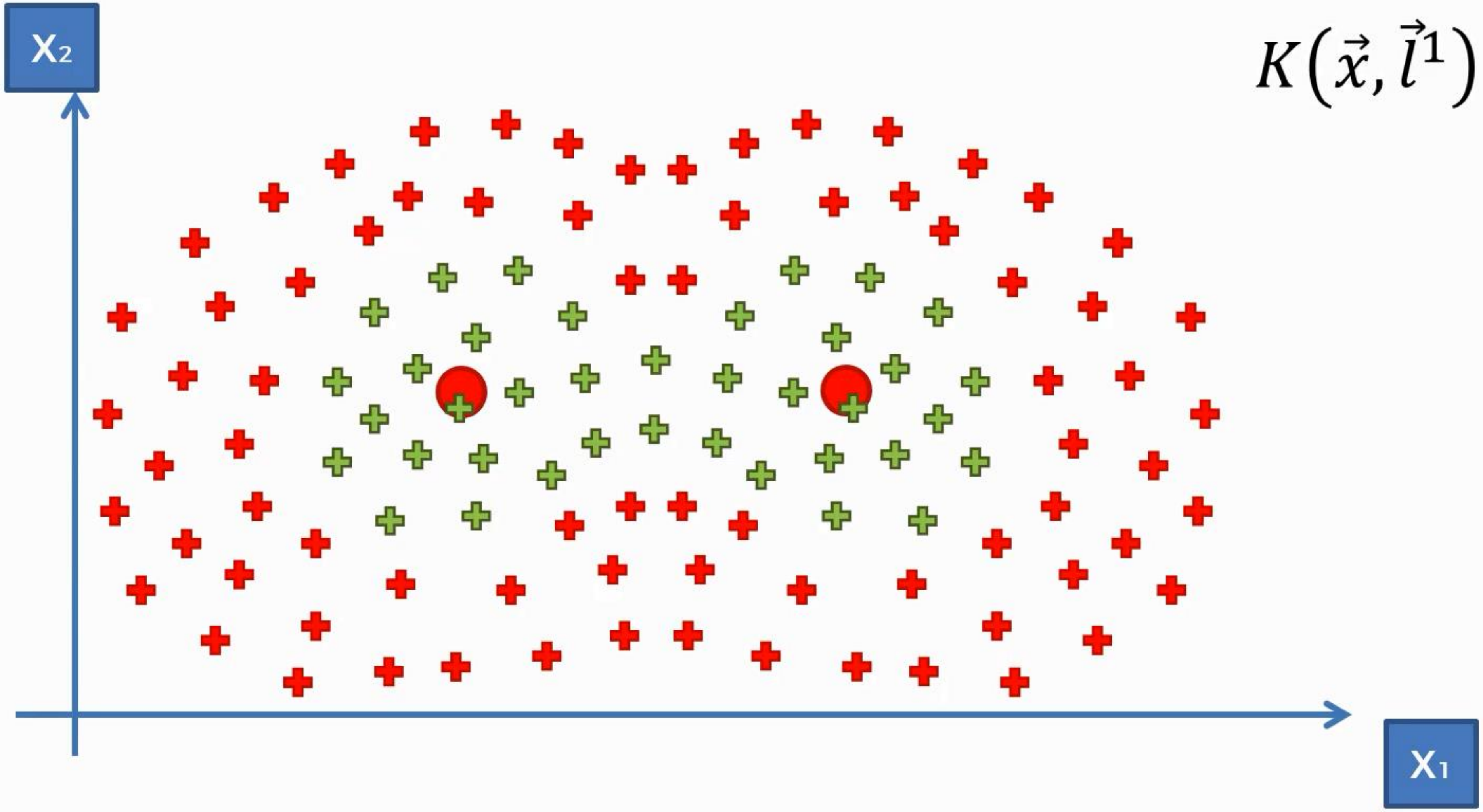
The Gaussian RBF Kernel



$$K(\vec{x}, \vec{l}^{1}) + K(\vec{x}, \vec{l}^{2})$$

(Simplified Formula)

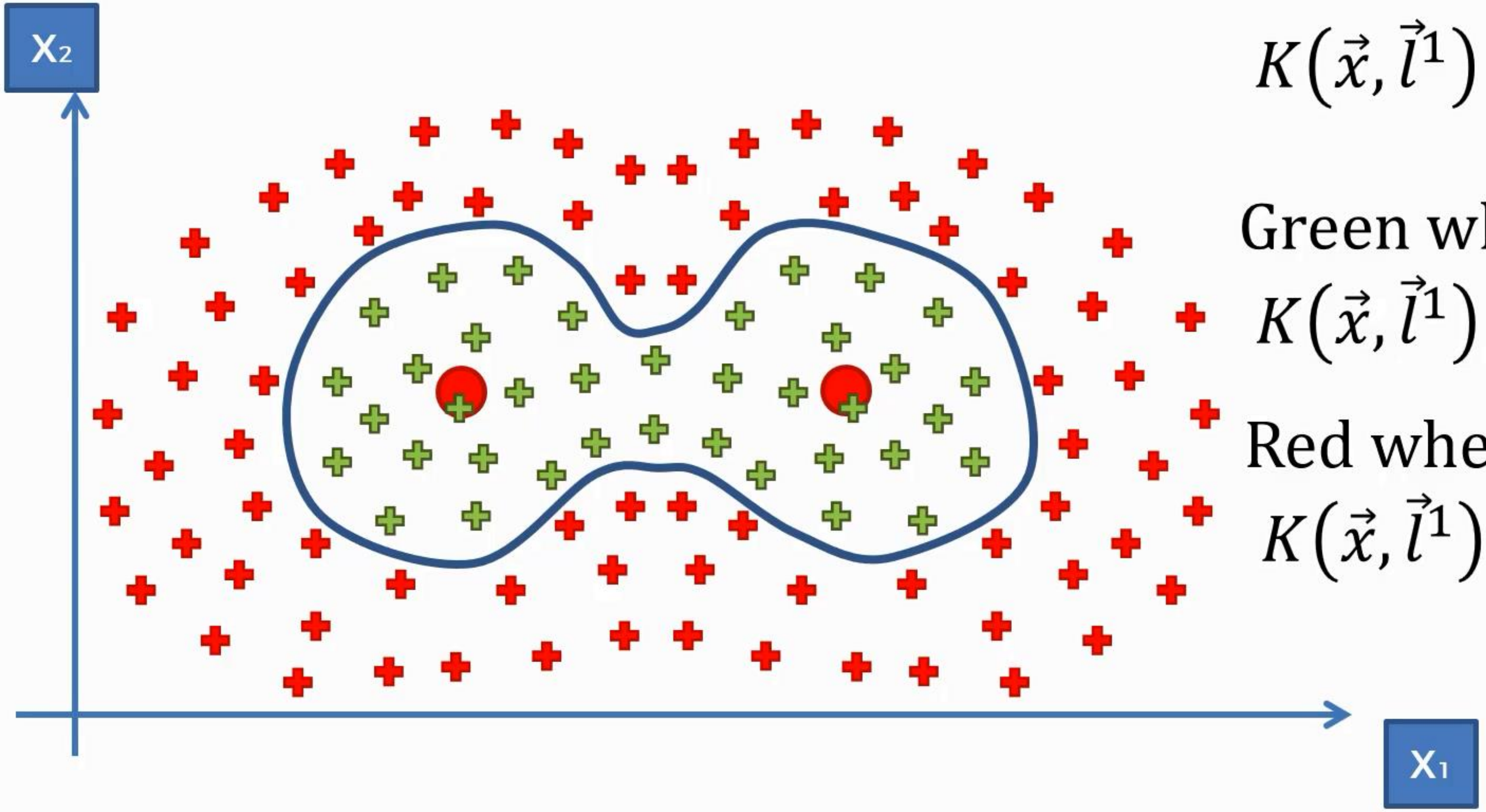
The Gaussian RBF Kernel



$$K(\vec{x}, \vec{l}^1) + K(\vec{x}, \vec{l}^2)$$

(Simplified Formula)

The Gaussian RBF Kernel



$$K(\vec{x},\vec{l}^{1}) + K(\vec{x},\vec{l}^{2})$$

(Simplified Formula)

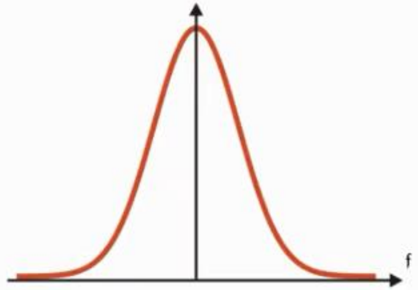
Green when:

$$K(\vec{x},\vec{l}^{1}) + K(\vec{x},\vec{l}^{2}) > 0$$

Red when:

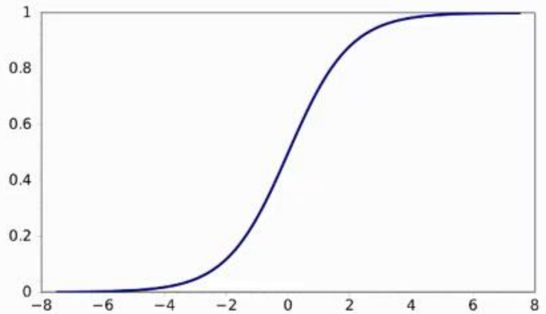
$$K(\vec{x},\vec{l}^{1}) + K(\vec{x},\vec{l}^{2}) = 0$$

Types of Kernel Functions



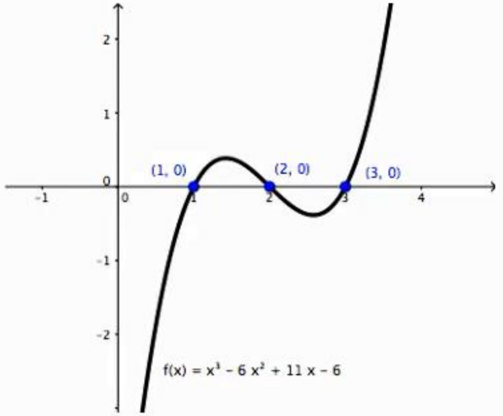
Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\left\|\vec{x}-\vec{l}^i\right\|^2}{2\sigma^2}}$$



Sigmoid Kernel

$$K(X,Y) = \tanh(\gamma \cdot X^T Y + r)$$



Polynomial Kernel

$$K(X,Y) = (\gamma \cdot X^T Y + r)^d, \gamma > 0$$

Data Preparation

- **Numerical Inputs**: SVM assumes that your inputs are numeric. If you have categorical inputs you may need to covert them to binary dummy variables (one variable for each category) using One-Hot Encoding.

- **Binary Classification**: SVM is intended for binary (two-class) classification problems.

Model Tuning

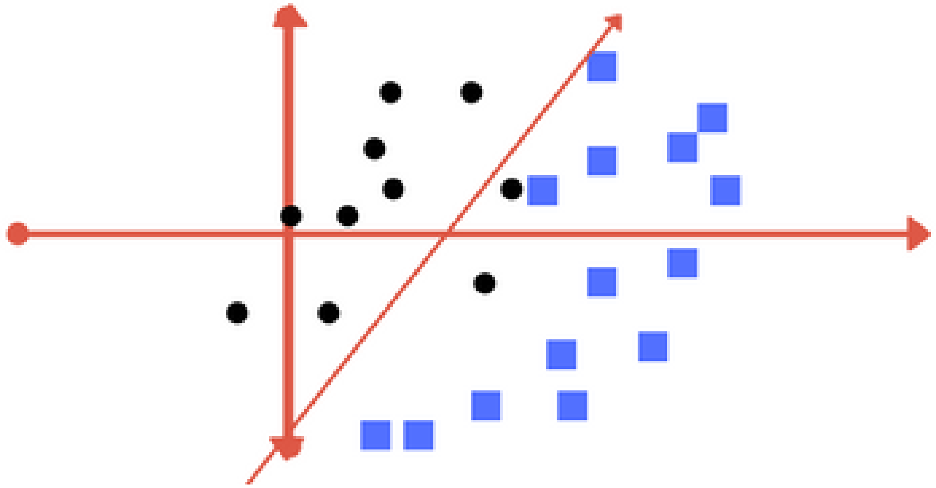
Tuning SVM Parameters to obtain better results.

Regularization (C Parameter)

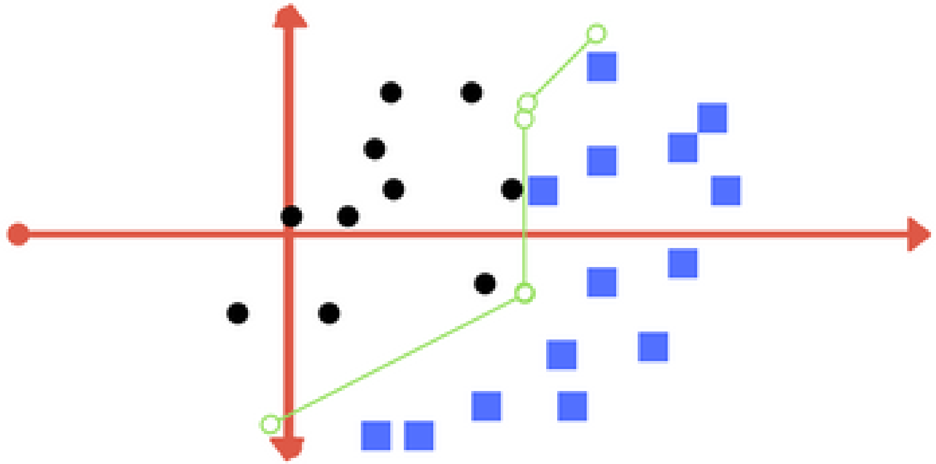
The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.

Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.



Low Regularization (C) value



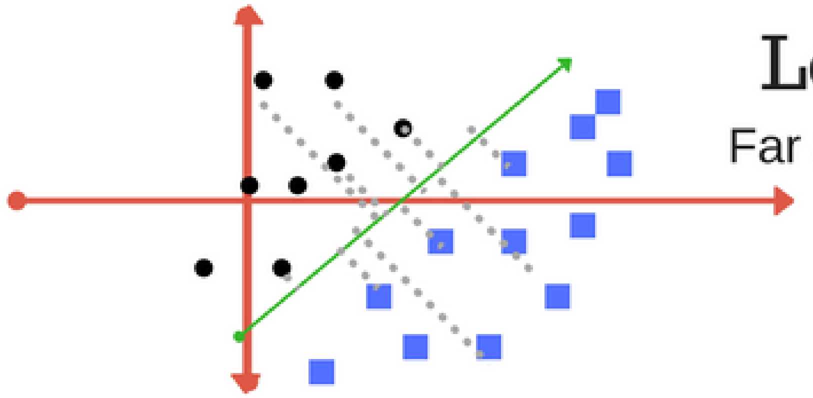
Higher Regularization (C) value

Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’.

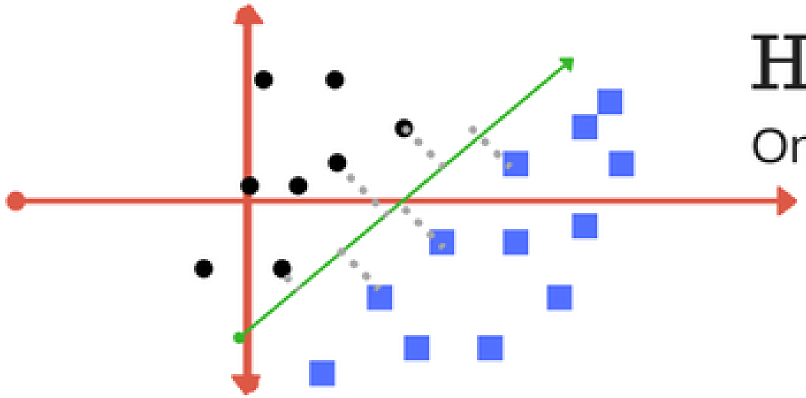
For lower gamma value, the points far away from plausible separation line are considered in calculation for the separation line.

Higher gamma means the points close to plausible line are considered in calculation.



Low Gamma

Far away points are also considered.



High Gamma

Only nearby points are considered.

Advantages

- It works really well with clear margin of separation
- It is effective in high dimensional spaces
- It is effective in cases where number of dimensions is greater than the number of samples
- It uses only a subset of training points in the decision function (called Support Vectors), so it is also memory efficient

Disadvantages

- It doesn’t perform well, when we have large data set because the required training time is higher

- It also doesn’t perform very well, when the data set has more noise i.e. target classes are overlapping

- SVM doesn’t directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

Further Reading

http://mlkernels.readthedocs.io/en/latest/kernels.html