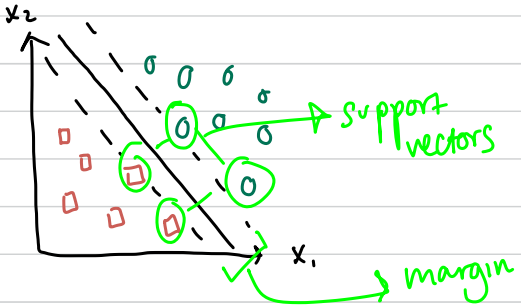
maximum margin classification using SVM

- a support vector machine (SVM) can be considered an extension of a perceptron
- in a perceptron, we aim to minimize the # of misclassifications, in SVM we aim to optimize the margin, or distance between the separating hyperplane & training examples closest to the hyperplane, which are referred to as the support vectors.



  - large margins tend to have a lower generalization error & are not prone to overfitting.

Dealing with non-linearity seperable cases using slack variables

- slack variable: often called C in SVM contexts. C is a hyperparameter for controlling the penalty for misclassification.
- soft-margin classification:

$C \to 0$ ↑bias (underfitting)
$C \to \infty$ ↑variance (overfitting)

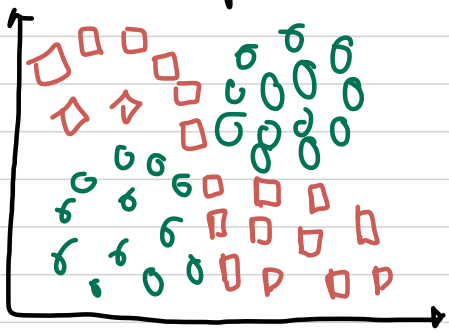
- Logistic Regression vs. SVMs:
  → in classifications, linear logistic regression & linear SVM yield similar results

Logistic Regression pros: simple model that can be easily implemented & updated, its also easier to explain mathematically.

SVM pros: not affected by outliers since it updates weights based on support vectors.

Solving non-linear problems using a kernel SVM

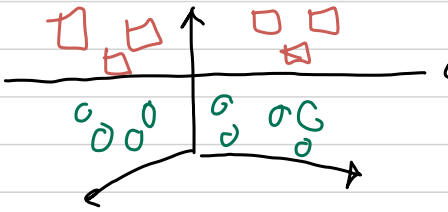
example of non-seperable data



cannot use a hyperplane to seperate the classes.

- In kernel methods, we create non-linear combinations of the original features & project them onto a higher dimensional space via a mapping function ϕ, where the data becomes linearly seperable.

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$



higher order hyperplane seperation

using the kernel trick to find seperating hyperplanes in high-Dimensional space.

- transform the data using a mapping function ϕ & train a linear SVM model.
- However, this is very computationally expensive, which is where the "trick" comes into play.
- in kernel SVM we replace the Dot product $x^{iT} x^j$ by $\phi(x^i)^T \phi(x^j)$.

$\phi(x^i)^T \phi(x^j)$ → expensive

define a function

$$K(x^i, x^j) = \phi(x^i)^T \cdot \phi(x^j)$$

Solving non-linear problems using a kernel SVM

- one of the most popular is the radial basis function (RBF) or Gaussian kernel

$$K(x^i, x^j) = \exp\left(-\frac{\|x^i - x^j\|^2}{2\sigma^2}\right) \text{ or } \exp\left(-\gamma \|x^i - x^j\|^2\right)$$

$$\gamma = \frac{1}{2\sigma^2}$$

- the kernel can be thought of as a similarity function between examples.
- the minus sign makes it such that due to the exp the similarity score will range between 1 (exactly similar) to 0 (very dissimilar)

→ γ can be thought of as a cut-off for the gaussian sphere. If γ is increased, we increase the influence of each training example (might lead to overfitting)