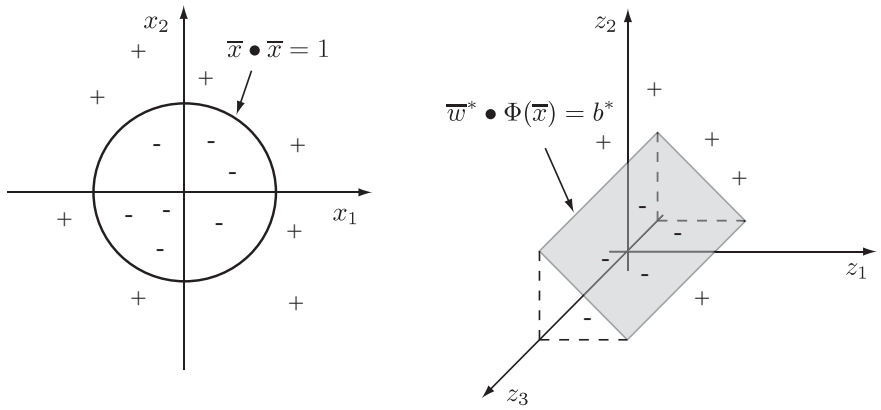
Non-Linear SVMs

- The linearity assumption on the training data is a very strong assumption.
- Not many real-world data sets are linearly separable and therefore our current setting is somewhat unrealistic.
- Turns out that the linear setting of SVMs can easily be extended to the non-linear setting by considering *kernel functions*.

A Non-Linear Data Set



a) b)

Here the figure a) represents a non-linear data set in our input space $\mathbb{R}^2$:

- There exists no linear decision surface $\overline{w} \bullet \overline{x} = b$ that would separate this data.
- The non-linear decision surface $\overline{x} \bullet \overline{x} = 1$ does separate the data.

Now, instead of computing a classification model in the input space, we first map our data set into a higher dimensional *feature space* and then compute the model,

$$\hat{f}(\overline{x}) = \text{sign}\left(\overline{w} \bullet \Phi(\overline{x}) - b\right),$$

with $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ and

$$\Phi(\overline{x}) = \Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2).$$

A Non-Linear Data Set

Observations:

- The mapping Φ maps our 2-D input space into a 3-D feature space.
- The mapping Φ converts our non-linear classification problem into a linear classification problem.
- It can be shown that all the points within the circle in input space are below the linear decision surface in feature space and all the points outside of the circle in input space are above the linear decision surface in feature space.
- We have just constructed a *non-linear decision function*!

A Non-Linear Decision Function

Given our data set and Φ we can construct a decision function that separates the non-linear data set,

$$\hat{f}(\overline{x}) = \text{sign}\left(\overline{w}^* \bullet \Phi(\overline{x}) - b^*\right).$$

with $\overline{w}^* = (w_1^*, w_2^*, w_3^*) = (1, 1, 0)$ and $b^* = 1$.

It is perhaps revealing to study this decision function in more detail,

$$\begin{aligned}
\hat{f}(\overline{x}) &= \text{sign}\left(\overline{w}^* \bullet \Phi(\overline{x}) - b^*\right) \\
&= \text{sign}\left(w_1^* x_1^2 + w_2^* x_2^2 + w_3^* \sqrt{2} x_1 x_2 - b^*\right) \\
&= \text{sign}\left(\sum_{i=1}^{3} w_i^* z_i - b^*\right),
\end{aligned}$$

where $\Phi(\overline{x}) = \Phi(x_1, x_2) = (z_1, z_2, z_3) = \overline{z}$.

We obtain a decision surface in feature space whose complexity depends on the number of dimensions of the feature space.

A Non-Linear Decision Function

We can expect that the more complex the non-linear decision surface is in the input space, the more complex the linear decision surface in feature space (the larger d),

$$\hat{f}(\overline{x}) = \text{sign}\left(\sum_{i=1}^{d} w_i^* z_i - b^*\right).$$

But, now consider the dual representation of $\overline{w}^*$,

$$\overline{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i),$$

then,

$$\begin{aligned}
\hat{f}(\overline{x}) &= \text{sign}\left(\sum_{i=1}^{d} w_i^* z_i - b^*\right) \\
&= \text{sign}\left(\overline{w}^* \bullet \overline{z} - b^*\right) \\
&= \text{sign}\left(\overline{w}^* \bullet \Phi(\overline{x}) - b^*\right) \\
&= \text{sign}\left(\sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i) \bullet \Phi(\overline{x}) - b^*\right).
\end{aligned}$$

Kernel Functions

Observations:

- We have reduced the problem from a problem in terms of feature space dimensionality to a problem that depends on the number of support vectors.
- Functions of the form $\Phi(\overline{x}) \bullet \Phi(\overline{y})$ are called kernel functions.
- In our particular case we have $\Phi(\overline{x}) \bullet \Phi(\overline{y}) = (\overline{x} \bullet \overline{y})^2$, that is the computation in feature space reduces to a computation in input space. (convince yourself of this)

Kernel Functions

If we let $k(\overline{x}, \overline{y}) = \Phi(\overline{x}) \bullet \Phi(\overline{y})$ be a kernel function, then we can write our support vector machine in terms of kernels,

$$\hat{f}(\overline{x}) = \text{sign}\left(\sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i) \bullet \Phi(\overline{x}) - b^*\right)$$

$$= \text{sign}\left(\sum_{i=1}^{l} \alpha_i^* y_i k(\overline{x}_i, \overline{x}) - b^*\right)$$

We can apply the same kind of reasoning to b^* which is the offset term in feature space,

$$b^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(\overline{x}_i) \bullet \Phi(\overline{x}_{sv+}) - 1$$

$$= \sum_{i=1}^{l} \alpha_i^* y_i k(\overline{x}_i, \overline{x}_{sv+}) - 1$$

This means, that the support vector machine in feature space is completely determined by the support vectors and an appropriate kernel function.

The fact that we are free to choose any kernel function for our model is called the *kernel trick*.

Kernel Functions

Kernel Name	Kernel Function	Free Parameters
Linear Kernel	$k(\overline{x}, \overline{y}) = \overline{x} \bullet \overline{y}$	none
Homogeneous Polynomial Kernel	$k(\overline{x}, \overline{y}) = (\overline{x} \bullet \overline{y})^d$	$d \geq 2$
Non-Homogeneous Polynomial Kernel	$k(\overline{x}, \overline{y}) = (\overline{x} \bullet \overline{y} + c)^d$	$d \geq 2, c > 0$
Gaussian Kernel	$k(\overline{x}, \overline{y}) = e^{-\frac{\lvert\overline{x}-\overline{y}\rvert^2}{2\sigma^2}}$	$\sigma > 0$