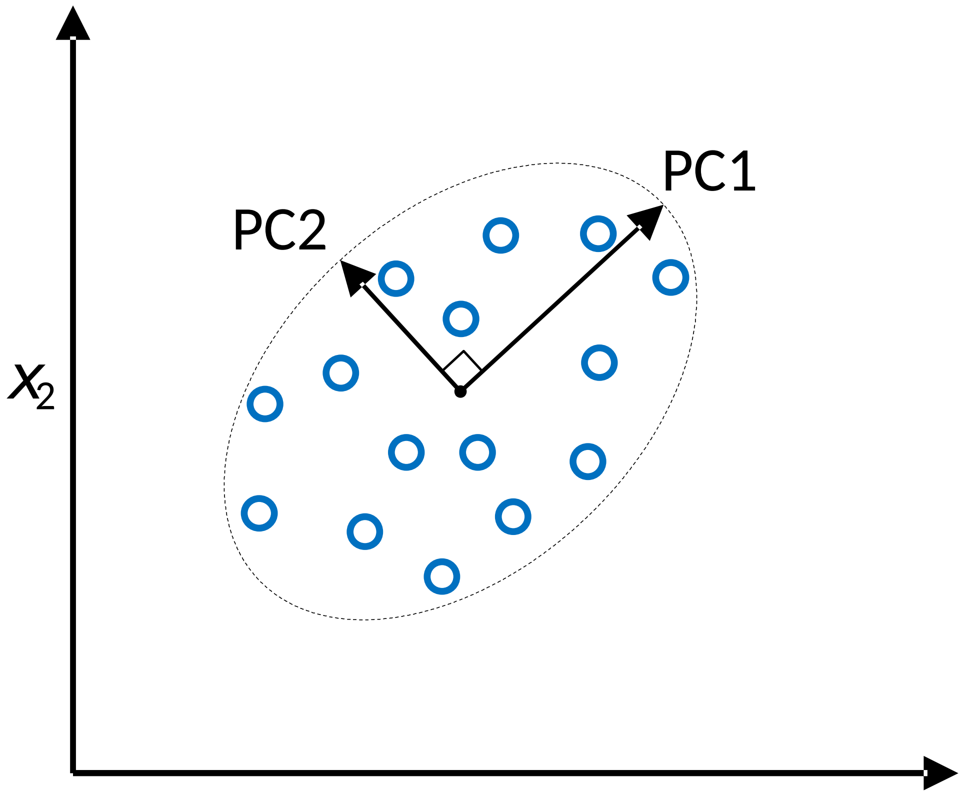
PCA+SVM

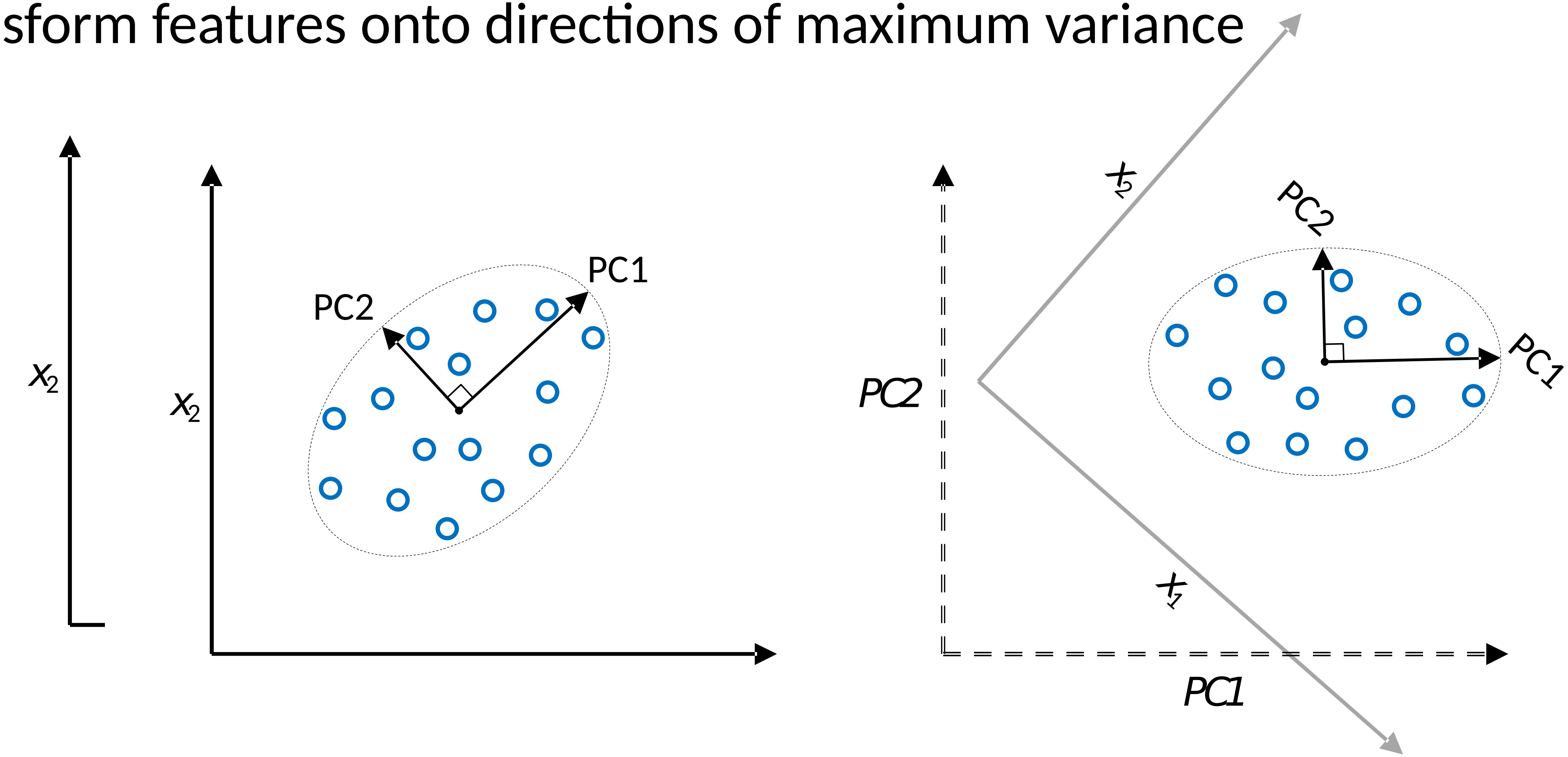
Principal Component Analysis (PCA)

Find directions of maximum variance



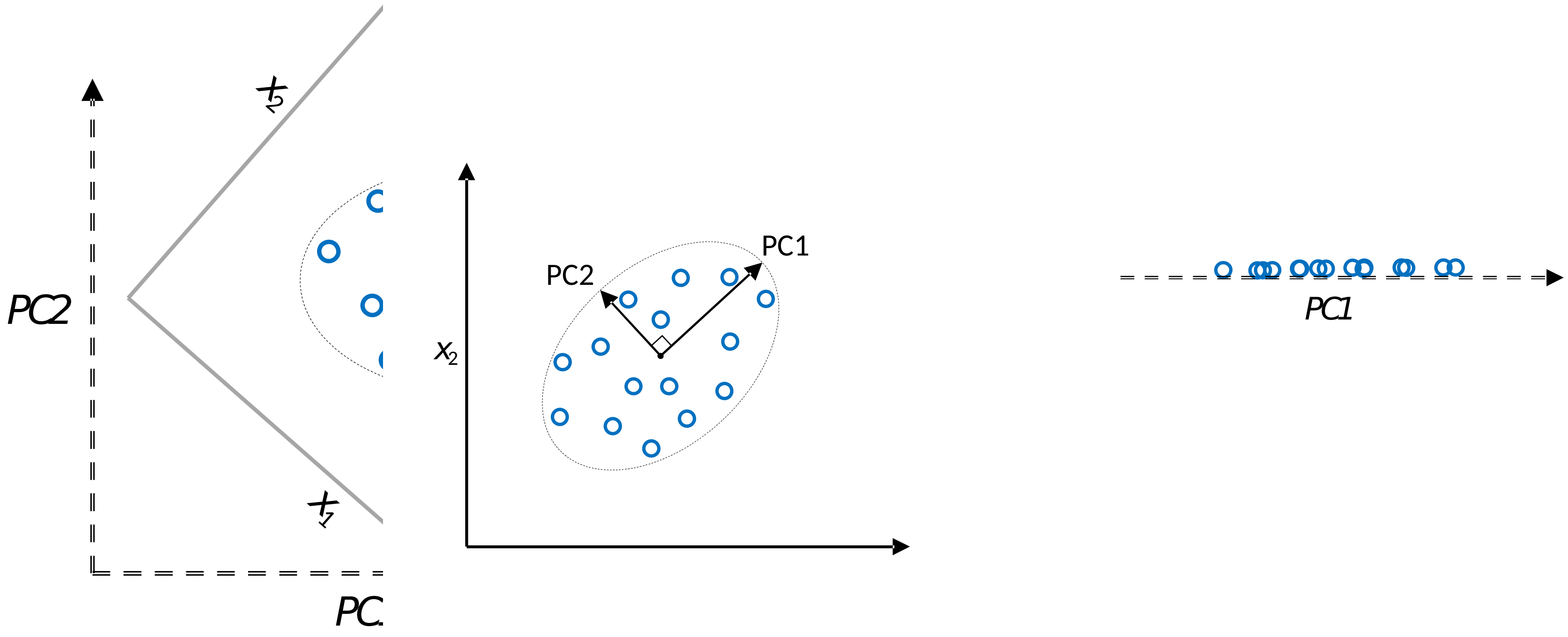
Principal Component Analysis (PCA)

Transform features onto directions of maximum variance



Principal Component Analysis (PCA)

Usually consider a subset of vectors of most variance (dimensionality reduction)



Principal Component Analysis (PCA)

Suppose N data I_i $(i=l,2,---,N)$, each data is denoted as a column vector X_i , and the dimension is M.

Mathematically, the transformation is defined by a set of *M by M* weigh matrix W that map each row vector X_i to a new vector of principal component *scores* T_i

$$\mathbf{T} = \mathbf{X}\mathbf{W}$$

To find principal components (the axes of the ellipsoid) we compute the covariance matrix of the data and calculate the eigenvalues and corresponding eigenvectors of this covariance matrix.

Principal component analysis (PCA)

The mean of the images is given by:

$$\bar{x} = \sum_{i=1}^{N} \frac{x_i}{N}$$

The covariance matrix of images is given by

$$C = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T = \frac{1}{N} X X^T$$

where $X = \left[x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_N - \bar{x}\right]$.

We break this matrix down into two separate components:

- directions of data
- magnitude (or importance) of each direction

To do so, we use Singular Value Decomposition (SVD) algorithm.

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$$

Σ is an *N*-by-*M* rectangular diagonal matrix of positive numbers called the singular values.

U is an *N*-by-*N* matrix, the columns of which are orthogonal unit vectors.

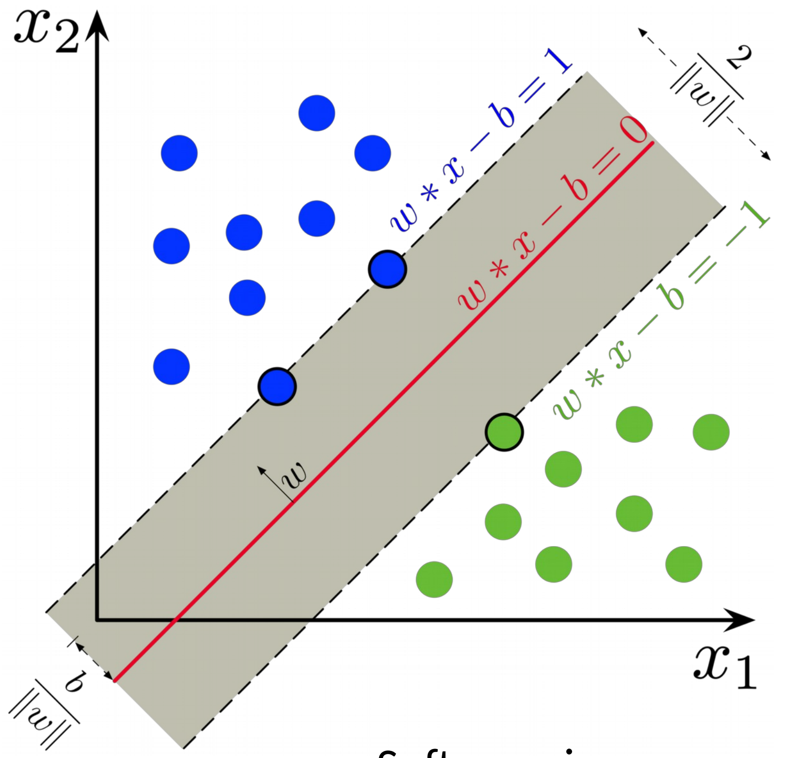
W is a *p*-by-*p* whose columns are orthogonal unit vectors.

$$\begin{aligned}\mathbf{X}^T\mathbf{X} &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{W}^T \\ &= \mathbf{W}\hat{\mathbf{\Sigma}}^2\mathbf{W}^T\end{aligned}$$

The decomposition of X to its principal components is as below:

$$\begin{aligned}\mathbf{T} &= \mathbf{X}\mathbf{W} \\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{W}^T\mathbf{W} \\ &= \mathbf{U}\boldsymbol{\Sigma}\end{aligned}$$

Support Vector Machine (SVM)



Data set:

$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n),$$

The decision border:

$$\vec{w} \cdot \vec{x} - b = 0,$$

Hard margin:

$$\vec{w} \cdot \vec{x} - b = 1$$

$$\vec{w} \cdot \vec{x} - b = -1$$

The Decision:

$$\vec{w} \cdot \vec{x}_i - b \geq 1 \quad \text{if} \quad y_i = 1$$

$$\vec{w} \cdot \vec{x}_i - b \leq -1 \quad \text{if} \quad y_i = -1$$

In sum:

$$y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n. \qquad (1)$$

Soft margin:

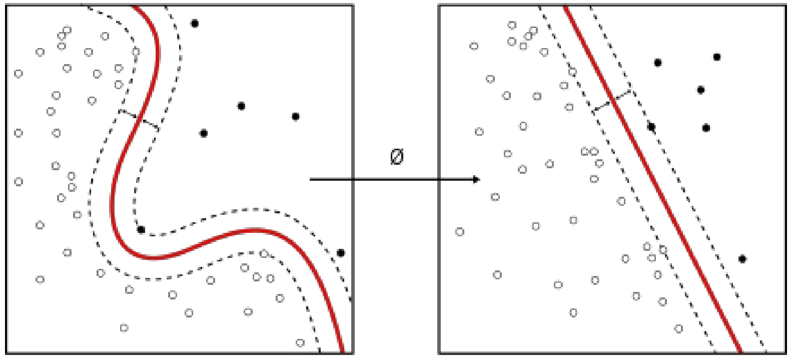
$$\max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right).$$

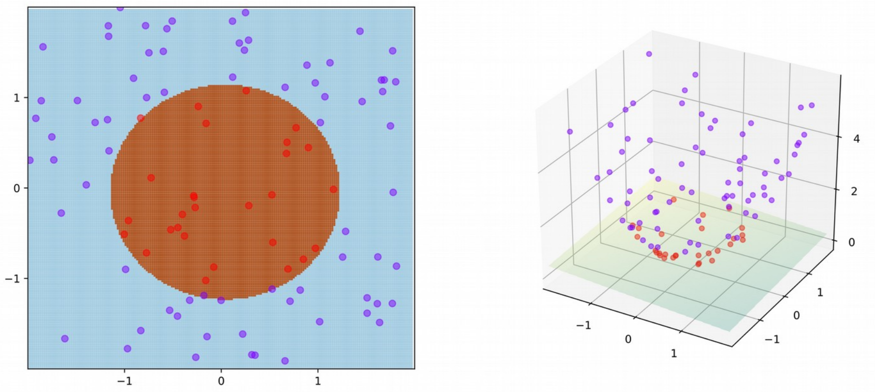
Minimize $\|\vec{w}\|$ subject to $y_i(\vec{w} \cdot \vec{x}_i - b) \geq 1$ for $i = 1, \ldots, n$

SVM classifier:

$$\vec{x} \mapsto \operatorname{sgn}(\vec{w} \cdot \vec{x} - b)$$

Applying kernels to make non-linear problems to linear problems:





Polynomial $k(\overrightarrow{x_i}, \overrightarrow{x_j}) = (\overrightarrow{x_i} \cdot \overrightarrow{x_j})^d$

Gaussian $k(\overrightarrow{x_i}, \overrightarrow{x_j}) = \exp(-\gamma \|\overrightarrow{x_i} - \overrightarrow{x_j}\|^2)$

Hyperbolic tangent $k(\overrightarrow{x_i}, \overrightarrow{x_j}) = \tanh(\kappa \overrightarrow{x_i} \cdot \overrightarrow{x_j} + c)$