Pattern Classification

EET3053

Lecture 07: Support Vector Machine

Dr. Kundan Kumar
Associate Professor
Department of ECE

Faculty of Engineering (ITER)
S'O'A Deemed to be University, Bhubaneswar, India-751030

Linear Machine: Support Vector Machine

Introduction

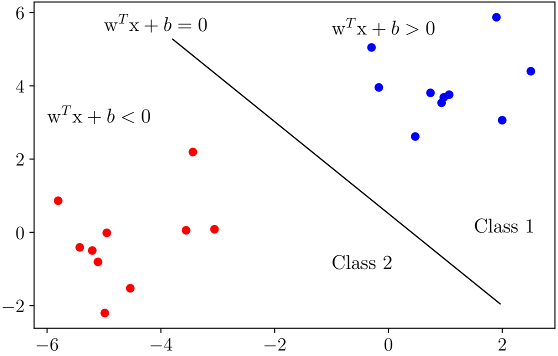
- Support vector machines (SVMs) are a linear machines initially developed for two class problems, which construct a hyperplane or set of hyperplanes in a high- or infinite-dimensional space.
- SVMs are a set of supervised learning methods used for
  - classification,
  - regression and
  - outliers detection.
- The advantages of support vector machines are:
  - Effective in high dimensional spaces.
  - Also, effective in cases where number of dimensions is greater than the number of samples.
  - Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
  - Versatile: different SVM kernels can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Introduction

- The disadvantages of support vector machines include:
  - If the number of features is much greater than the number of samples then choosing regularization to avoiding over-fitting is crucial.
  - SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called Kernel trick.
- Kernel trick implicitly maps their input into high-dimensional feature space.

Linear decision boundary

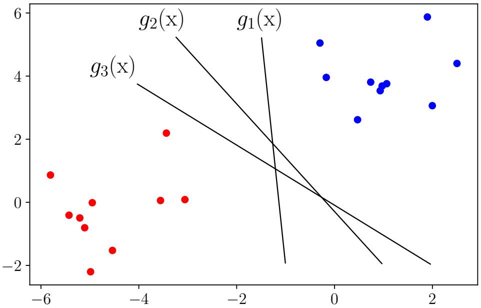
- Binary classification can be viewed as the task of separating classes in feature space using decision boundary:



$$f(x) = \text{sign}(w^T x + b)$$

What is a good Decision Boundary?

- Consider a two-class, linearly separable classification problem, many decision boundaries are possible.
- Are all decision boundaries equally good?
- Which of the linear separators is optimal?
- The perceptron algorithm can be used to find such a boundary.



Linear SVM: Objective

- Let us training data set, $\mathcal{D}$, a set of n points.

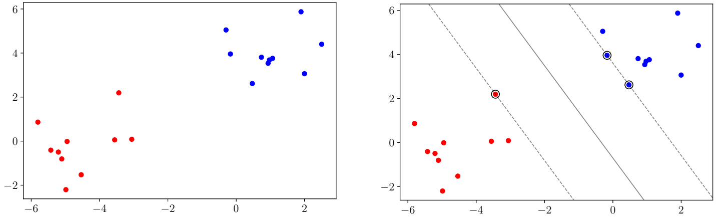
$$\mathcal{D} = \{(\mathrm{x}_i, y_i) \ \mid \ \mathrm{x}_i \in \Re^d, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

$\mathrm{x}_i \quad \rightarrow \quad d$-dimensional real vector

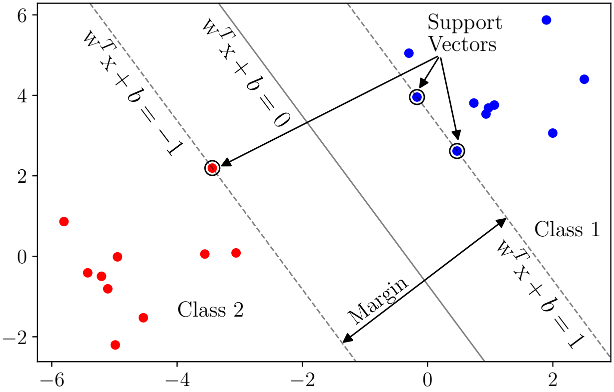
- Objective: find maximum-margin hyperplane

$$\mathrm{w}^T \mathrm{x} + b = 0$$

where w is the normal vector to the hyperplane and b is the bias/intercept.



Linear SVM: pictorial representation

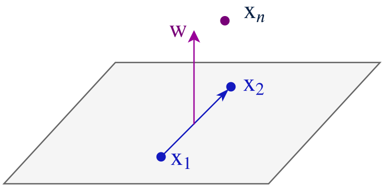


Preliminary concepts

- Let x_n be the nearest data point to the plane $\mathrm{w}^T\mathrm{x} + b = 0$.
- How far is it?
- Normalize w and b such that:

$$|\mathrm{w}^T\mathrm{x}_n + b| = 1$$

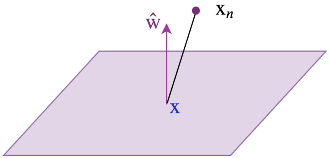
  - Now, we need to compute the distance between x_n and the plane $\mathrm{w}^T\mathrm{x} + b = 0$, where $|\mathrm{w}^T\mathrm{x}_n + b| = 1$.
  - The vector w is $\perp$ to the plane in the $\mathcal{X}$ space:
  - Take x_1 and x_2 on the plane



$$\mathrm{w}^T\mathrm{x}_1 + b = 0 \text{ and } \mathrm{w}^T\mathrm{x}_2 + b = 0 \quad \Rightarrow \quad \mathrm{w}^T(\mathrm{x}_1 - \mathrm{x}_2) = 0$$

Preliminary concepts

The distance between x_n and the plane:



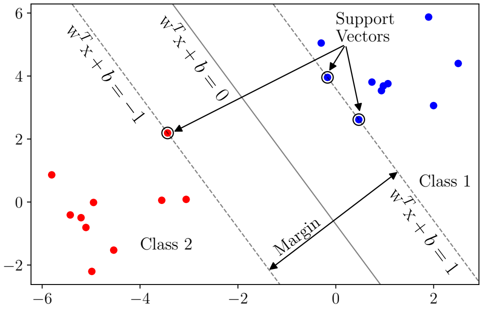
- Take any point x on the plane
- Projection of $\mathrm{x}_n - \mathrm{x}$ on $\hat{\mathrm{w}}$

$$\hat{\mathrm{w}} = \frac{\mathrm{w}}{||\mathrm{w}||}$$

$$\Rightarrow \quad \text{distance} = |\hat{\mathrm{w}}^T(\mathrm{x}_n - \mathrm{x})|$$

$$\text{distance} = \frac{1}{||\mathrm{w}||}|\mathrm{w}^T\mathrm{x}_n - \mathrm{w}^T\mathrm{x}| = \frac{1}{||\mathrm{w}||}|\mathrm{w}^T\mathrm{x}_n + b - \mathrm{w}^T\mathrm{x} - b| = \frac{1}{||\mathrm{w}||}$$

Problem formulation



- Two hyperplanes

$$\mathrm{w}^T \mathrm{x} + b = 1$$
$$\mathrm{w}^T \mathrm{x} + b = -1$$

- So the distance between the hyperplane is

$$\frac{b+1}{||\mathrm{w}||} - \frac{b-1}{||\mathrm{w}||} = \frac{2}{||\mathrm{w}||}$$

(need to be maximize)

- Therefore, $||\mathrm{w}||$ need to be minimize.

Problem formulation

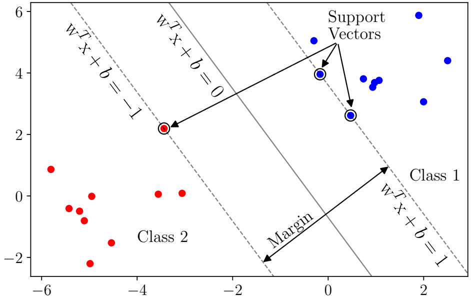
- We need to minimize $||\mathrm{w}||$ to maximize the margin.
- We also have to restrict data points from falling into the margin, so add the following constraints:
  - $\mathrm{w}^T\mathrm{x}_i + b \geq 1$ for x_i of the 1st class.
  - $\mathrm{w}^T\mathrm{x}_i + b \leq -1$ for x_i of the 2nd class.
- This can be written as

$$y_i(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, \ldots, n$$

- Combining the above two

$$\underset{\mathrm{w},b}{\text{Minimize}} \quad ||\mathrm{w}||$$

subject to $y_i(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, \ldots, n$



Problem formulation

- Problem is difficult to solve because it depends on $||\mathrm{w}||$, the norm of w, which involves a square root.
- Substitute $||\mathrm{w}||$ with $\frac{1}{2}||\mathrm{w}||^2$ (just for mathematical convenience)
- Then problem is formulated as

$$\underset{\mathrm{w},b}{\text{Minimize}} \quad \frac{1}{2}||\mathrm{w}||^2$$
$$\text{subject to} \quad y_i(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, \ldots, n$$

where $\mathrm{w} \in \Re^d$ and $b \in \Re$

- The above problem is constraint optimization problem.
- Read about Lagrangian and inequality constraint KKT

Problem solution: Lagrange formulation

- There is no direct solution of the formulated constraint optimization problem.
- To obtain the dual, take positive Lagrange multiplier α_i multiplied by each constraint and subtract from the objective function.

$$\text{Minimize} \quad \mathcal{L}(\mathrm{w}, b, \alpha) = \frac{1}{2}\mathrm{w}^T\mathrm{w} - \sum_{i=1}^{n} \alpha_i (y_i(\mathrm{w}^T \mathrm{x}_i + b) - 1)$$

w.r.t. w and b and maximize w.r.t. each $\alpha_i \geq 0$

- We can find the constraint as

$$\nabla_{\mathrm{w}} \mathcal{L} = \mathrm{w} - \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

Problem solution: Lagrange formulation

- We obtained

$$\mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i \qquad \text{and} \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Substitute in Lagrangian optimization problem,

$$\mathcal{L}(\mathrm{w}, b, \alpha) = \frac{1}{2}\mathrm{w}^T\mathrm{w} - \sum_{i=1}^{n} \alpha_i (y_i(\mathrm{w}^T\mathrm{x}_i + b) - 1)$$

we get

$$\mathcal{L}(\alpha) = \sum_{n=1}^{n} \alpha_n - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathrm{x}_i^T \mathrm{x}_j$$

Maximize w.r.t. to α subject to $\alpha_i \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

The solution - quadratic programming

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T \begin{bmatrix} y_1 y_1 x_1^T x_1 & y_1 y_2 x_1^T x_2 & \cdots & y_1 y_n x_1^T x_n \\ y_2 y_1 x_2^T x_1 & y_2 y_2 x_2^T x_2 & \cdots & y_2 y_n x_2^T x_n \\ \vdots & \vdots & \ddots & \vdots \\ y_n y_1 x_n^T x_1 & y_n y_2 x_n^T x_2 & \cdots & y_n y_n x_n^T x_n \end{bmatrix} \alpha + \left(-1^T\right) \alpha$$

subject to $\mathrm{y}^T \alpha = 0$ and $0 \leqslant \alpha \leqslant \infty$

QP hand us α

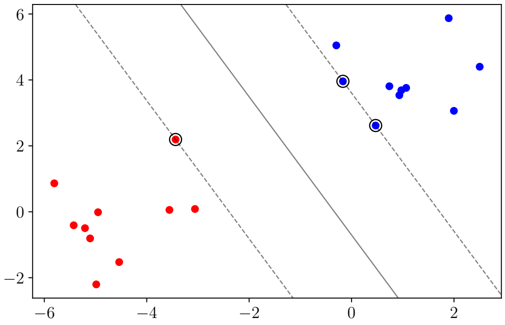
- Solution: $\alpha = \alpha_1, \ldots, \alpha_n$

$$\Rightarrow \mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i$$

- KKT condition: For $i = 1, \ldots, n$

$$\alpha_i (y_i (\mathrm{w}^T \mathrm{x}_i + b) - 1) = 0$$

- For non-zero value of α ($\alpha_n > 0$), x_n are support vectors.



Support vectors

- Closest x_i's to the plane achieve the margin

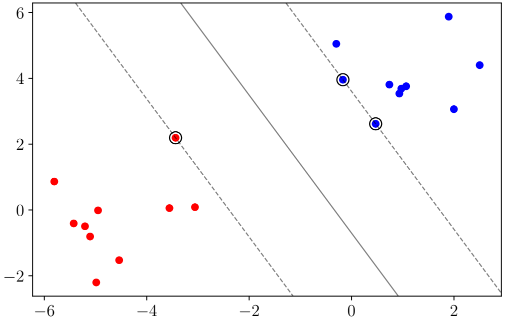
$$\Rightarrow y_i(\mathrm{w}^T \mathrm{x_i} + b) = 1$$

- We have the weight vector

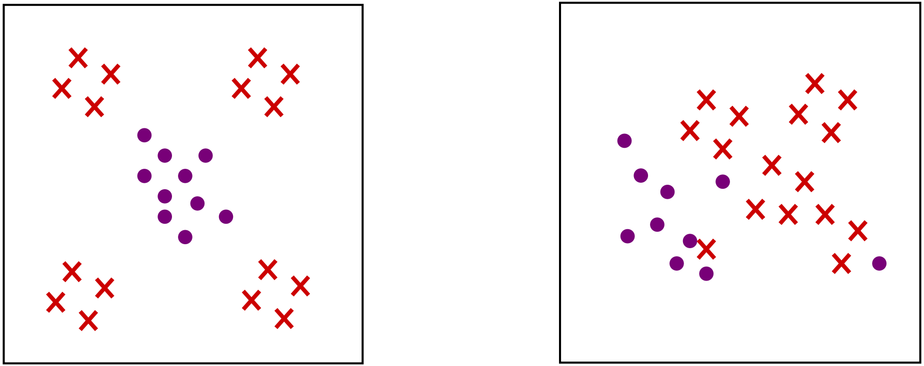
$$\mathrm{w} = \sum_{x_i \text{ is SV}} \alpha_i y_i \mathrm{x}_i$$

- Solve for b: using any Support vector (SV):

$$y_i(\mathrm{w}^T \mathrm{x}_i + b) = 1$$



Non-separable features



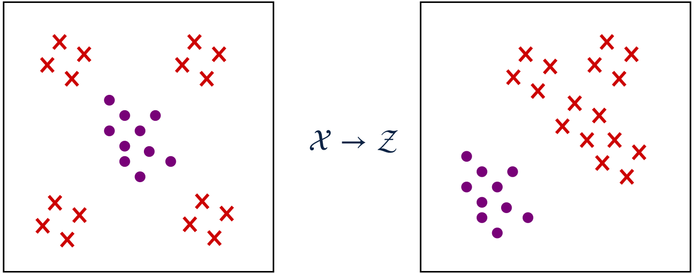
Kernel Trick

Kernel trick: z instead of x

- Dual problem:

$$\mathcal{L}(\alpha) = \sum_{n=1}^{n} \alpha_n - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathrm{z}_i^T \mathrm{z}_j$$

Maximize w.r.t. to α subject to $\alpha_i \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$



Kernel Trick: What do we need from the $\mathcal{Z}$ space?

$$\mathcal{L}(\alpha) = \sum_{n=1}^{n} \alpha_n - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathrm{z}_i^T \mathrm{z}_j$$

Constraints: $\alpha \geq 0$ for $i = 1, \ldots, n$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$

$$g(\mathrm{x}) = \mathrm{sign}(\mathrm{w}^T \mathrm{z} + b)$$ need $\mathrm{z}_i^T \mathrm{z}$

where

$$\mathrm{w} = \sum_{\mathrm{z}_i \text{ is SV}} \alpha_i y_i z_i$$

and b:

$$y_j(\mathrm{w}^T \mathrm{z}_j + b) = 1$$ need $\mathrm{z}_i^T \mathrm{z}_j$

Kernel Trick: generalized inner product

- Given two points x and $\mathrm{x}' \in \mathcal{X}$, we need $\mathrm{z}^T\mathrm{z}'$.
- Let $\mathrm{z}^T\mathrm{z}' = K(\mathrm{x}, \mathrm{x}')$ (the kernel: inner product of x and x')
- Example: $\mathrm{x} = (x_1, x_2)^T \rightarrow$ 2nd-order Φ

$$\mathrm{z} = \Phi(\mathrm{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2)$$

$$K(\mathrm{x}, \mathrm{x}') = \mathrm{z}^T\mathrm{z}' = 1 + x_1x_1' + x_2x_2' + x_1^2{x'}_1^2 + x_2^2{x'}_2^2 + x_1{x'}_1x_2{x'}_2$$

Kernel Trick

- Can we compute $K(\mathrm{x}, \mathrm{x}')$ without transforming x and x'?
- Consider:

$$\begin{aligned} K(\mathrm{x}, \mathrm{x}') &= (1 + \mathrm{x}^T \mathrm{x}')^2 = (1 + x_1 {x'}_1 + x_2 {x'}_2)^2 \\ &= 1 + x_1^2 {x'}_1^2 + x_2^2 {x'}_2^2 + 2x_1 {x'}_1 + 2x_2 {x'}_2 + 2x_1 {x'}_1 x_2 {x'}_2 \end{aligned}$$

- This is the inner production of

$$(1, x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2)$$

$$(1, {x'}_1^2, {x'}_2^2, \sqrt{2}{x'}_1, \sqrt{2}{x'}_2, \sqrt{2}{x'}_1 {x'}_2)$$

Non-linear Kernels

- Following are some basic non-linear kernels:
  - Linear:

$$K(\mathrm{x}_i, \mathrm{x}_j) = \mathrm{x}_i^T \mathrm{x}_j$$

  - Polynomial:

$$K(\mathrm{x}_i, \mathrm{x}_j) = (\gamma \mathrm{x}_i^T \mathrm{x}_j + r)^d, \gamma > 0$$

  - Radial basis function:

$$K(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(-\gamma \|\mathrm{x}_i - \mathrm{x}_j\|^2\right), \gamma > 0$$

  - Sigmoid:

$$K(\mathrm{x}_i, \mathrm{x}_j) = \tanh\left(\gamma \mathrm{x}_i^T \mathrm{x}_j + r\right), \gamma > 0$$

  where, γ, r, and d are kernel parameters.
- These kernels were used in various application where radial basis function (RBF) kernel is widely adopted as a non-linear kernel due to its capability of mapping the feature vectors from input feature space to infinite dimensional space to handle highly non-linear feature distribution.

Kernel formulation of SVM

- Remember quadratic programming?
- The only difference in quadratic coefficients as:

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T \begin{bmatrix} y_1y_1z_1^Tz_1 & y_1y_2z_1^Tz_2 & \cdots & y_1y_nz_1^Tz_n \\ y_2y_1z_2^Tz_1 & y_2y_2z_2^Tz_2 & \cdots & y_2y_nz_2^Tz_n \\ \vdots & \vdots & \ddots & \vdots \\ y_ny_1z_n^Tz_1 & y_ny_2z_n^Tz_2 & \cdots & y_ny_nz_n^Tz_n \end{bmatrix} \alpha + \left(-1^T\right)\alpha$$

subject to $\mathrm{y}^T\alpha = 0$ and $0 \leqslant \alpha \leqslant \infty$

The final hypothesis

- Express $g(\mathrm{x}) = \text{sign}(\mathrm{w}^T \mathrm{z} + b)$ in terms of $K(\_, \_)$

$$\mathrm{w} = \sum_{z_n \text{ in SV}} \alpha_n y_n \mathrm{z}_n \quad \Rightarrow \quad g(\mathrm{x}) = \text{sign}\left(\sum_{\alpha_n > 0} \alpha_n y_n K(\mathrm{x}_n, \mathrm{x}) + b\right)$$

where

$$b = y_j - \sum_{\alpha_i > 0} \alpha_i y_i K(x_i, x_j)$$

for any support vector $(\alpha_i > 0)$

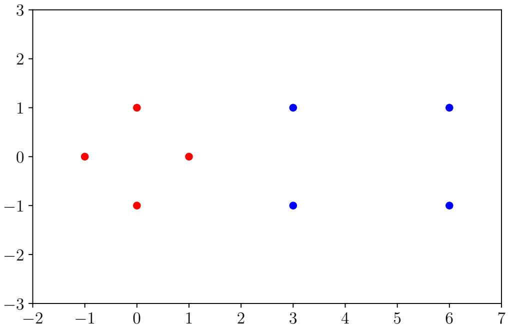
Problem to be solved: Linear (trivial problem)

- Suppose we are given the following positively labeled data points in $\Re^2$:

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

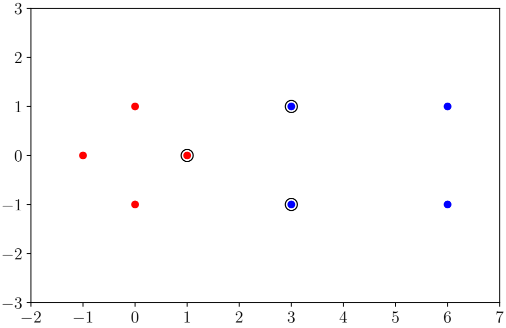
- and the following negatively labeled data points in $\Re^2$

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$



Solution

- Since the data is linear separable, we can use a linear SVM.
- By inspection, it should be obvious that there are three support vectors.



SVM: Soft Margin Formulation

Soft Margin Classification

- In basic SVM, the optimization problem is formulated for margin maximization when the feature vectors are linearly separable.
- However, a greater margin can be achieved by allowing classifier for some misclassification error during training itself.
- After allowing the misclassification of some features, the inequality constraint in basic SVM is replaced with $y_i(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 - \xi_i$, where $\xi_i \geq 0$ are slack variables.

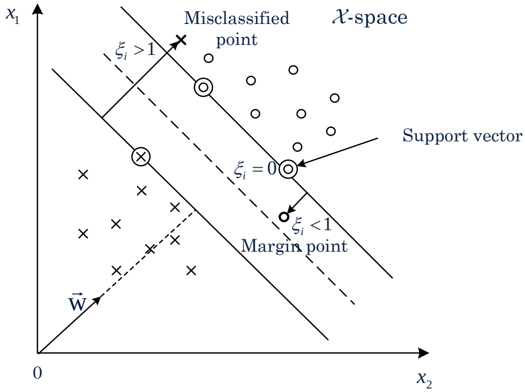


Figure: $\mathcal{X}$-space with support vector, penalized misclassification, and margin error

The new optimization problem: C-SVM

- Slack variables ξ_i can be added to allow misclassification of difficult or noisy examples, resulting margin called soft.
- Slack variables account for the misclassification and margin errors.
- The primal optimization problem with penalized misclassification and margin error becomes.

$$\begin{aligned} \underset{\mathrm{w},b}{\text{minimize}} \quad & \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to}: \quad & y_i(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i + b) \geq 1 - \xi_i, \text{ and} \\ & \xi_i \geq 0, \; i = 1, 2, \ldots, n, \end{aligned} \tag{1}$$

- where C is a regularization parameter which sets the trade-off between margin maximization and minimizing the amount of slack (misclassifications and margin error).

Lagrange formulation

Using Lagrange multipliers, the dual problem is expressed in terms of Lagrangian coefficients as

$$\mathcal{L}(\mathrm{w}, b, \xi, \alpha, \beta) = \frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\mathrm{w}^T\mathrm{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\beta_i\xi_i$$

Minimize w.r.t. w, b, and ξ and maximize w.r.t. each $\alpha_n \geq 0$ and $\beta_n \geq 0$

$$\nabla_{\mathrm{w}} L = \mathrm{w} - \sum_{i=1}^{n}\alpha_i y_i \mathrm{x}_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

and the solution is ...

$$\text{Maximize} \quad \mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathrm{x}_i^T \mathrm{x}_j \text{ w.r.t. to } \alpha$$

$$\text{subject to } 0 \leqslant \alpha_i \leqslant C \text{ for } n = 1, \ldots, N \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\Rightarrow \quad \mathrm{w} = \sum_{i=1}^{n} \alpha_i y_i \mathrm{x}_i$$

$$\text{minimize} \qquad \frac{1}{2} \mathrm{w}^T \mathrm{w} + C \sum_{i=1}^{n} \xi_i$$

References

[1] Hart, P. E., Stork, D. G., & Duda, R. O. (2000). Pattern classification. Hoboken: Wiley.

[2] Gose, E. (1997). Pattern recognition and image analysis.

Thank you!