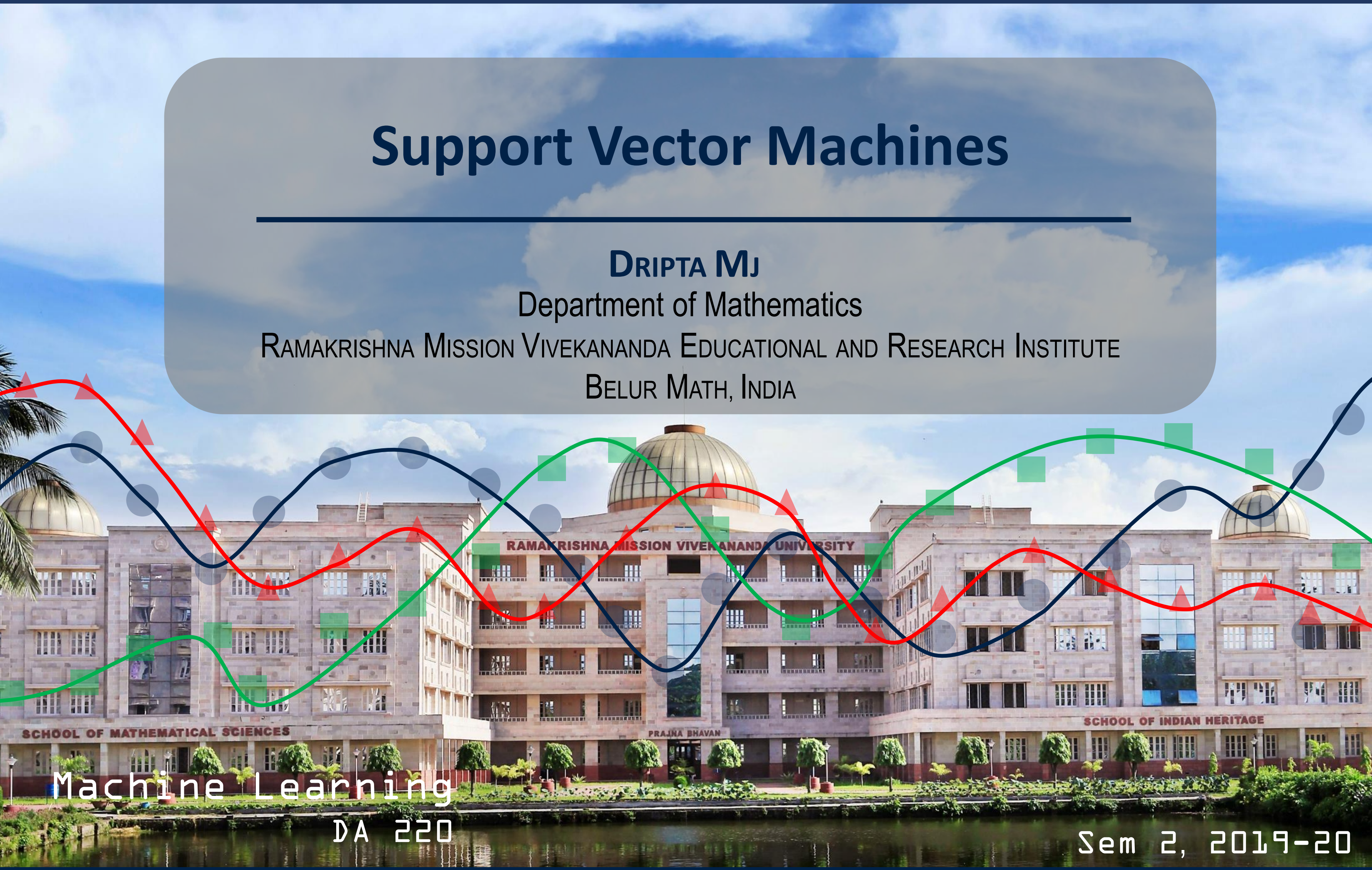
Support Vector Machines

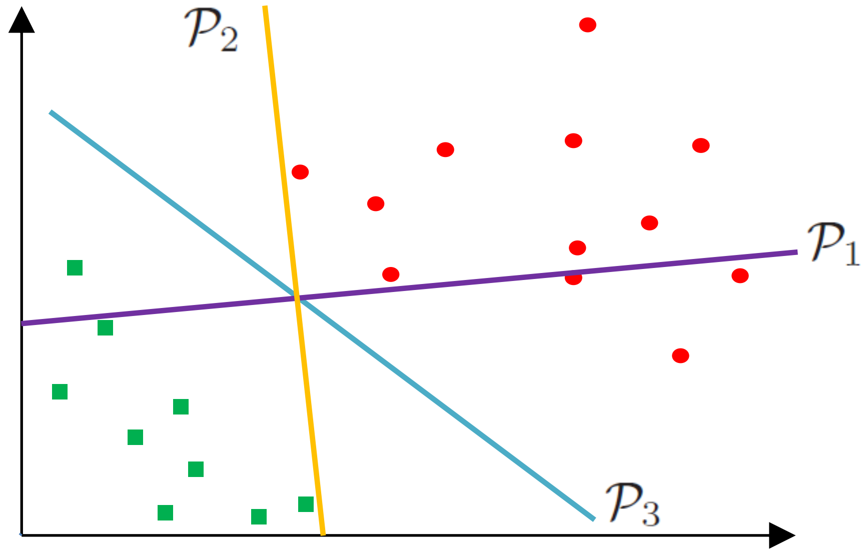
Dripta Mj
Department of Mathematics
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur Math, India



Machine Learning
DA 220

Sem 2, 2019-20

Introduction



- Find a hyperplane that separates the classes.
  - $\mathcal{P}_1$ does not separate the classes.
- Many hyperplanes are possible that separates the classes.
  - $\mathcal{P}_2$ separates the classes but with small separation between them.
  - $\mathcal{P}_3$ also separates the classes with large separation.

Margin

- Geometric margin γ_n is the perpendicular distance from the point $\mathbf{x}^{(n)}$ to the hyperplane

$$\gamma_n = y^{(n)} \left(\frac{\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(n)} + w_0}{||\mathbf{w}||} \right)$$

- Margin is defined as the minimum of the geometric margin.

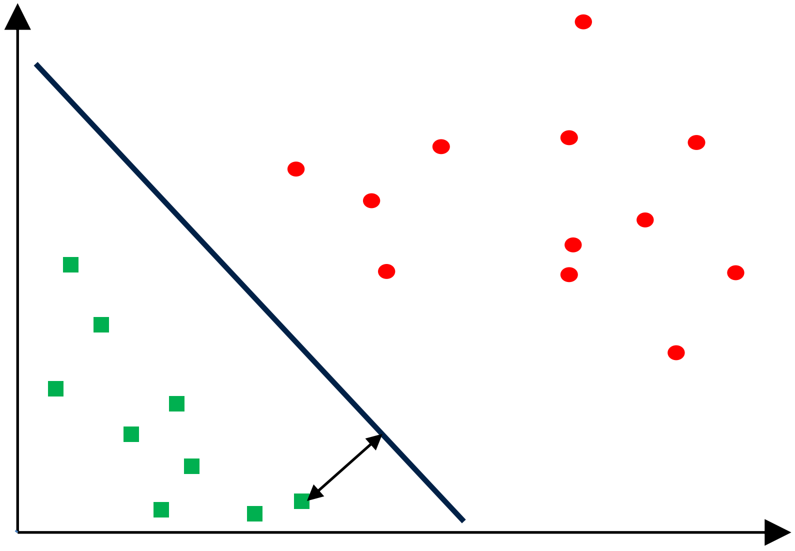
$$\gamma = \min_{\mathcal{D}} \gamma_n$$

- Functional margin $\widehat{\gamma}_n$ of an example $(\mathbf{x}^{(n)}, y^{(n)})$ with respect to the hyperplane is

$$\widehat{\gamma}_n = y^{(n)} (\mathbf{w}^{\mathrm{T}} \mathbf{x}^{(n)} + w_0)$$

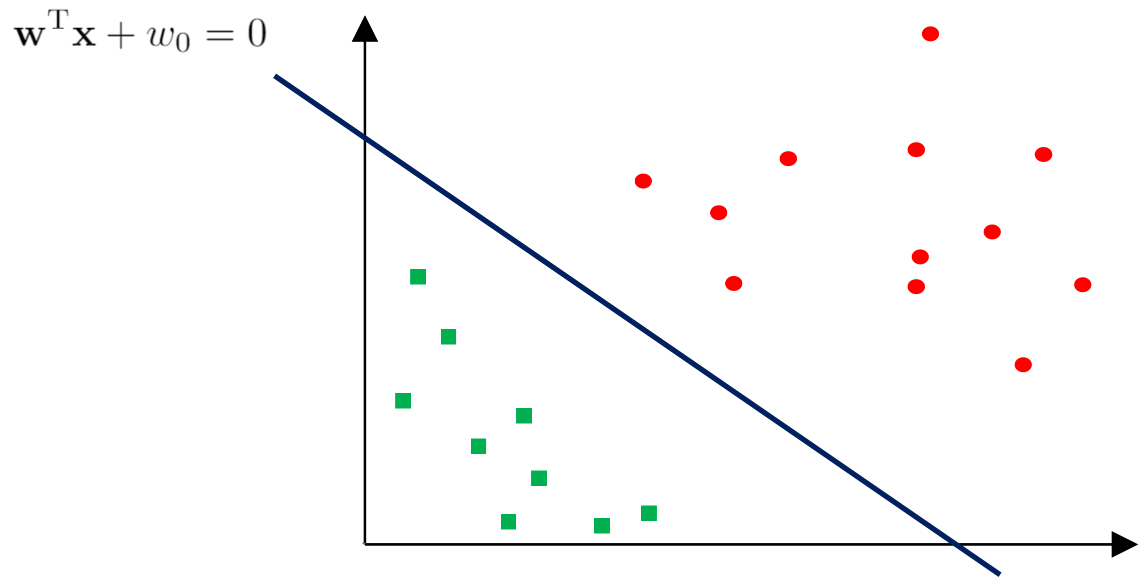
- $+$ve $\widehat{\gamma}_n$ means the example is correctly classified.
- $-$ve $\widehat{\gamma}_n$ means the example is incorrectly classified.

Maximum margin hyperplane



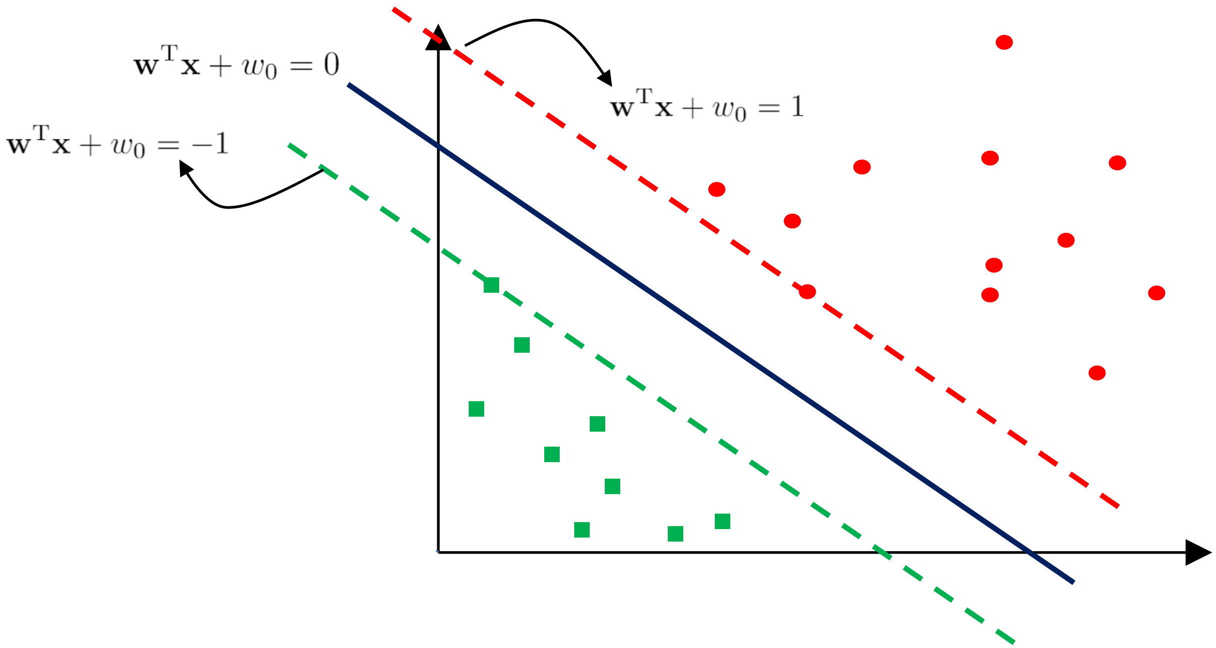
- Learn the hyperplane with the maximum separation.
- Support Vector Machines provide a framework for the learning the maximum margin hyperplane.
- SVMs find the most important examples in the training dataset that define the separating hyperplane. These examples are called the "support vectors".

Intuition



- Separating hyperplane: $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$.
- For good generalization want the examples to be away from the hyperplane.
- If $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 \geq 0$, then $y^{(n)} = 1$, i.e. $\mathbf{x}^{(n)}$ belongs to class $\mathcal{C}_1$.
  - If $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 >> 0$, then higher is the confidence of $\mathbf{x}^{(n)}$ belonging to class $\mathcal{C}_1$.
- If $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 < 0$, then $y^{(n)} = -1$, i.e. $\mathbf{x}^{(n)}$ belongs to class $\mathcal{C}_2$.
  - If $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 << 0$, then higher is the confidence of $\mathbf{x}^{(n)}$ belonging to class $\mathcal{C}_2$.
- But can scale $\mathbf{w}$ and w_0 to achieve large values of $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$

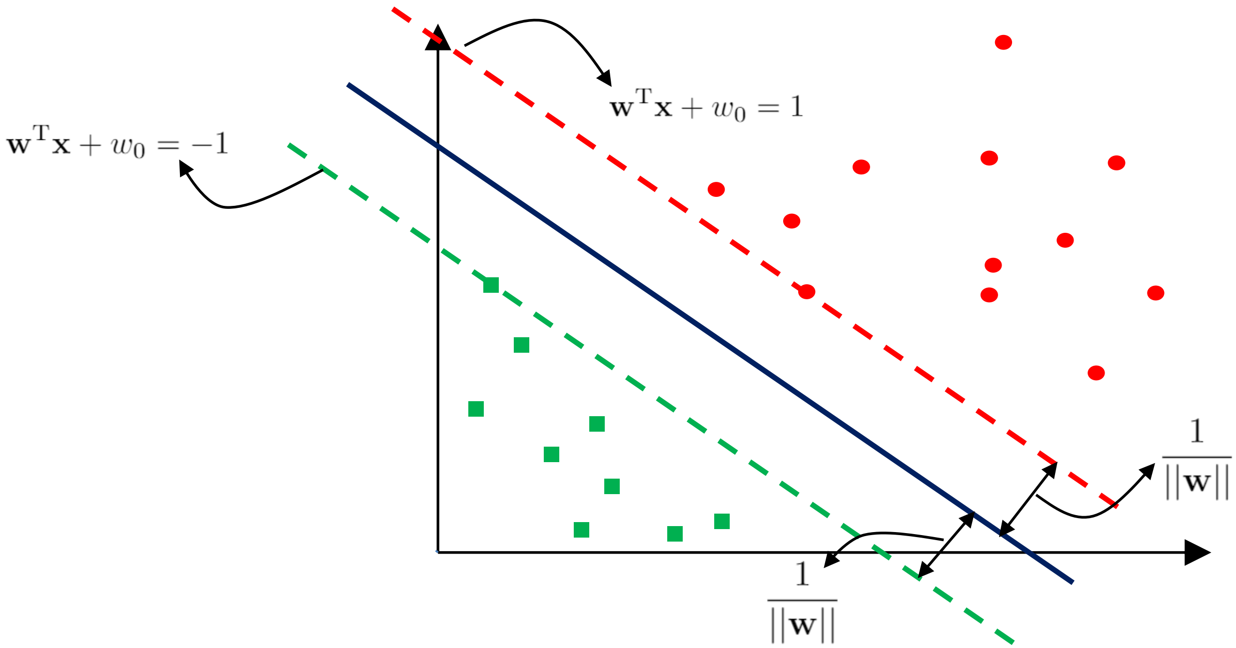
Margin boundaries



- Decision boundary (hyperplane) $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 = 0$ is to be chosen such that
  - If $\mathbf{x}^{(n)}$ is in $\mathcal{C}_1$ ($y^{(n)} = 1$): $\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0 \geq 1$
  - If $\mathbf{x}^{(n)}$ is in $\mathcal{C}_2$ ($y^{(n)} = -1$): $\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0 \leq -1$
- So we have $\min\limits_{n=(1,\ldots,N)} |\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0| = 1$
- Margin condition:

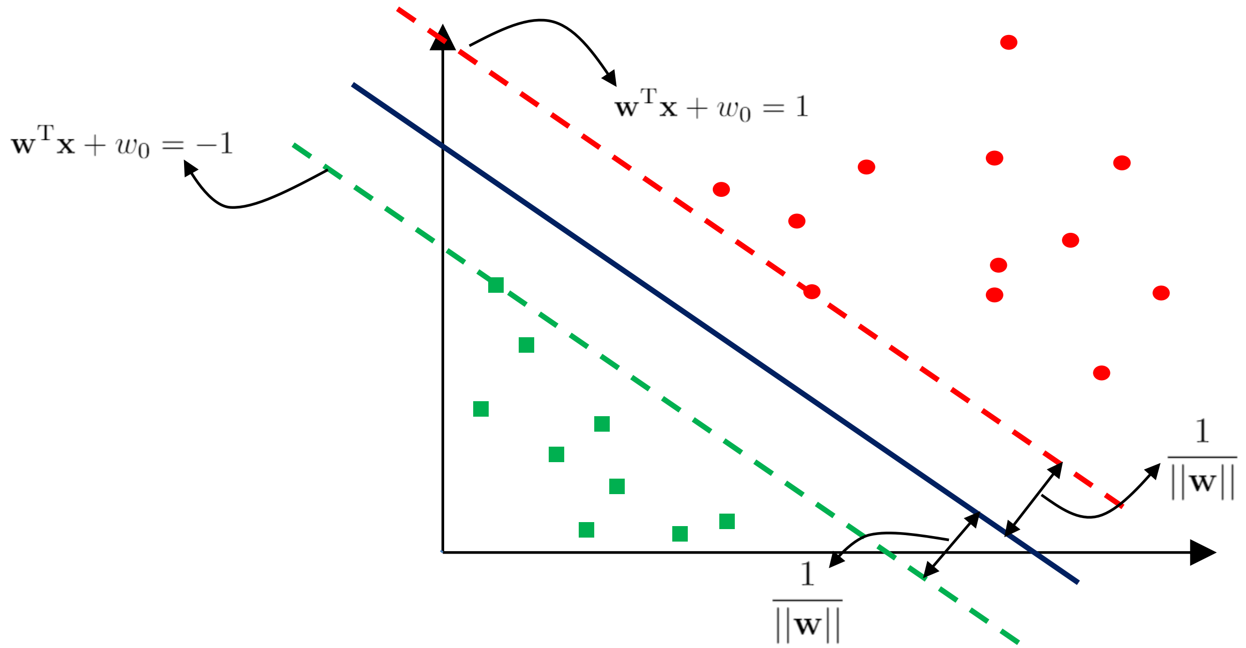
$$y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0) \geq 1, \qquad n = 1, 2, \ldots N$$

Support Vector Machines



- The goal is to find the optimal hyperplane separating the classes that has the maximal margin.
- Recall, the signed distance of a point $\mathbf{x}$ from the decision boundary is given as $\frac{f(\mathbf{x})}{||\mathbf{w}||}$.
- The distance between the two margins is then $\frac{2}{||\mathbf{w}||}$.
- Obtain a decision boundary (hyperplane) with the maximum possible margin.

Hard-margin SVM

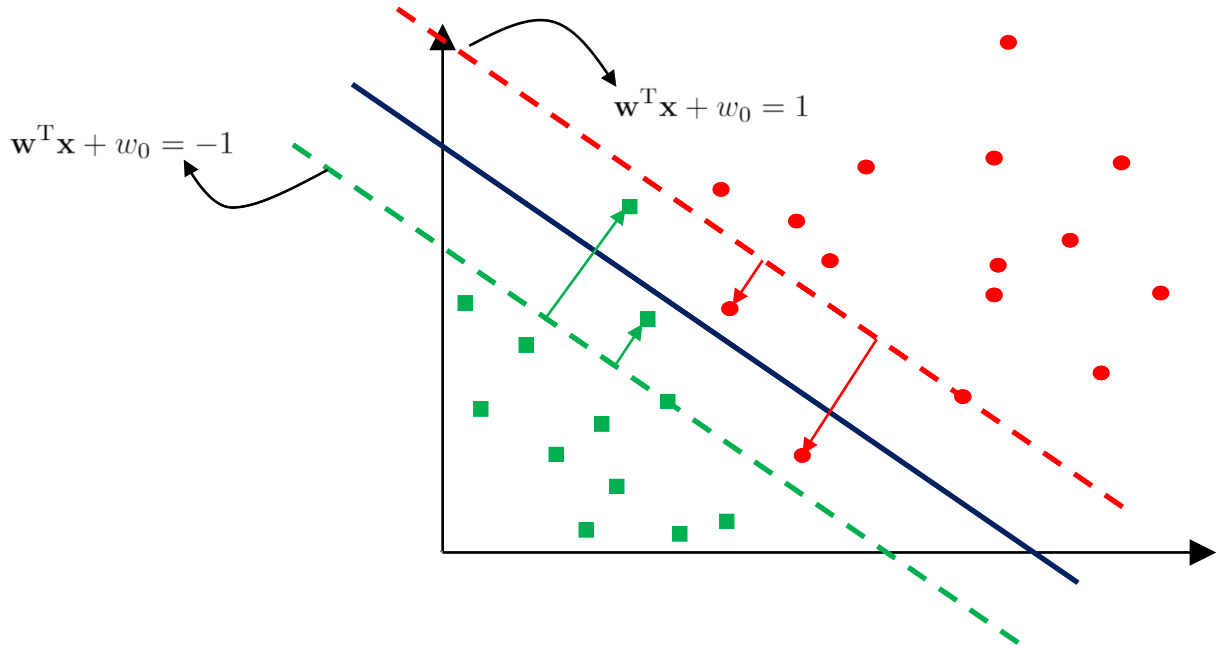


$$\text{Maximize } \frac{1}{||\mathbf{w}||} \longleftrightarrow \text{Minimize } ||\mathbf{w}||^2 \text{ or } \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

$$\boxed{\begin{aligned} &\min_{\mathbf{w}, w_0} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to} \quad y_n[\mathbf{w}^T\mathbf{x}_n + w_0] \geq 1, \qquad n = 1, \ldots, N \end{aligned}}$$

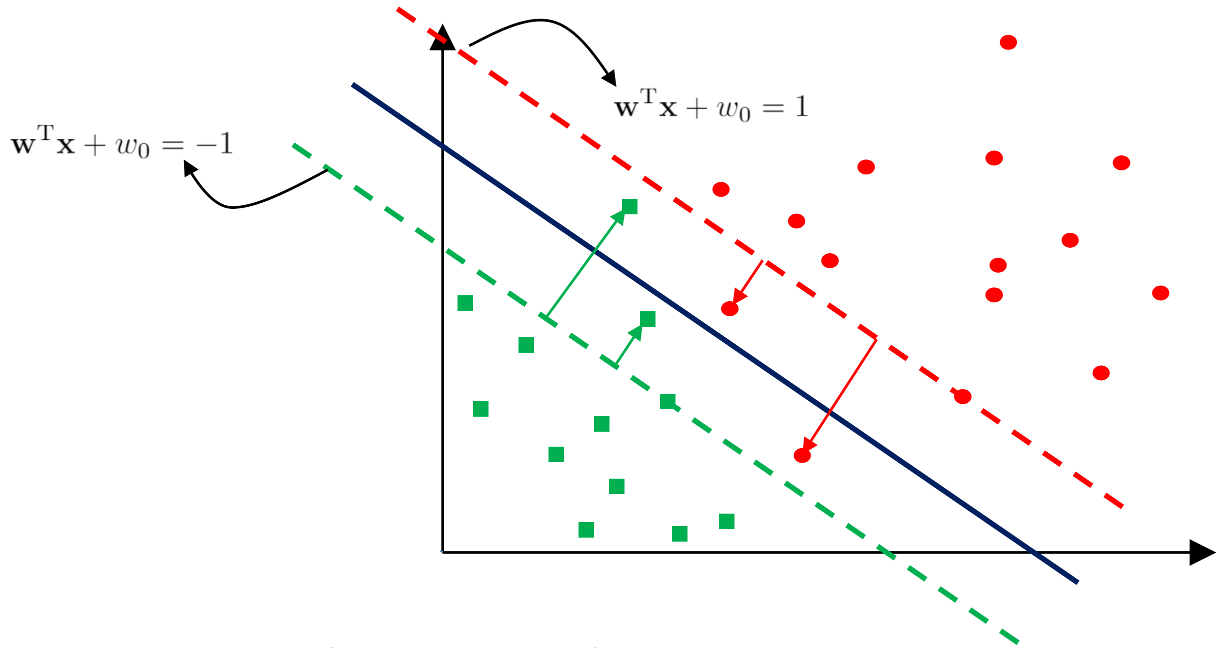
Hard-margin SVM objective

Slack variables



- Data not linearly separable in input space (due to noise).
- For nonlinear boundary, perfect separation of training data in the feature space can lead to poor generalization.
- Method modified to permit a few points to lie on the wrong side of the separating hyperplane.
- Approach: Use slack variables ξ_n, where $n = 1, .., N$, for every data point.

Soft-margin SVM



- Each example (say the nth) is associated with a variable $\xi_n \geq 0$ which indicates the degree to which the margin constraint is violated.
- ξ_ns are known as the "slack" variables.
- Soft-margin constraint: $y^{(n)}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0) \geq 1 - \xi_n$.

$$\min_{\mathbf{w}, w_0, \boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n=1}^{N}\xi_n$$

$$\text{subject to} \quad y^{(n)}[\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0] \geq 1 - \xi_n, \quad \text{and} \quad \xi_n \geq 0, \qquad n = 1, \ldots, N$$

Constrained Optimization

Constrained optimization problem

Optimization objective $\min_{\mathbf{w}} f(\mathbf{w})$ subject to $g_p(\mathbf{w}) \leq 0, \quad p = 1, ..., P$

$h_q(\mathbf{w}) = 0, \quad q = 1, ..., Q$

- Lagrangian:

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = f(\mathbf{w}) + \sum_{p=1}^{P} \lambda_p g_p(\mathbf{w}) + \sum_{q=1}^{Q} \gamma_q h_q(\mathbf{w})$$

$$\lambda_p \geq 0, \quad p = 1, ..., P$$

where λ_ps and γ_qs are the Lagrange multipliers.

- Suppose

$$L_P(\mathbf{w}) = \max_{\boldsymbol{\lambda} \geq 0, \boldsymbol{\gamma}} \mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$$

then

$$L_P(\mathbf{w}) = \begin{cases} \infty & \text{if } \ g_p(\mathbf{w}) > 0 \ \text{ or } \ h_q(\mathbf{w}) \neq 0 \quad \text{(any constraint violated)} \\ f(\mathbf{w}) & \text{otherwise} \end{cases}$$

Primal problem

- Therefore we have

$$\begin{aligned}\min_{\mathbf{w}} L_P(\mathbf{w}) &= \min_{\mathbf{w}} \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\gamma}} \mathcal{L}(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\gamma}) \\ &= \min_{\mathbf{w}} f(\mathbf{w})\end{aligned}$$

- So solving for $\min_{\mathbf{w}} \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\gamma}} \mathcal{L}(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\gamma})$ is equivalent to solving our original optimization problem.

- This is known as the **primal problem**, and

$$\mathcal{P} = \min_{\mathbf{w}} \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\gamma}} \mathcal{L}(\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\gamma})$$

  is the value of the primal problem.

Dual problem

- On interchanging the order of max and min we obtain the **dual problem**:

$$\mathcal{D} = \max_{\boldsymbol{\lambda}\geq 0,\boldsymbol{\gamma}} \min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$$

  where $\mathcal{D}$ is the value of the dual problem.

- The primal and the dual problem are related as

$$\mathcal{D} \leq \mathcal{P}$$

  with the equality holding when the following conditions are satisfied:

  – f and g_ps are convex i.e. their Hessian is positive semi-definite. Note, linear and affine functions are also convex.

  – h_qs are affine i.e. they can be represented in the form $h_q(\mathbf{z}) = \mathbf{a}_q^{\mathrm{T}}\mathbf{z} + \mathbf{b}_q$.

Solving hard-margin SVM

- Hard-margin SVM objective:

$$\boxed{\begin{aligned} &\min_{\mathbf{w},w_0} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ &\text{subject to} \quad y^{(n)}[\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} + w_0] \geq 1 \qquad n = 1, \ldots, N \end{aligned}}$$

- Lagrangian:

$$\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{n=1}^{N} \lambda_n (1 - y^{(n)}[\mathbf{w}^T\mathbf{x}^{(n)} + w_0])$$

- Objective:

$$\min_{\mathbf{w},w_0} \max_{\boldsymbol{\lambda}\geq 0} \mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{w},w_0} \mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\lambda})$$

- Partial derivatives of $\mathcal{L}$ with respect to $\mathbf{w}$ and w_0 yield:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \boxed{\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}^{(n)}}$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} \lambda_n y^{(n)} = 0$$

Solving hard-margin SVM

- Substitution of the conditions in $\mathcal{L}$ yields

$$\max_{\boldsymbol{\lambda}\geq 0} \min_{\mathbf{w},w_0} \mathcal{L}(\mathbf{w},\boldsymbol{\lambda},w_0) = \max_{\boldsymbol{\lambda}\geq 0} -\frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}\lambda_m\lambda_n y^{(m)}y^{(n)}\left((\mathbf{x}^{(m)})^T\mathbf{x}^{(n)}\right) + \sum_{n=1}^{N}\lambda_n$$

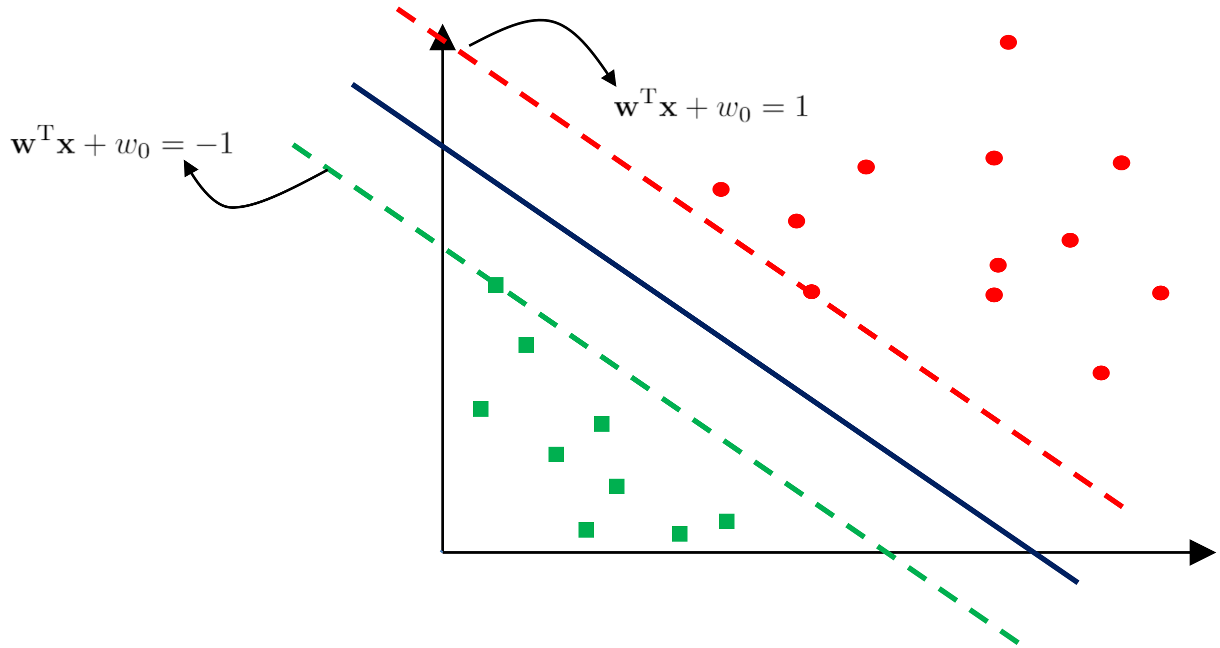
$$= \max_{\boldsymbol{\lambda}\geq 0} -\frac{1}{2}\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{D}\boldsymbol{\lambda} + \boldsymbol{\lambda}^{\mathbf{T}}\mathbf{1} \quad \text{where} \quad \mathbf{D}_{mn} = y^{(m)}y^{(n)}\left(\mathbf{x}^{(m)}\right)^{\mathrm{T}}\mathbf{x}^{(n)}$$

$$= \min_{\boldsymbol{\lambda}\geq 0} \frac{1}{2}\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{D}\boldsymbol{\lambda} - \boldsymbol{\lambda}^{\mathbf{T}}\mathbf{1}$$

$$\text{subject to} \quad \sum_{n=1}^{N}\lambda_n y^{(n)} = 0$$

- This is a convex optimization problem and can solved using standard techniques.

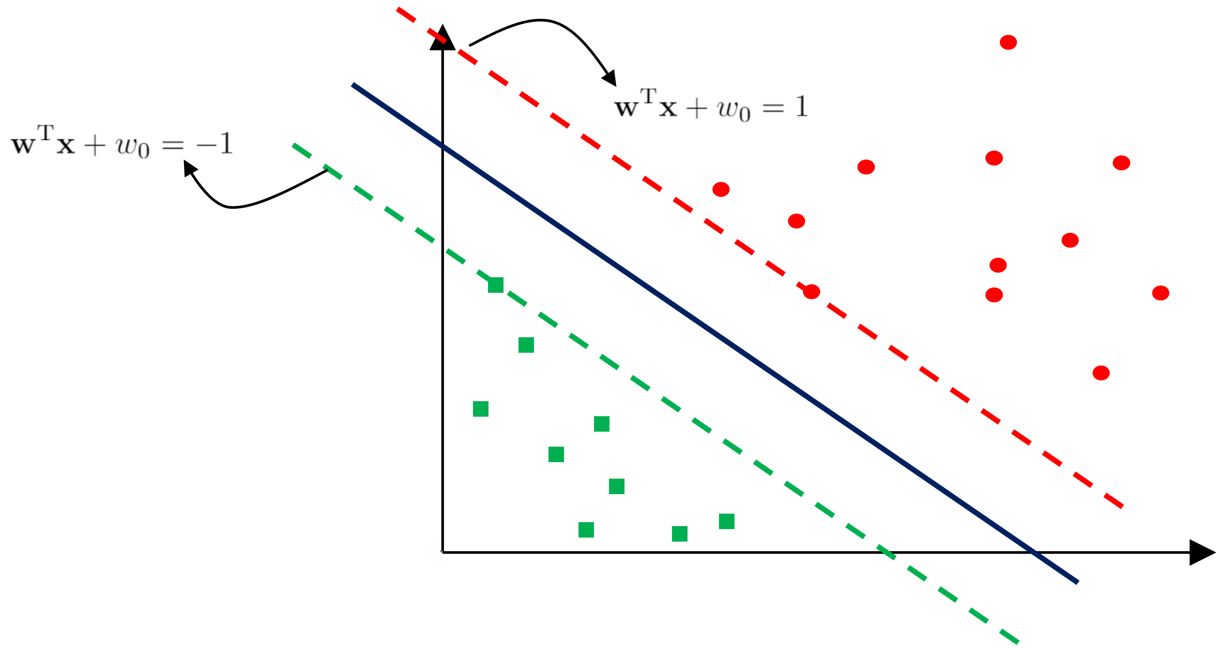
Solution to hard-margin SVM



- The solution to $\mathbf{w}$ can be found as

$$\mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}^{(n)}$$

Solution to hard-margin SVM



- For data points in class $\mathcal{C}_1$ we have

$$\min_{\mathbf{x}\in\mathcal{C}_1} \left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right) = 1 \quad \Rightarrow \quad \min_{\mathbf{x}\in\mathcal{C}_1} \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right) + w_0 = 1$$

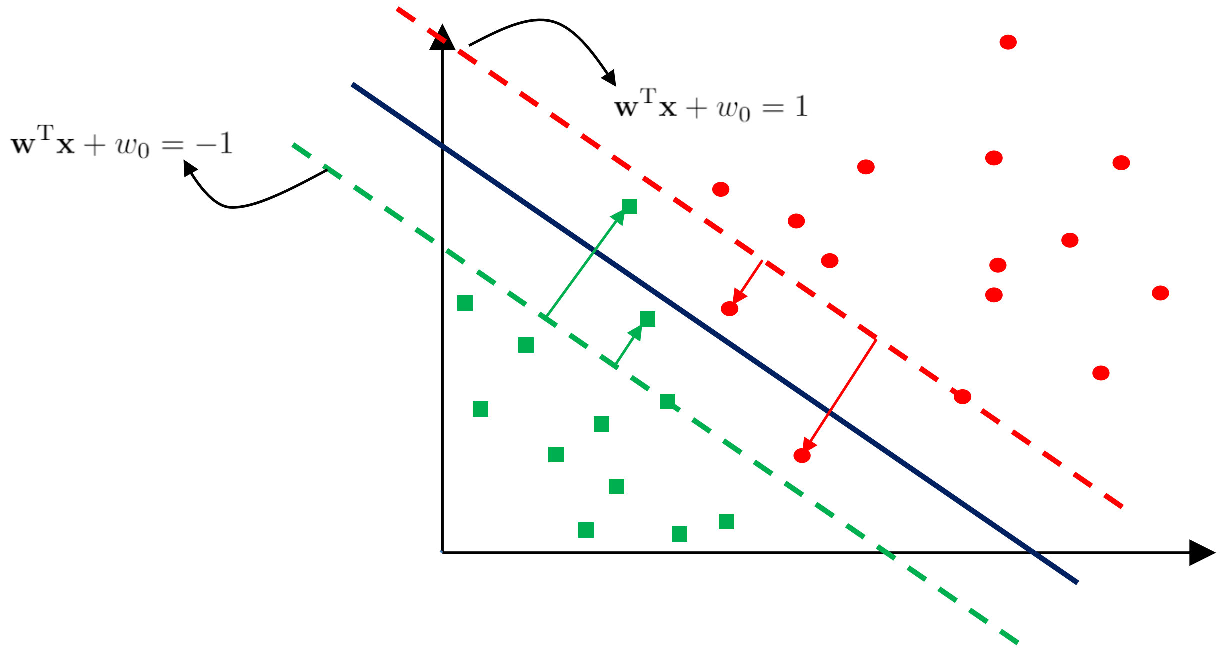
- For data points in class $\mathcal{C}_2$ we have

$$\max_{\mathbf{x}\in\mathcal{C}_2} \left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right) = -1 \quad \Rightarrow \quad \max_{\mathbf{x}\in\mathcal{C}_2} \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right) + w_0 = -1$$

- On adding the two equations yields

$$w_0 = -\frac{1}{2}\left(\min_{\mathbf{x}\in\mathcal{C}_1} \mathbf{w}^{\mathrm{T}}\mathbf{x} + \max_{\mathbf{x}\in\mathcal{C}_2} \mathbf{w}^{\mathrm{T}}\mathbf{x}\right)$$

Solving soft-margin SVM



- Lagrangian

$$\mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\gamma}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\lambda_n(1 - \xi_n - y^{(n)}[\mathbf{w}^T\mathbf{x}^{(n)} + w_0]) - \sum_{n=1}^{N}\gamma_n\xi_n$$

- Objective:

$$\min_{\mathbf{w}, w_0, \boldsymbol{\xi}} \max_{\boldsymbol{\lambda}, \boldsymbol{\gamma}} \mathcal{L}(\mathbf{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\gamma})$$

Solving soft-margin SVM

- Taking partial derivatives with respect to the primal variables $(\mathbf{w}, w_0, \xi_n)$ and setting them to zero:
  - With respect to $\mathbf{w}$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} \lambda_n y^{(n)} \mathbf{x}$$

  - With respect to w_0

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} \lambda_n y^{(n)} = 0$$

  - With respect to ξ_n

$$\frac{\partial \mathcal{L}}{\partial \xi_n} = 0 \quad \Rightarrow \quad \lambda_n + \gamma_n = C$$

- Solution of $\mathbf{w}$ is of the same form as in the hard-margin SVM.
- Since $\gamma_n \geq 0$ and $\lambda_n + \gamma_n = C$, we have $\lambda_n \leq C$.

Solving soft-margin SVM

- Substituting $\mathbf{w}$ in $\mathcal{L}$ and using the constraints imposed by the other equations, the dual problem is obtained as

$$\max_{\boldsymbol{\lambda}\le C,\boldsymbol{\gamma}\ge 0} L_D(\boldsymbol{\lambda},\gamma) = \max_{\boldsymbol{\lambda}\le C,\boldsymbol{\gamma}\ge 0} -\frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}\lambda_m\lambda_n y^{(m)}y^{(n)}\big((\mathbf{x}^{(m)})^T\mathbf{x}^{(n)}\big) + \sum_{n=1}^{N}\lambda_n$$

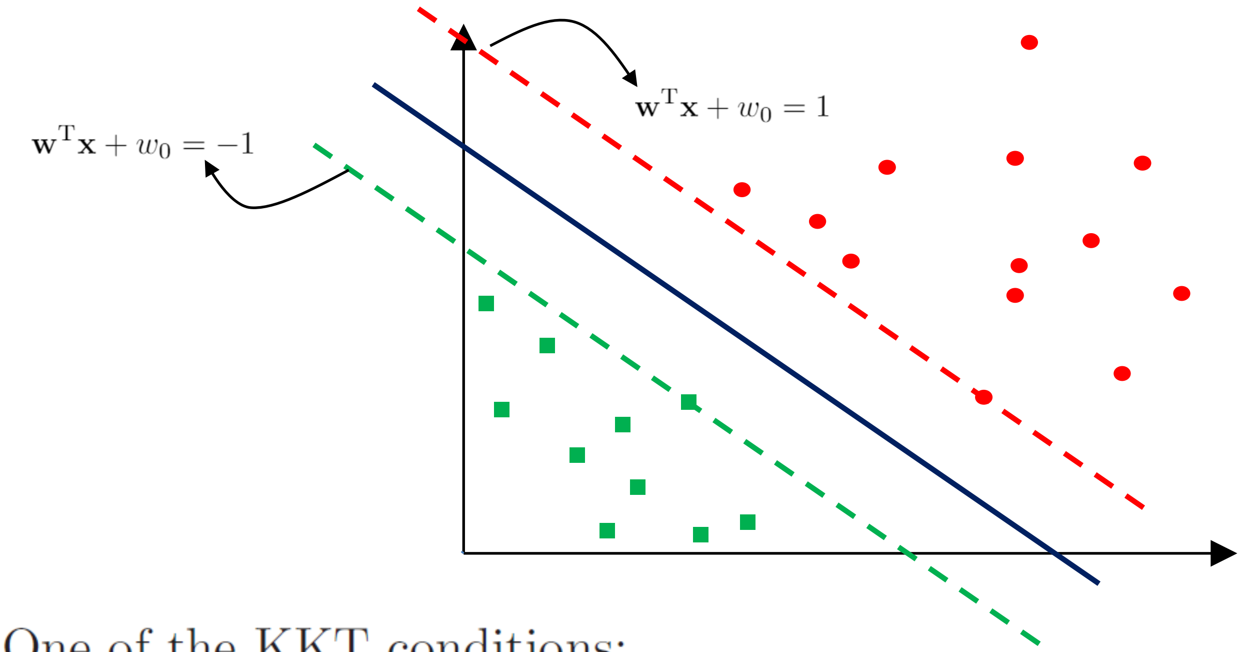
$$= \max_{\boldsymbol{\lambda}\le C} -\frac{1}{2}\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{D}\boldsymbol{\lambda} + \boldsymbol{\lambda}^{\mathbf{T}}\mathbf{1} \qquad \text{where} \quad \mathbf{D}_{mn} = y^{(m)}y^{(n)}\left(\mathbf{x}^{(m)}\right)^{\mathrm{T}}\mathbf{x}^{(n)}$$

$$= \min_{\boldsymbol{\lambda}\le C} \frac{1}{2}\boldsymbol{\lambda}^{\mathbf{T}}\mathbf{D}\boldsymbol{\lambda} - \boldsymbol{\lambda}^{\mathbf{T}}\mathbf{1}$$

$$\text{subject to} \quad \sum_{n=1}^{N}\lambda_n y^{(n)} = 0$$

- This is a convex optimization problem and can be solved using Quadratic programming solvers.

Hard-margin support vectors



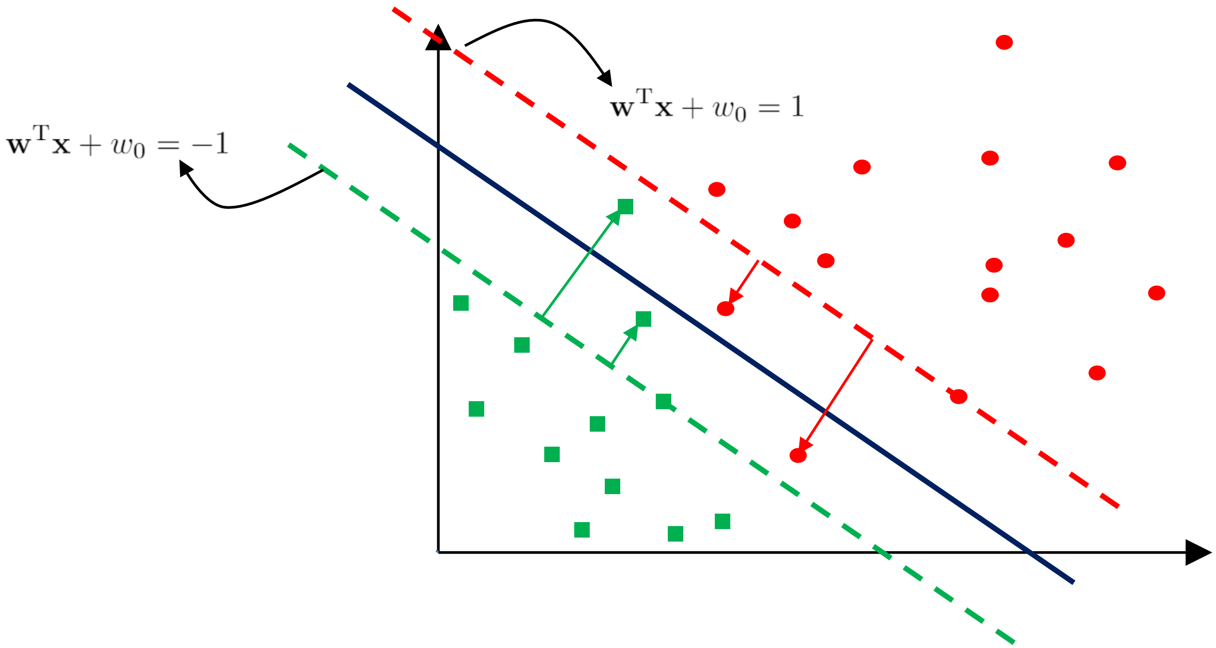
- One of the KKT conditions:

$$\lambda_n \left(y^{(n)} \left(\mathbf{w}^T \mathbf{x} + w_0 \right) - 1 \right) = 0$$

- This implies that λ_ns are non-zero for only those examples which lie on the margin boundaries.
  - For all other examples $\lambda_n = 0$

- Therefore the solution of the decision boundary is only affected by examples which lie on the margin boundaries.
  - These examples are called the support vectors.

Soft-margin support vectors



- Three types of support vectors:
  - $\xi_n = 0$: Examples lying on the margin boundaries.
  - $0 < \xi_n < 1$: Examples lying in the margin region and on the correct side of the separating hyperplane.
  - $\xi_n \geq 1$: Examples lying on the wrong side of the separating hyperplane.

Multi-class Classification

One-against-all

- Suppose the number of classes is J.
- Approach: Construct J SVM models
  - The jth SVM model is trained such that
    - examples in the jth class are labelled positive
    - examples in all other classes are labelled negative
- Finally we have J decision functions

$$
\begin{aligned}
\left(\mathbf{w}^{(1)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(1)} &= 0 \\
\left(\mathbf{w}^{(2)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(2)} &= 0 \\
&\;\vdots \\
\left(\mathbf{w}^{(J)}\right)^{\mathrm{T}}\mathbf{x} + w_0^{(J)} &= 0
\end{aligned}
$$

- Prediction:

$$y^* = \arg\max_{j=[1,2,\ldots,J]} \left(\left(\mathbf{w}^{(j)}\right)^{\mathrm{T}}\mathbf{x}^* + w_0^{(j)}\right)$$

One-against-one

- Construct a classifier using data from two classes.
  - Say the jth classifier comprise mth and nth class.
- Training: In total construct $J(J-1)/2$ classifiers.
- Prediction:
  - Can use a voting strategy
    - If the jth classifier predicts the point to be in class m, then increase vote of class m by one
    - otherwise increase vote of class n by one
  - Repeat the process for all the $J(J-1)/2$ classifiers.
  - Assign example to the class which receives the highest number of votes.