SVM

Discussion:
Practice with SVM

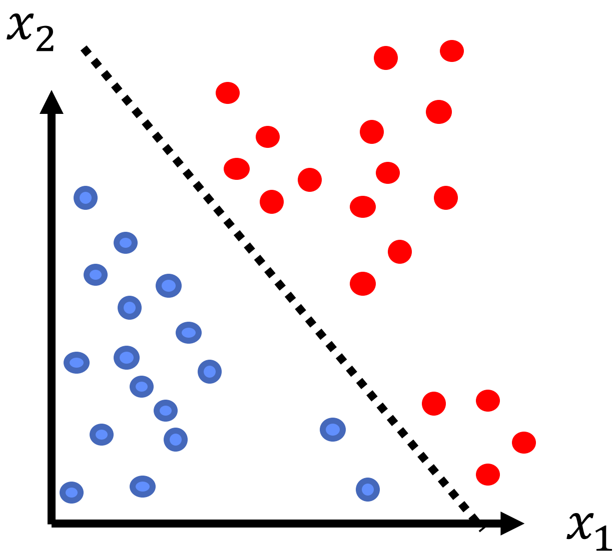
ECE/CS 498 DS
University of Illinois

Classification and Supervised Learning

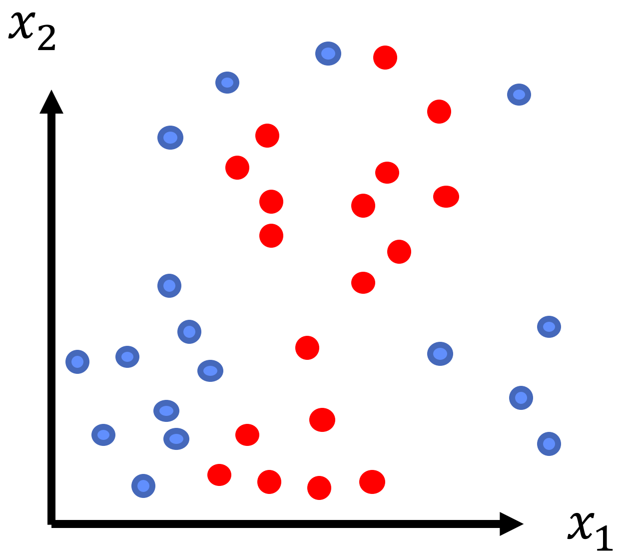
- One common application of machine learning is to perform classification, which is a form of supervised learning
- **Supervised Learning**: The process of using labeled input-output pairs to determine a mapping from inputs to an output
  - Labels are ”outputs” for data points – e.g. items in an image, type of dog breed given dog characteristics
- **Classification**: The process of classifying (or categorizing) a new data point into one of a finite number of classes (or bins) using existing labeled data
  - e.g. figuring out how to determine whether there is a dog or a cat in an image

Linearly Separable Data

- Binary classification becomes much simpler if the dataset is **linearly separable** – that is, there is a linear threshold that separates all samples from the two classes
  - For example, suppose the color of each datapoint represents its label



Linearly Separable



Not Linearly Separable

Hyperplanes

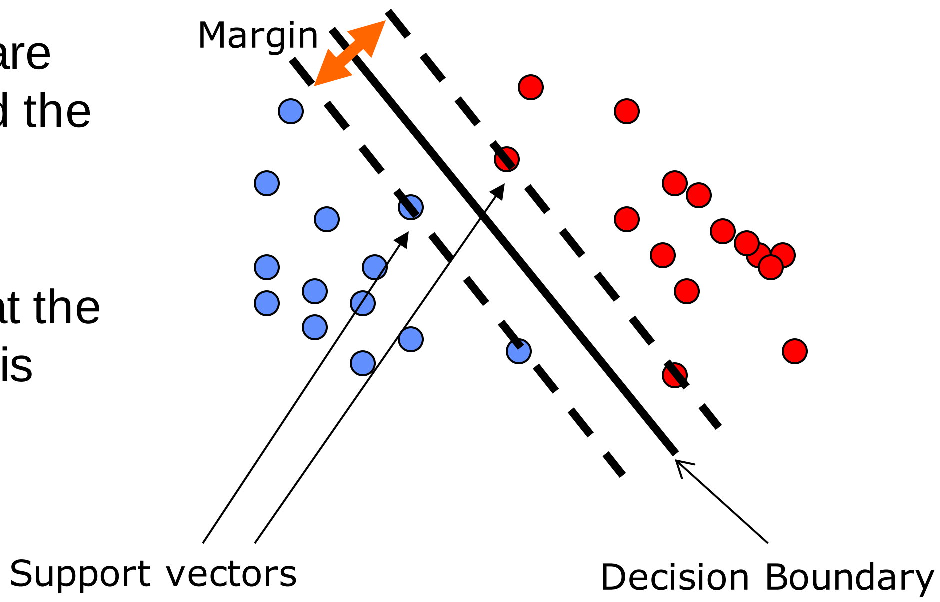
- We define such linear thresholds as **hyperplanes**. Depending on the dimensionality of the datapoints, hyperplanes take on different shapes:
  - In $\mathbb{R}^2$, a hyperplane is a line: $w_1 x_1 + w_2 x_2 + b = 0$
  - In $\mathbb{R}^3$, a hyperplane is a plane: $w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$
  - …..
- In general, the distance between a hyperplane and a datapoint $\boldsymbol{x} = [x_1, x_2, \ldots, x_d] \in \mathbb{R}^d$ is

$$\frac{|w_1 x_1 + w_2 x_2 + \ \ldots + w_d x_d + b|}{\sqrt{w_1^2 + w_2^2 + \ \ldots + w_d^2}}$$

$$= \frac{|\boldsymbol{w}^T \boldsymbol{x} + b|}{||\boldsymbol{w}||}$$

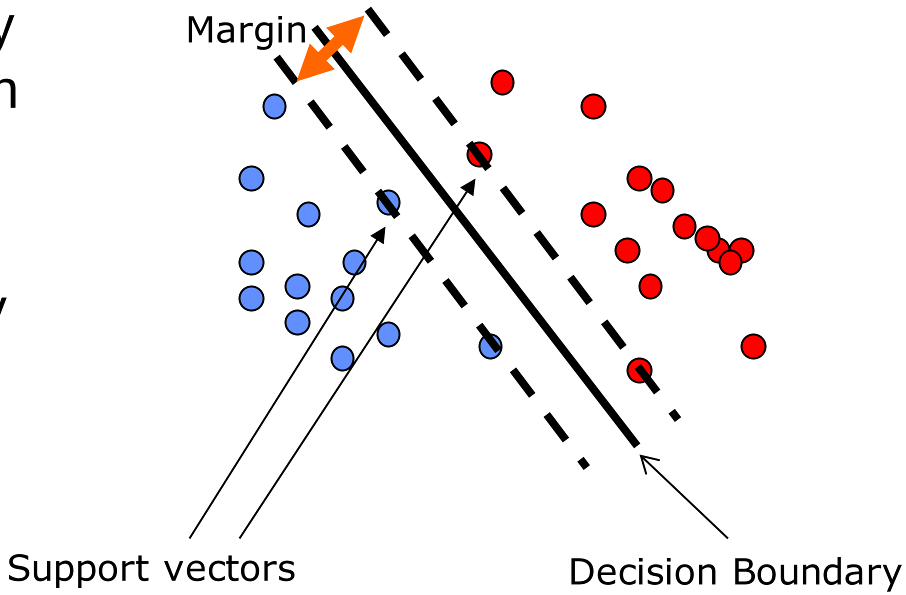
SVM

- Support vector machines are **classifiers** that build on the concept of **linear separability** by finding the *optimal* **hyperplane** to use as the decision boundary for classification tasks
  - At least two **support vectors** are used to define a margin around the decision boundary/hyperplane
  - In this case, *optimal* means that the margin around the hyperplane is maximal



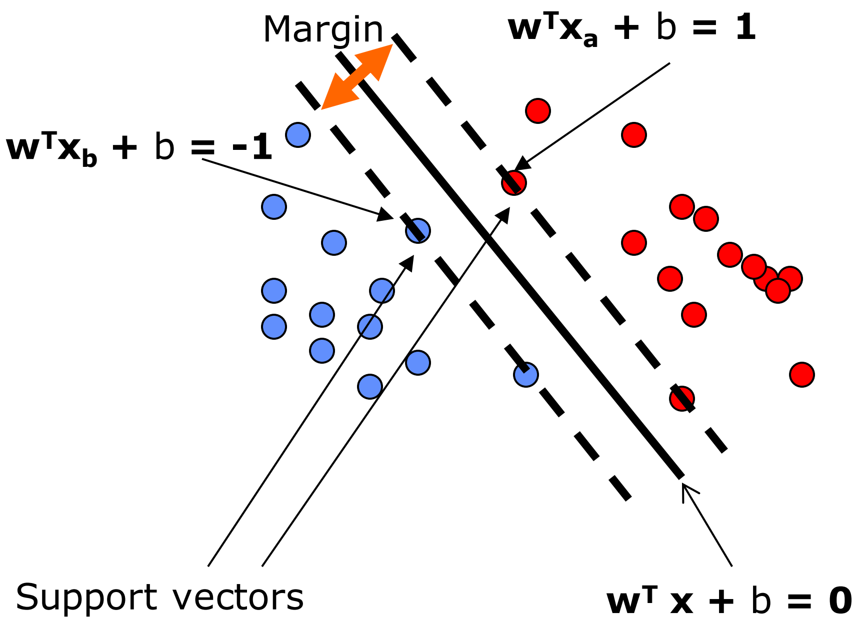
SVM: Support Vectors

- The support vectors are the data points that are most difficult to classify – i.e. the points closest to the decision boundary
- The support vectors are automatically determined by the SVM algorithm
- The support vectors lie along hyperplanes that are parallel to and equidistant from the decision boundary



SVM: Hyperplane Equations

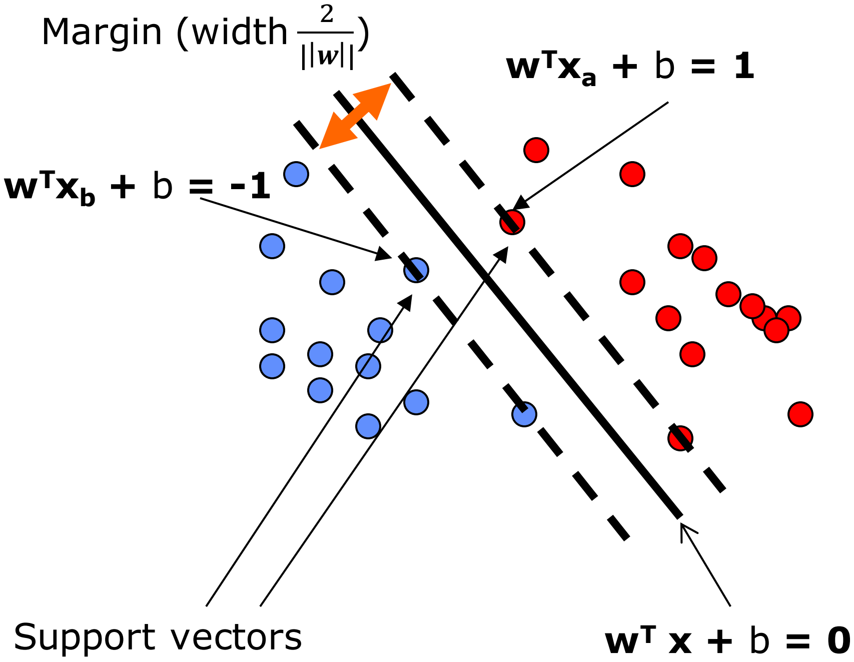
- By convention,
  - The decision boundary is defined by the hyperplane $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = 0$
  - The support vectors lie on the hyperplanes $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = 1$ and $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = -1$



SVM: Margin Width

- The margin width is the distance between the two hyperplanes containing the support vectors

- This is just
$$\frac{2}{||\boldsymbol{w}||}$$



SVM: Constraint

- **Constraint**: SVM must classify all points correctly with no data points within the margin
- Suppose each data point $\boldsymbol{x_i} \in \mathbb{R}^d$ has a corresponding label $y_i \in \{-1, 1\}$
- The constraint can be expressed mathematically as

$$For\ all\ \{(\boldsymbol{x_i}, y_i)\}, \qquad y_i(w^T\boldsymbol{x_i} + b) \geq 1$$

  - Think of $\hat{y}_i = w^T\boldsymbol{x_i} + b$ as the predicted label
  - Requiring that $y_i * \hat{y}_i \geq 0$ enforces that the correct classification is made
  - Requiring that $y_i * \hat{y}_i \geq 1$ enforces (i) that the correct classification is made and (ii) that the data point does not lie within the margin of the decision boundary

SVM: Objective

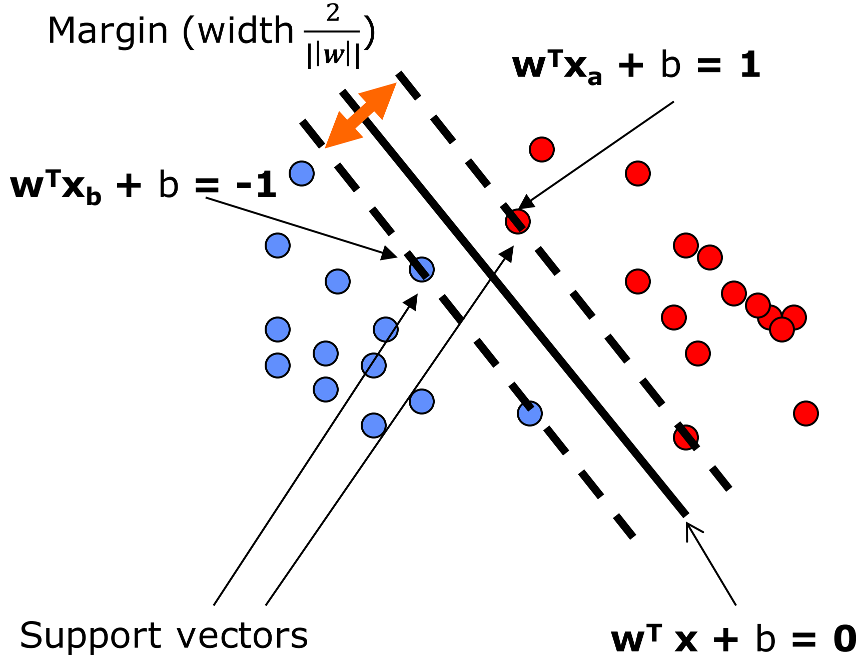
- **Objective**: margins around the hyperplane should be as large as possible

- In other words, we wish to

$$maximize \frac{2}{||\boldsymbol{w}||}, \text{ or}$$

$$minimize\ ||\boldsymbol{w}||, \text{ or}$$

$$minimize \frac{||\boldsymbol{w}||^2}{2}$$



- We minimize $\frac{||\boldsymbol{w}||^2}{2}$ instead of $||\boldsymbol{w}||$ because it will be more convenient later when evaluating derivatives

SVM: Optimization Problem

- Thus, the optimization problem for SVM can be summarized as follows:

$$\underset{\boldsymbol{w},b}{argmin} \frac{||\mathbf{w}||^2}{2} \; such \; that \;\; y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 \; \forall \; (\boldsymbol{x_i}, y_i)$$

- This is known as **hard-margin** SVM, where all points need to satisfy the constraint $y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1$

Practice: Hard-Margin SVM

- Suppose that we have the following labeled data points:
  - $\boldsymbol{x_1} = [1, 2, 3], y_1 = +1$
  - $\boldsymbol{x_2} = [4, 1, 2], y_2 = +1$
  - $\boldsymbol{x_3} = [-1, 2, -1], y_3 = -1$
- Which of the following $\boldsymbol{w}$ would be selected by a hard-margin SVM if $b = -0.4$?
  - $\boldsymbol{w} = [0.3, 0, 0.4]$
  - $\boldsymbol{w} = [0.2, 0, 0.4]$
  - $\boldsymbol{w} = [0.1, 0, 0.4]$
  - $\boldsymbol{w} = [0.4, 0, 0.2]$
- Recall, SVM will prefer the weights that
  - (i) correctly classify the data points
  - (ii) maximize the margin

Practice: Hard-Margin SVM

- Data Points: $\boldsymbol{x_1} = [1, 2, 3], \boldsymbol{x_2} = [4, 1, 2], \boldsymbol{x_3} = [-1, 2, -1]$,
  $\mathrm{y}_1 = +1, \mathrm{y}_2 = +1, \mathrm{y}_3 = -1$
- Test $\boldsymbol{w} = [0.3, 0, 0.4]$, $b = -0.4$
  - Check that constraints are satisfied
    $y_1(\boldsymbol{w}^T\boldsymbol{x_1} + b) = (1)([0.3, 0, 0.4]^T[1, 2, 3] - 0.4) = 1.1 \geq 1$
    $y_2(\boldsymbol{w}^T\boldsymbol{x_2} + b) = (1)([0.3, 0, 0.4]^T[4, 1, 2] - 0.4) = 1.6 \geq 1$
    $y_3(\boldsymbol{w}^T\boldsymbol{x_3} + b) = (-1)([0.3, 0, 0.4]^T[-1, 2, -1] - 0.4) = 1.1 \geq 1$

  All constraints are satisfied

  - Optimize margin width

$$\frac{||\boldsymbol{w}||^2}{2} = \frac{(0.3^2 + 0^2 + 0.4^2)}{2} = 0.125$$

Practice: Hard-Margin SVM

- Data Points: $\boldsymbol{x_1} = [1, 2, 3], \boldsymbol{x_2} = [4, 1, 2], \boldsymbol{x_3} = [-1, 2, -1],$
  $y_1 = +1, y_2 = +1, y_3 = -1$
- Test $\boldsymbol{w} = [0.2, 0, 0.4]$, $b = -0.4$
  - Check that constraints are satisfied

  When checked in a similar manner to that of the previous slide, all constraints are satisfied
  - Optimize margin width

$$\frac{||\boldsymbol{w}||^2}{2} = \frac{(0.2^2 + 0^2 + 0.4^2)}{2} = 0.10$$

This is a smaller $\frac{||\boldsymbol{w}||^2}{2}$ value than that from the weights in choice 1.
Thus, so far, $\boldsymbol{w} = [0.2, 0, 0.4]$ is the preferred choice

Practice: Hard-Margin SVM

- Data Points: $\boldsymbol{x_1} = [1, 2, 3], \boldsymbol{x_2} = [4, 1, 2], \boldsymbol{x_3} = [-1, 2, -1],$
  $$\mathrm{y}_1 = +1, \mathrm{y}_2 = +1, \mathrm{y}_3 = -1$$
- Test $\boldsymbol{w} = [0.1, 0, 0.4]$, $b = -0.4$
  - Check that constraints are satisfied
    $y_1(\boldsymbol{w}^T\boldsymbol{x_1} + b) = (1)([0.1, 0, 0.4]^T[1, 2, 3] - 0.4) = 0.9 \ngeq 1$

Constraints aren't satisfied, so this choice won't be selected.

Thus, $\boldsymbol{w} = [0.2, 0, 0.4]$ is still the preferred choice

Practice: Hard-Margin SVM

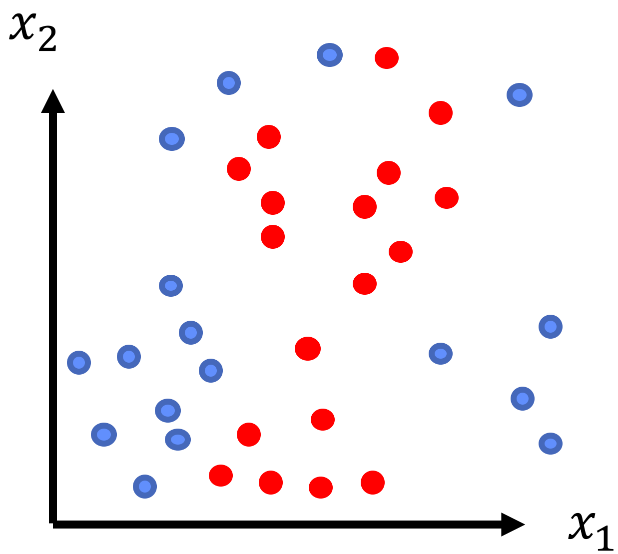
- Data Points: $\boldsymbol{x_1} = [1, 2, 3], \boldsymbol{x_2} = [4, 1, 2], \boldsymbol{x_3} = [-1, 2, -1],$
  $$\mathrm{y}_1 = +1, \mathrm{y}_2 = +1, \mathrm{y}_3 = -1$$
- Test $\boldsymbol{w} = [0.4, 0, 0.2]$, $b = -0.4$
  - Check that constraints are satisfied
    $y_1(\boldsymbol{w}^T\boldsymbol{x_1} + b) = (1)([0.4, 0, 0.2]^T[1, 2, 3] - 0.4) = 0.6 \ngeq 1$

Constraints aren't satisfied, so this choice won't be selected.

Thus, $\boldsymbol{w} = [0.2, 0, 0.4]$ is the final preferred choice

Soft-margin SVM: Motivation

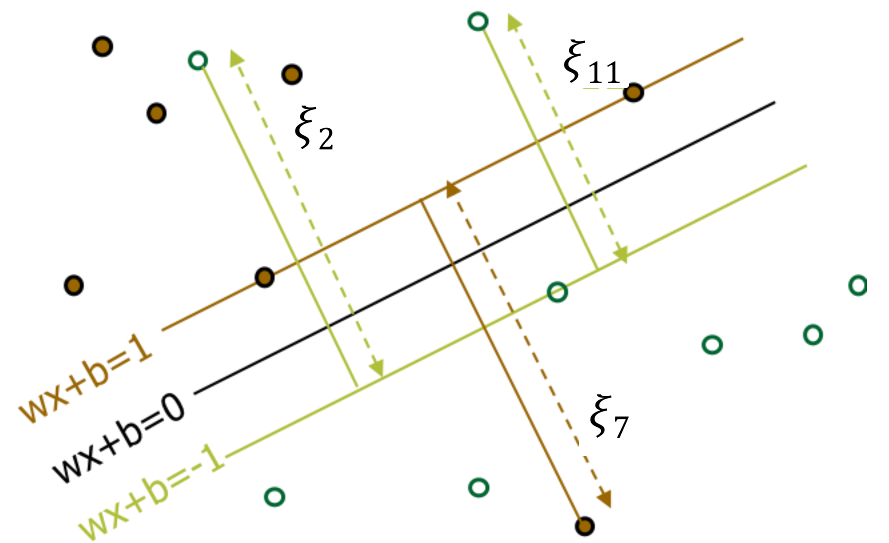
- What happens if the dataset is not linearly separable?
  - Constraints can never be satisfied, and SVM will not arrive at a solution!



Not Linearly Separable

Soft Margin SVM: Constraint

- We can still operate on non-linearly separable datasets by allowing ***slack*** for the n classifications



- Each slack term ξ_i should be nonzero and should report exactly how much the constraint $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$ is being violated.

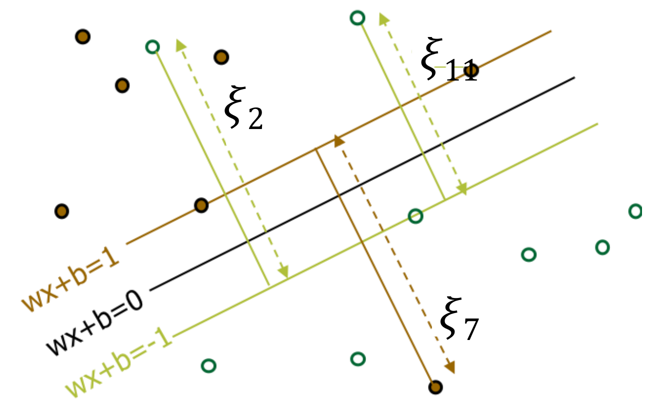
- Thus, we adjust our constraints to be
  - $(1)\ \xi_i \geq 0\ \forall\ i$
  - $(2)\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i\ \forall\ i$

Soft Margin SVM: Objective

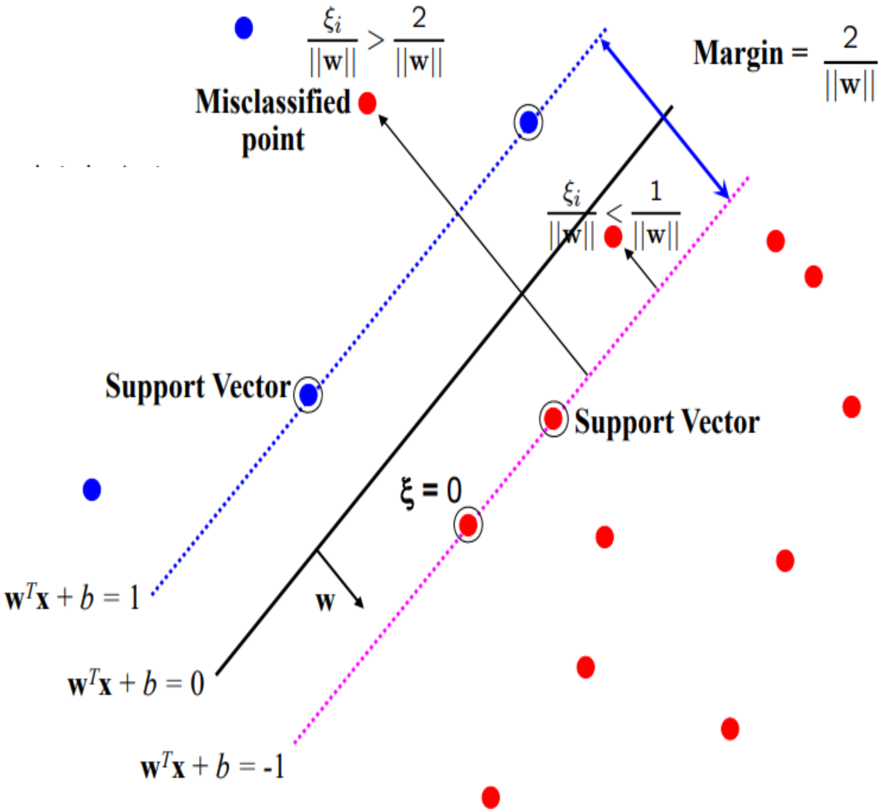
- Ideally, we wish to minimize the amount of slack that is required. Thus, we adjust our objective to be

$$\underset{\mathbf{w},b}{argmin} \frac{||\mathbf{w}||^2}{2} + C \sum_{j=1}^{n} \xi_j$$

  - If k represents the number of data points that require slack, k could be used instead of n in the above expression since the remaining $n - k$ points have $\xi = 0$
  - C is a hyper-parameter that can be used to control overfitting
    - As $C \to \infty$, ξ_j must grow smaller in order to minimize the objective function ⇒ we approach hard-margin SVM
    - As $C \to 0$, the effective cost of the slack terms are minimized and $||\mathbf{w}|| \to 0$ ⇒ margins grow arbitrarily large
    - Visualize effects of changing C at https://cs.stanford.edu/~karpathy/svmjs/demo/



SVM: Issues with Separability



- **Ideal Behavior**: $\xi_i = 0$
  - No slack is required
- **Margin violation**: $0 < \xi_i \leq 1$
  - Point lies between margin and correct side of hyperplane
- **Misclassification**: $\xi_i > 1$
  - Point is on the wrong side of margin

Soft Margin SVM: Optimization Problem

- The optimization problem for soft-margin SVM can be summarized as follows:

$$\underset{\boldsymbol{w},b}{argmin} \frac{||\mathbf{w}||^2}{2} + C\sum_{j=1}^{k} \xi_j \ such\ that\ \ y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) \geq 1 - \xi_i\ and\ \xi_i \geq 0\ \forall\ i$$

- We can rewrite this as an unconstrained optimization problem:
  - $y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) \geq 1 - \xi_i \Rightarrow \xi_i \geq 1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b)$
  - To minimize ξ_i, we want $\xi_i = 1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b)$
  - Since $\xi_i \geq 0\ \forall\ i,\ \xi_i = max\{0, 1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b)\} = relu(1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b))$
    - Ideally classified datapoints (which have $y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) \geq 1$) should have 0 slack
    - Non-ideal datapoints (which have $y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) < 1$) should have slack of $\xi_i = 1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b)$

Soft Margin SVM: Optimization Problem

- Thus, the unconstrained optimization problem is

$$\underset{\boldsymbol{w},b}{argmin} \frac{||\mathbf{w}||^2}{2} + C \sum_{i=1}^{n} max\{0, 1 - y_i\,(\boldsymbol{w}^T \boldsymbol{x_i} + b)\}$$

$$= \underset{\boldsymbol{w},b}{argmin}\, L(\boldsymbol{w}, b)$$

Soft Margin SVM: Gradient Descent

- To find the optimal weights and bias, we will need to perform gradient descent. To do this, we calculate partial derivatives of L
- First, we take the partial derivative of L with respect to the weights $\boldsymbol{w}$

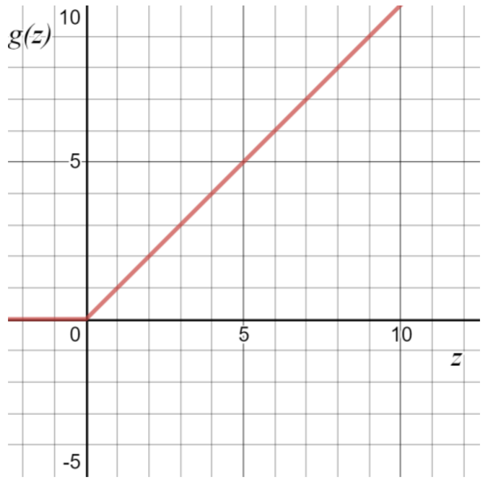
$$\frac{\partial L}{\partial \boldsymbol{w}} = \frac{\partial\left(\frac{||\mathbf{w}||^2}{2} + C\sum_{i=1}^{n} \max\{0, 1 - y_i\,(\boldsymbol{w}^T\boldsymbol{x_i} + b)\}\right)}{\partial \boldsymbol{w}}$$

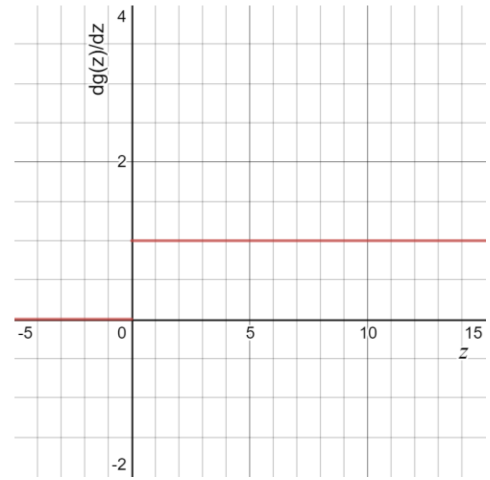
$$= \boldsymbol{w} + C\sum_{i=1}^{n} \frac{\partial \max\{0, 1 - y_i\,(\boldsymbol{w}^T\boldsymbol{x_i} + b)\}}{\partial \boldsymbol{w}}$$

- To complete this computation, we need to figure out the derivative of the max function

Soft Margin SVM: Gradient Descent

- Let $g(z) = \max\{0, \mathrm{z}\}$





- The plot of $g(z)$ is shown on the left, and its derivative is shown on the right. Thus,

$$\frac{\partial \max\{0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b)\}}{\partial \boldsymbol{w}} = \begin{cases} -y_i\boldsymbol{x_i} & if\ y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) < 1 \\ 0 & else \end{cases}$$

$$= \mathbb{I}(\ y_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) < 1)(-y_i\boldsymbol{x_i})$$

Soft Margin SVM: Gradient Descent

- Now, we can plug this back into our original equation:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} + C \sum_{i=1}^{n} \frac{\partial \max\{0, 1 - y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b)\}}{\partial \boldsymbol{w}}$$

$$= \boldsymbol{w} + C \sum_{i=1}^{n} \mathbb{I}(y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) < 1)(-y_i \boldsymbol{x_i})$$

Soft Margin SVM: Gradient Descent

- Similarly,

$$\frac{\partial L}{\partial b} = \frac{\partial\left(\frac{||\mathbf{w}||^2}{2} + C\sum_{i=1}^{n} \max\{0, 1 - y_i\,(\boldsymbol{w}^T \boldsymbol{x_i} + b)\}\right)}{\partial b}$$

$$= C \sum_{i=1}^{n} \frac{\partial \max\{0, 1 - y_i\,(\boldsymbol{w}^T \boldsymbol{x_i} + b)\}}{\partial b}$$

$$= C \sum_{i=1}^{n} \mathbb{I}(\, y_i(\boldsymbol{w}^T \boldsymbol{x_i} + b) < 1)(-y_i)$$

SVM Gradient Descent Example

- Suppose that in the beginning of iteration t of gradient descent, we have $\boldsymbol{w_t} = [4, 4]$, $b_t = -1, C = 1$

- We wish to train using the data points
  - $\boldsymbol{x_1} = [1, 1], y_1 = 1$
  - $\boldsymbol{x_2} = [2, -1], y_2 = -1$

- Calculate $\boldsymbol{w_{t+1}}$ and b_{t+1}, which are the updated weights and bias terms after one iteration of gradient descent

SVM Gradient Descent Example

- Initial conditions: $\boldsymbol{w_t} = [4, 4]$, $b_t = -1, C = 1$
- Data points: $\boldsymbol{x_1} = [1, 1]$, $\boldsymbol{x_2} = [2, -1], y_1 = 1, y_2 = -1$

$$\frac{\partial L}{\partial \boldsymbol{w_t}} = \boldsymbol{w_t} + C \sum_{i=1}^{n} \mathbb{I}(\, y_i(\boldsymbol{w_t^T x_i} + b_t) < 1)(-y_i \boldsymbol{x_i})$$

$$= \boldsymbol{w_t} + C\mathbb{I}(\, y_1(\boldsymbol{w_t^T x_1} + b_t) < 1)(-y_1 \boldsymbol{x_1})$$
$$+C\mathbb{I}(\, y_2(\boldsymbol{w_t^T x_2} + b_t) < 1)(-y_2 \boldsymbol{x_2})$$

$$= [4, 4] + \mathbb{I}(1(8 - 1) < 1)(-1 * [1, 1])$$
$$+\mathbb{I}((-1)(4 - 1) < 1)(-(-1) * [2, -1])$$

$$= [4, 4] + [0,0] + [2, -1] = [6, 3]$$

$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \frac{\partial L}{\partial \boldsymbol{w_t}} = [4, 4] - [6, 3] = [-2, 1]$$

SVM Gradient Descent Example

- Initial conditions: $\boldsymbol{w_t} = [4, 4]$, $b_t = -1, C = 1$
- Data points: $\boldsymbol{x_1} = [1, 1]$, $\boldsymbol{x_2} = [2, -1], y_1 = 1, y_2 = -1$

$$\frac{\partial L}{\partial b_t} = C \sum_{i=1}^{n} \mathbb{I}(\, y_i(\boldsymbol{w_t^T} \boldsymbol{x_i} + b_t) < 1)(-y_i)$$

$$= C\mathbb{I}(\, y_1(\boldsymbol{w_t^T} \boldsymbol{x_1} + b_t) < 1)(-y_1)$$
$$+ C\mathbb{I}(\, y_2(\boldsymbol{w_t^T} \boldsymbol{x_2} + b_t) < 1)(-y_2)$$

$$= \mathbb{I}(1(8 - 1) < 1)(-1)$$
$$+ \mathbb{I}((-1)(4 - 1) < 1)(-(-1)\,)$$

$$= 0 + 1 = 1$$

$$b_{t+1} = b_t - \frac{\partial L}{\partial b_t} = (-1)\ - 1 = -2$$

SVM: Kernel Functions

- Another way to combat issues with linear separability is to first **transform** the dataset to a higher dimension where there may be increased linear separability

Fig 1

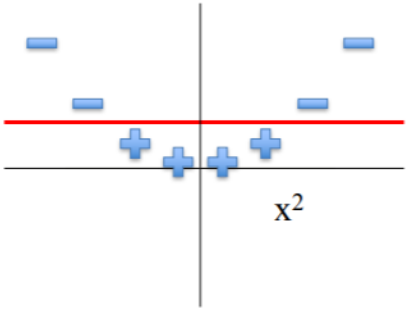


Fig 2

- For example, the data points in Fig 1 are not linearly separable
- Applying **transformation** $\Phi(x) = x^2$ gives data points in Fig 2, which are linearly separable
- Apply SVM on transformed data

Kernels

- Define a kernel $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \Phi(\boldsymbol{x_i})^T \Phi(\boldsymbol{x_j})$, the dot product of two transformed data points
- In SVM, it can be easier to simply use $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ during gradient descent when minimizing the objective function

Example of commonly used Kernels ($x_i, x_j \in \mathbb{R}$)

- Polynomial kernel of order 2: $K(x_i, x_j) = 1 + x_i + x_j + x_i^2 + x_j^2 + 2x_i x_j$
- Radial Basis Function: $K(\mathrm{x}_i, x_j) = \exp\left(-\frac{(\mathbf{x}_i - \boldsymbol{x}_j)^2}{2\gamma^2}\right)$