Smart Mobile Platform

Support Vector Machine (SVM)

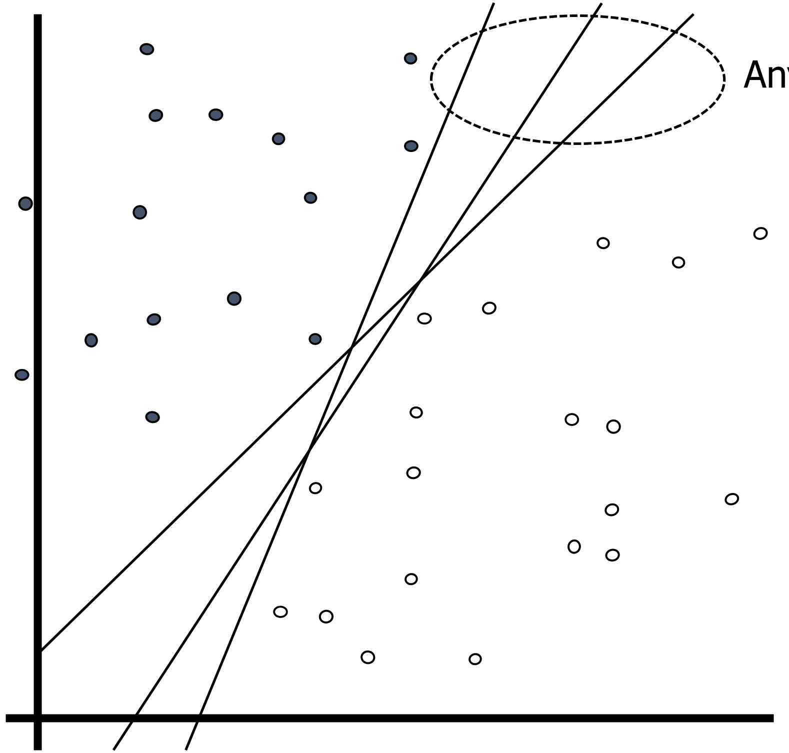
Prof. Joongheon Kim
Korea University, School of Electrical Engineering
https://joongheon.github.io
joongheon@korea.ac.kr

SVM (Outline)

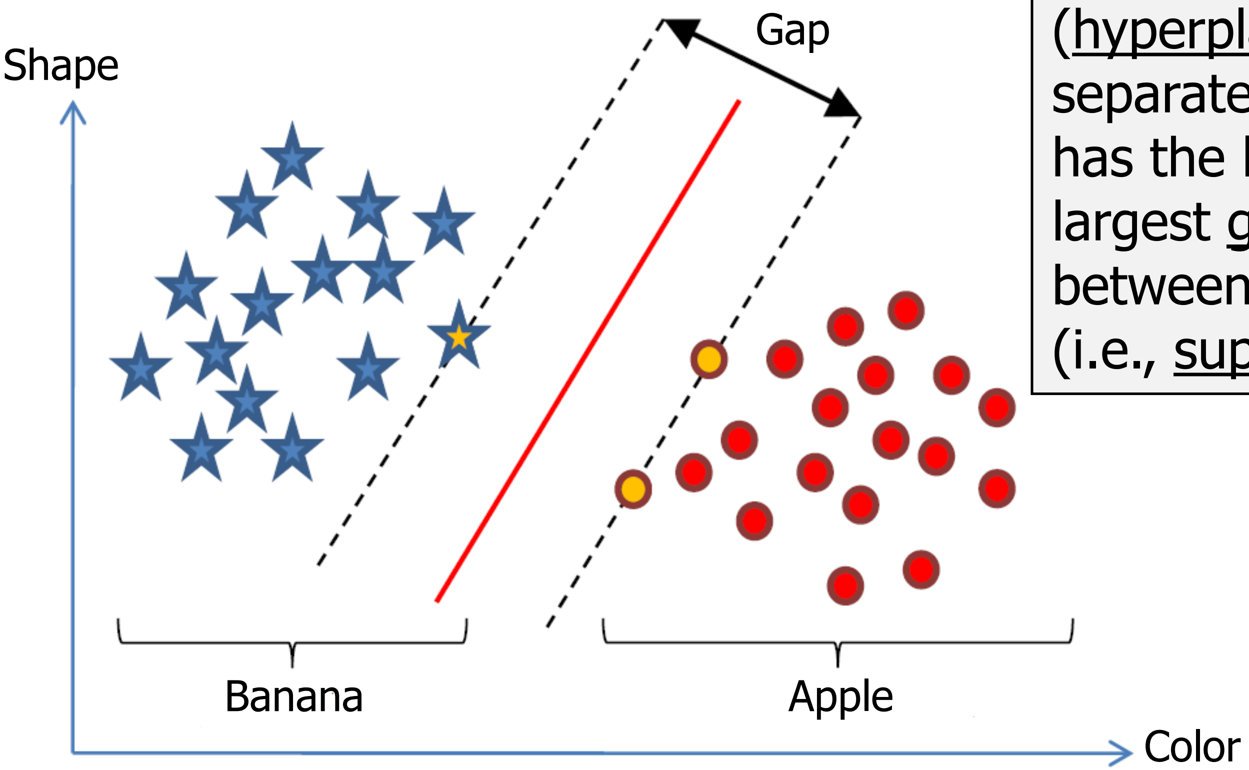
- **Support Vector Machine (SVM)**
  - **Main Idea**
  - Hyperplane in n-Dimensional Space
  - Brief Introduction to Optimization for SVM
  - SVM for Classification

Main Idea

- How can we classify the give data?

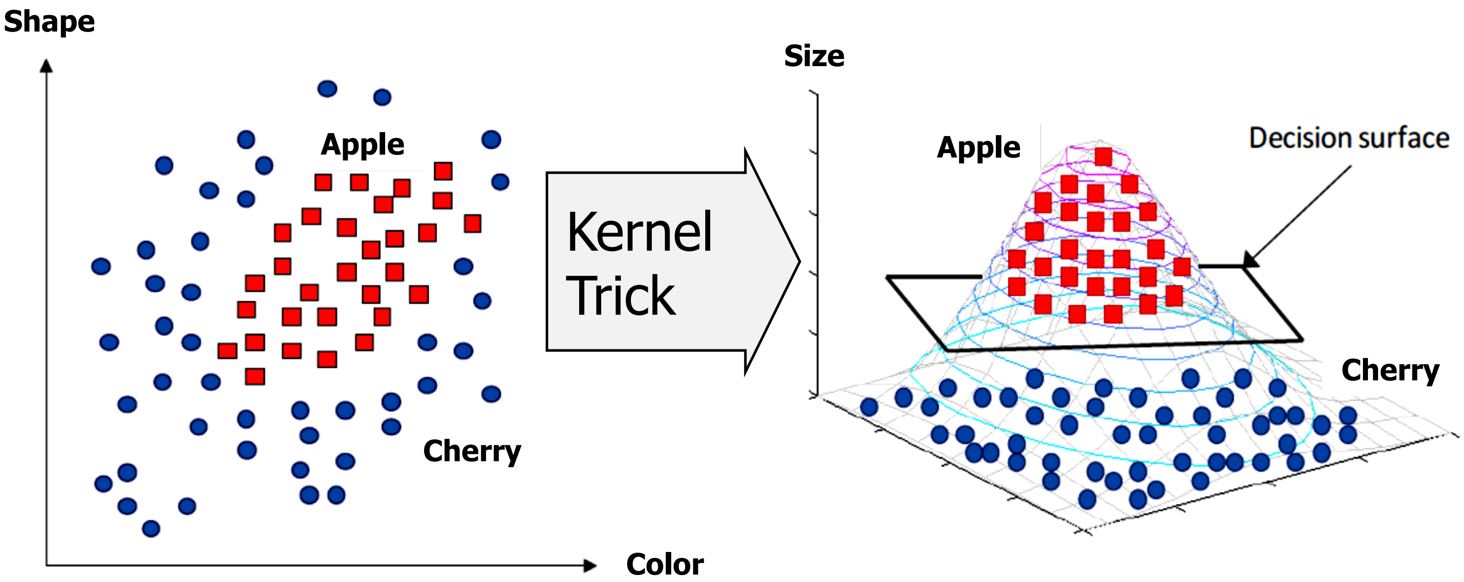


Any of these would be fine. But which is the best?



Find a linear decision surface (hyperplane) that can separate object classes and has the largest distance (i.e., largest gap (or margin)) between border-line objects (i.e., support vectors);

Main Idea

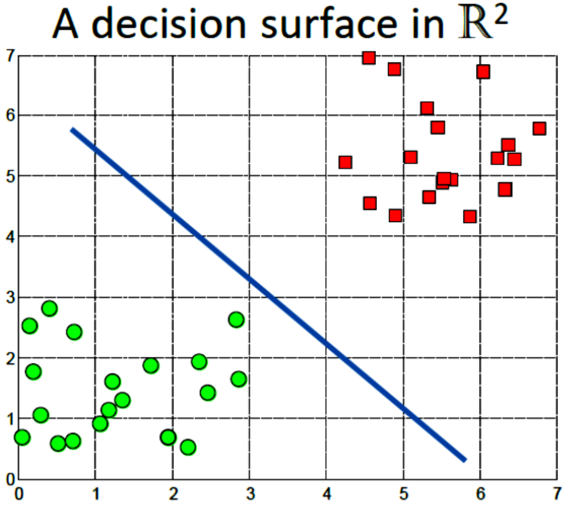


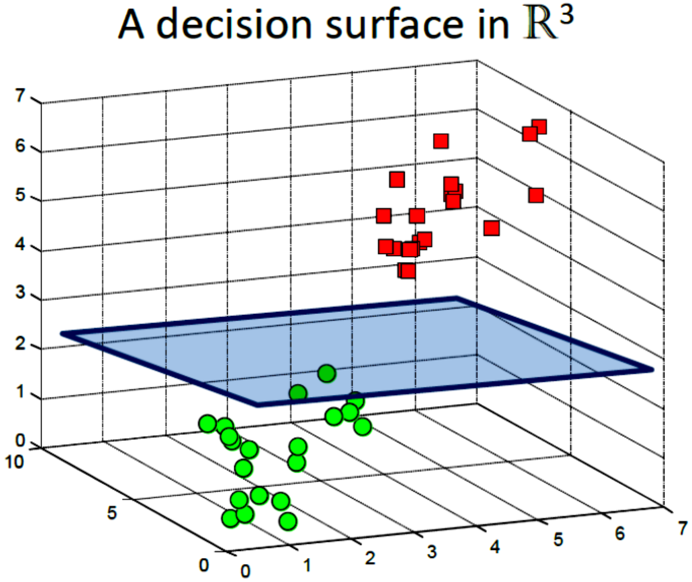
- If linear decision surface does not exist, the data is mapped into a higher dimensional space (feature space) where the separating decision surface is found.
- The feature space is constructed via mathematical projection (kernel trick).

- **Support Vector Machine (SVM)**
  - Main Idea
  - **Hyperplane in n-Dimensional Space**
  - Brief Introduction to Optimization for SVM
  - SVM for Classification

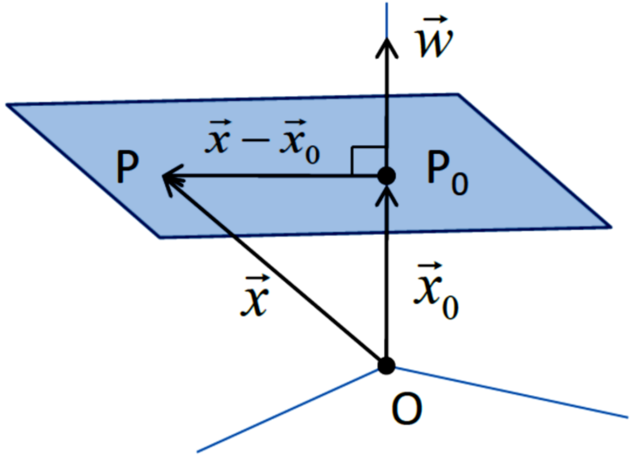
Hyperplane in n-Dimensional Space

[Definition (Hyperplane)] A subspace of one dimension less than its ambient space, i.e., the hyperplane in n-dimensional space means the $n-1$ subspace.





- Equations of a Hyperplane



- An equation of a hyperplane is defined by a point (P_0) and a perpendicular vector to the plane ($\vec{w}$) at that point.

- Define vectors: $\vec{x}_0$ and $\vec{x}$ where P is an arbitrary point on a hyperplane.

- A condition for P to be one the plane is that the vector $\vec{x} - \vec{x}_0$ is perpendicular to $\vec{w}$:

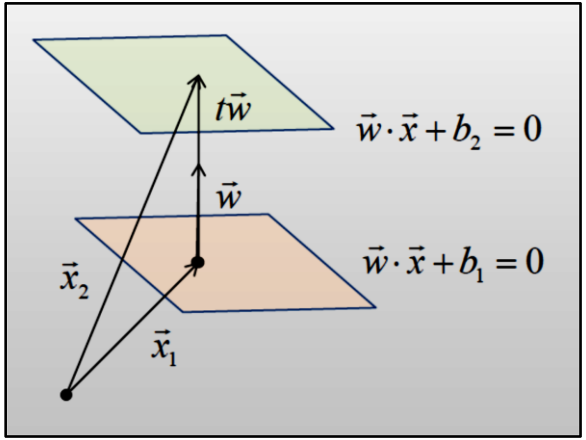
$$\vec{w} \cdot (\vec{x} - \vec{x}_0) = 0$$

$$\vec{w} \cdot \vec{x} - \vec{w} \cdot \vec{x}_0 = 0$$ and define $b = -\vec{w} \cdot \vec{x}_0$

$$\boxed{\vec{w} \cdot \vec{x} + b = 0}$$

- The above equations hold for R^n when $n > 3$.

- Equations of a Hyperplane



$\vec{x}_2 = \vec{x}_1 + t\vec{w}$

$D = \|t\vec{w}\| = |t|\|\vec{w}\|$

$$
\begin{aligned}
&\vec{w}\cdot\vec{x}_2 + b_2 = 0\\
&\vec{w}\cdot(\vec{x}_1 + t\vec{w}) + b_2 = 0\\
&\vec{w}\cdot\vec{x}_1 + t\|\vec{w}\|^2 + b_2 = 0\\
&(\vec{w}\cdot\vec{x}_1 + b_1) - b_1 + t\|\vec{w}\|^2 + b_2 = 0\\
&-b_1 + t\|\vec{w}\|^2 + b_2 = 0\\
&t = (b_1 - b_2)/\|\vec{w}\|^2
\end{aligned}
$$

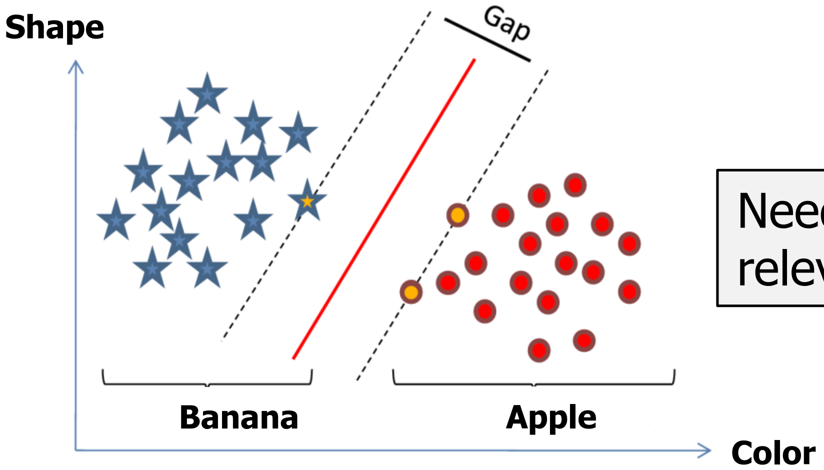
Therefore, $D = |t|\|\vec{w}\| = (b_1 - b_2)/\|\vec{w}\|$

Distance between two parallel hyperplanes $\vec{w}\cdot\vec{x} + b_1 = 0$ and $\vec{w}\cdot\vec{x} + b_2 = 0$ is equivalent to $D = \frac{|b_1 - b_2|}{\|\vec{w}\|}$.

- **Support Vector Machine (SVM)**
    - Main Idea
    - Hyperplane in n-Dimensional Space
    - **Brief Introduction to Optimization for SVM**
    - SVM for Classification

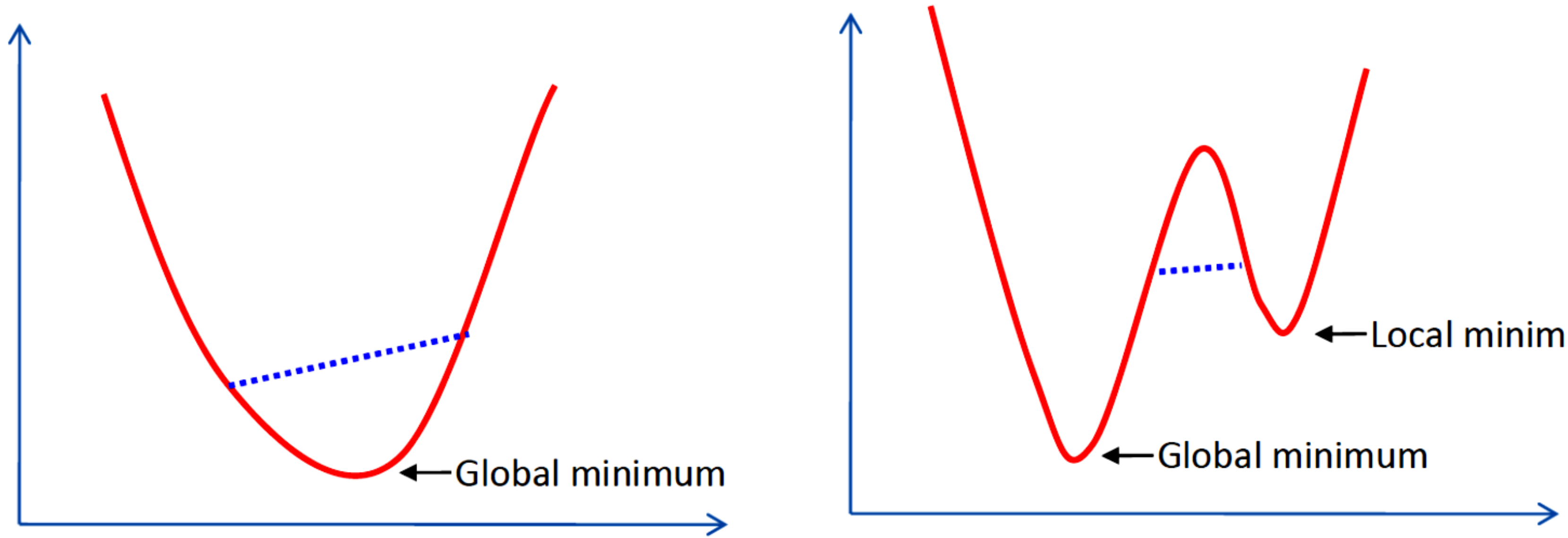
Brief Introduction to Optimization for Support Vector Machine

- Now, we understand
  - How to represent data (vectors)
  - How to define a linear decision surface (hyperplane)
- We need to understand
  - How to efficiently compute the hyperplane that separates two classes with the largest gap?



Need to understand the basics of relevant optimization theory

- Convex Functions
  - A function is called convex if the function lies below the straight line segment connecting two points, for any two points in the interval.
  - Property: Any local minimum is a global minimum.



- Quadratic programming (QP)
  - Quadratic programming (QP) is a special optimization problem: the function to optimize (objective) is quadratic, subject to linear constraints.
  - Convex QP problems have convex objective functions.
  - These problems can be solved easily and efficiently by greedy algorithms (because every local minimum is a global minimum).

- Quadratic programming (QP)
  - [Example], Constrained Optimization, i.e., **Lagrange Multiplier** is required.

Consider $\vec{x} = (x_1, x_2)$

Minimize $\boxed{\frac{1}{2}||\vec{x}||_2^2}$ subject to $\boxed{x_1 + x_2 - 1 \geq 0}$

Quadratic Objective   Linear Constraints

Consider $\vec{x} = (x_1, x_2)$

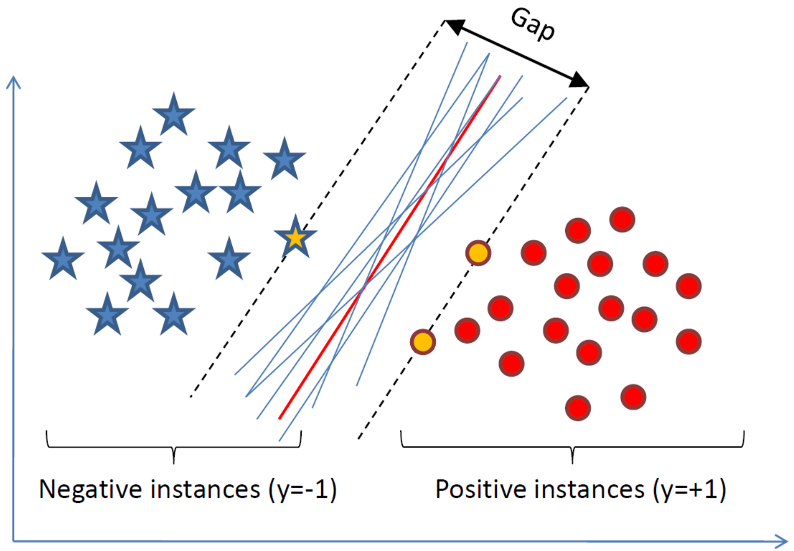
Minimize $\boxed{\frac{1}{2}(x_1^2 + x_2^2)}$ subject to $\boxed{x_1 + x_2 - 1 \geq 0}$

Quadratic Objective   Linear Constraints

- **Support Vector Machine (SVM)**
  - Main Idea
  - Hyperplane in n-Dimensional Space
  - Brief Introduction to Optimization for SVM
  - **SVM for Classification**

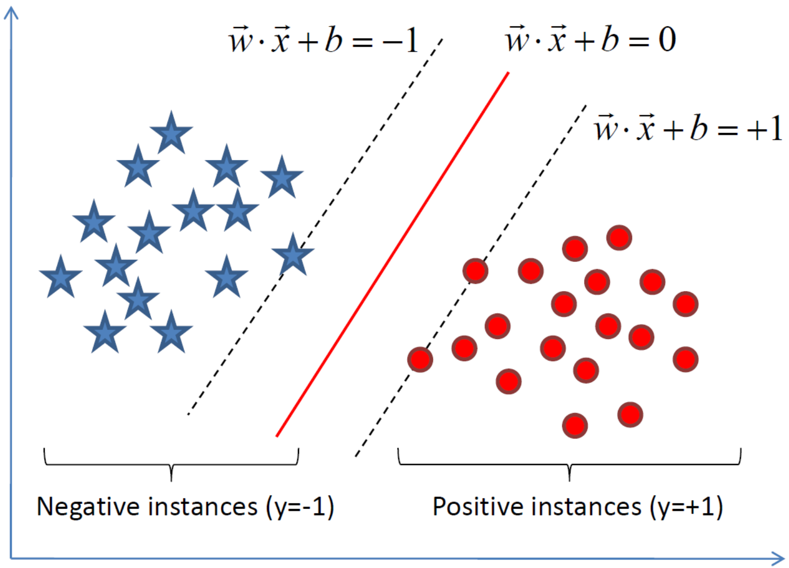
- SVM for Classification
  - (Case 1) Linearly Separable Data; Hard-Margin Linear SVM
  - (Case 2) Not Linearly Separable Data; Soft-Margin Linear SVM
  - (Case 3) Not Linearly Separable Data; Kernel Trick

- (Case 1) Linearly Separable Data; Hard-Margin Linear SVM



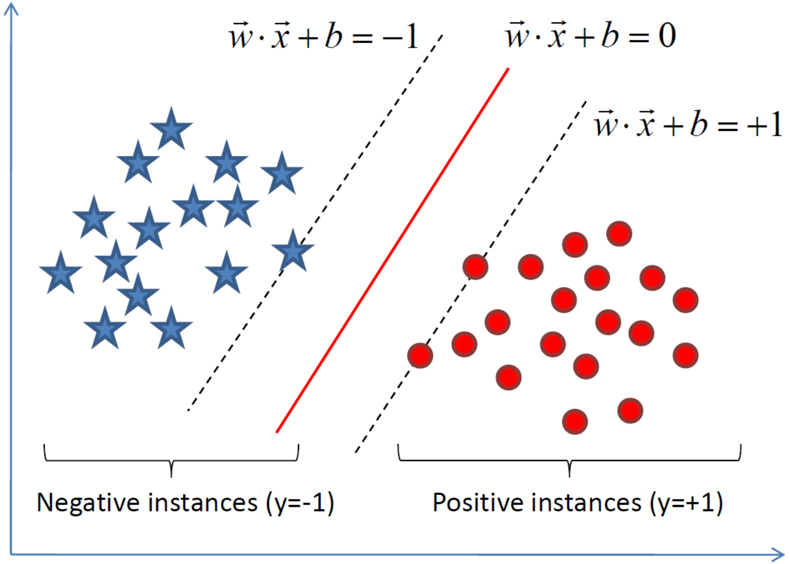
- Want to find a classifier (hyperplane) to separate negative instances from the positive ones.
- An infinite number of such hyperplanes exist.
- SVMs finds the hyperplane that maximizes the gap between data points on the boundaries (so-called support vectors).
- If the points on the boundaries are not informative (e.g., due to noise), SVMs will not do well.

- (Case 1) Linearly Separable Data; Hard-Margin Linear SVM



- The gap is distance between two parallel hyperplanes: $\vec{w}\cdot\vec{x}+b=-1$ and $\vec{w}\cdot\vec{x}+b=+1$
- Now, we know that $D=\frac{|b_1-b_2|}{\|\vec{w}\|}$, i.e., $D=\frac{2}{\|\vec{w}\|}$.
- Since we have to maximize the gap, we have to minimize $\|\vec{w}\|$.
- Or equivalently, we have to minimize $\frac{1}{2}\|\vec{w}\|^2$.

- (Case 1) Linearly Separable Data; Hard-Margin Linear SVM

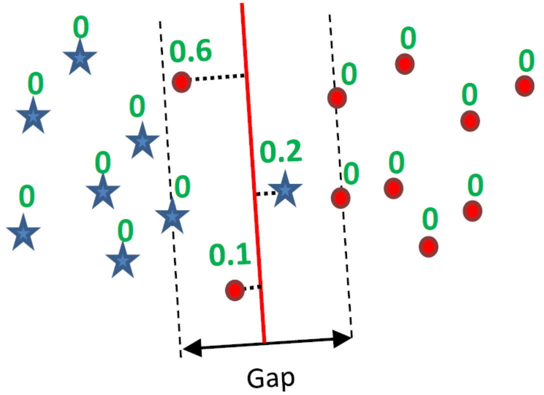


- In addition, we need to impose constrains that all instances are correctly classified. In our case, $\vec{w}\cdot\vec{x}_i+b\le -1$ if $y_i=-1$ $\vec{w}\cdot\vec{x}_i+b\ge +1$ if $y_i=+1$., i.e., equivalently, $y_i(\vec{w}\cdot\vec{x}_i+b)\ge 1$.

In summary,

- Minimize $\frac{1}{2}\|\vec{w}\|^2$ subject to $y_i(\vec{w}\cdot\vec{x}_i+b)\ge 1$, for $i=1,\cdots,N$

- (Case 2) Not Linearly Separable Data; Soft-Margin Linear SVM



- What if the data is not linearly separable? E.g., there are outliers or noisy measurements, or the data is slightly non-linear.

Approach

- Assign a <u>slack variable</u> to each instance $\xi_i \geq 0$, which can be thought of distance from the separating hyperplane if an instance is misclassified and 0 otherwise.
- Minimize $\frac{1}{2}||\vec{w}||^2 + C\sum_{i=1}^{N}\xi_i$ subject to $y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \xi_i$, for $i = 1, \cdots, N$

- (Case 3) Not Linearly Separable Data; Kernel Trick

Data is not linearly separable in the input space	Data is linearly separable in the feature space obtained by a kernel