Hard-Margin Support Vector Machines

Linear Classifiers

We've already seen several:

- Logistic Regression
- Perceptron

Def: A linear classifier makes decisions by computing a linear combination of features of the form $\mathbf{w}^T\mathbf{x} + b$ and then classifies based on

$$\mathbf{w}^T\mathbf{x} + b \geq 0$$

The boundary between the two classes is defined by $\mathbf{w}^T\mathbf{x} + b = 0$ and is called the separator or the decision boundary

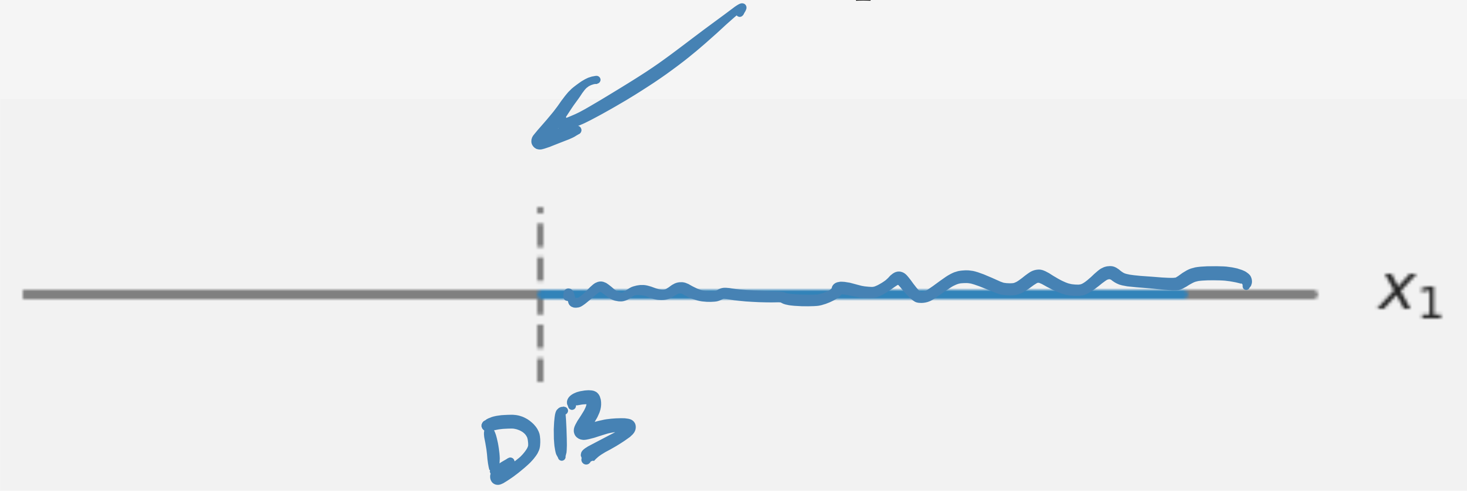
We estimate the weights and bias using the training data

Linear Classifiers

In 1D we just have a single feature, x_1

The decision rule $\mathbf{w}^T\mathbf{x} + b \geq 0$ becomes $w_1 x_1 + b \geq 0$ or, $x_1 \geq -\frac{b}{w_1}$

The decision boundary is just a single point $x_1 = -\frac{b}{w_1}$



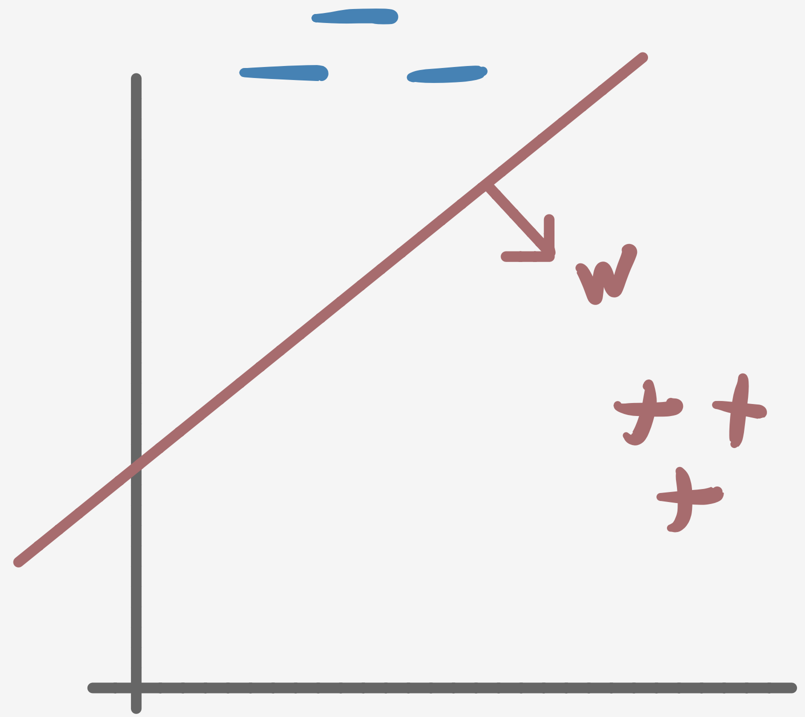
Linear Classifiers

$y \in \{1, -1\}$

In 2D we just have two features, x_1 and x_2

The decision rule $\mathbf{w}^T\mathbf{x} + b \geq 0$ becomes $w_1 x_1 + w_2 x_2 + b \geq 0$

The decision boundary is the line $x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2}$



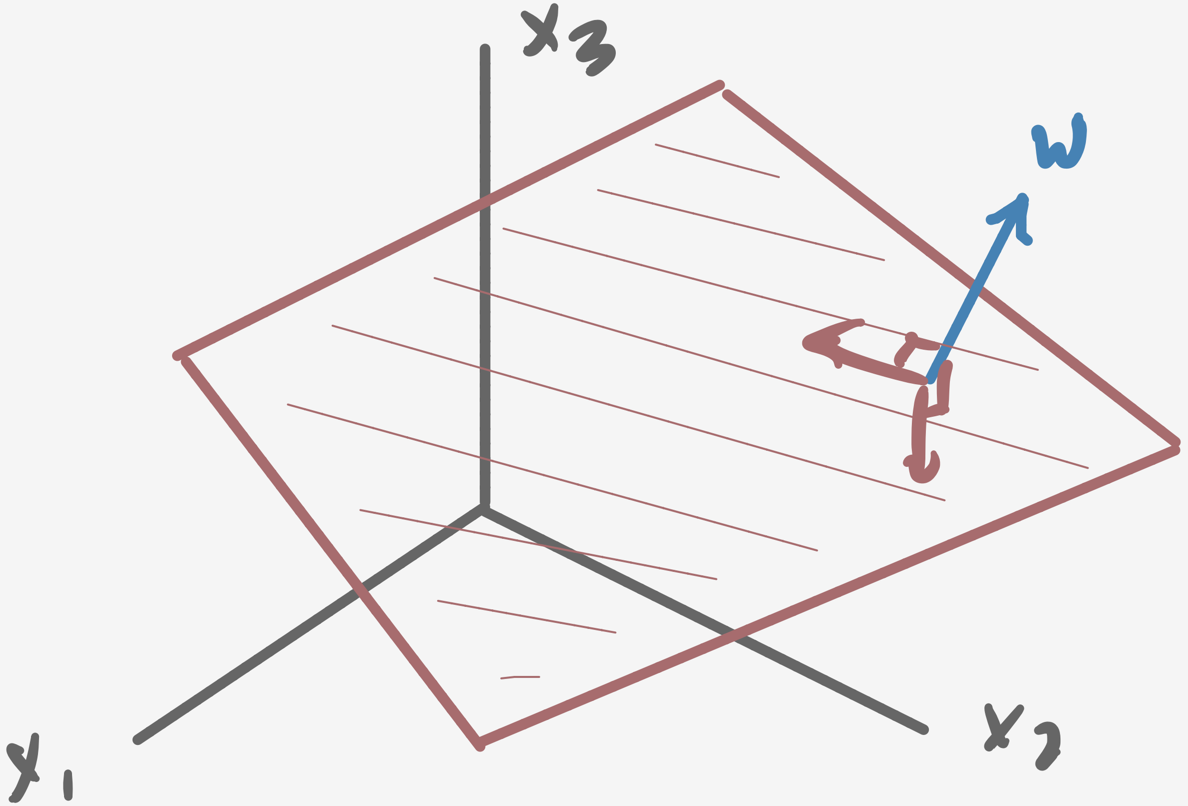
Note that the vector w is orthogonal to the DB

and points in the direction of the positive class

Linear Classifiers

In 3D we just have three features, x_1, x_2, and x_3

The decision rule $\mathbf{w}^T\mathbf{x} + b \geq 0$ gives the plane $\mathbf{w}^T\mathbf{x} + b = 0$ as the decision boundary



Linear Classifiers

In higher dimensions it becomes harder to visualize

We call the decision boundary the **separating hyperplane**

Different types of linear classifiers determine their parameters in different ways

- Naïve Bayes – Model joint probabilities and use Bayes Rule
- Logistic Regression – Model probability that an example belongs to a particular class
- The Perceptron – Use examples to update parameters until we find something that works

Support Vector Machines use a more geometric idea called a **Margin**

Support Vector Machines

Advantages:

- "Best off-the-shelf classifier" – Andrew Ng
- Nice theoretical bounds
- Allows for nonlinear classification
- Optimization problem for learning parameters is convex

Disadvantages:

- No probabilistic interpretation
- Can be prone to overfitting in the nonlinear case

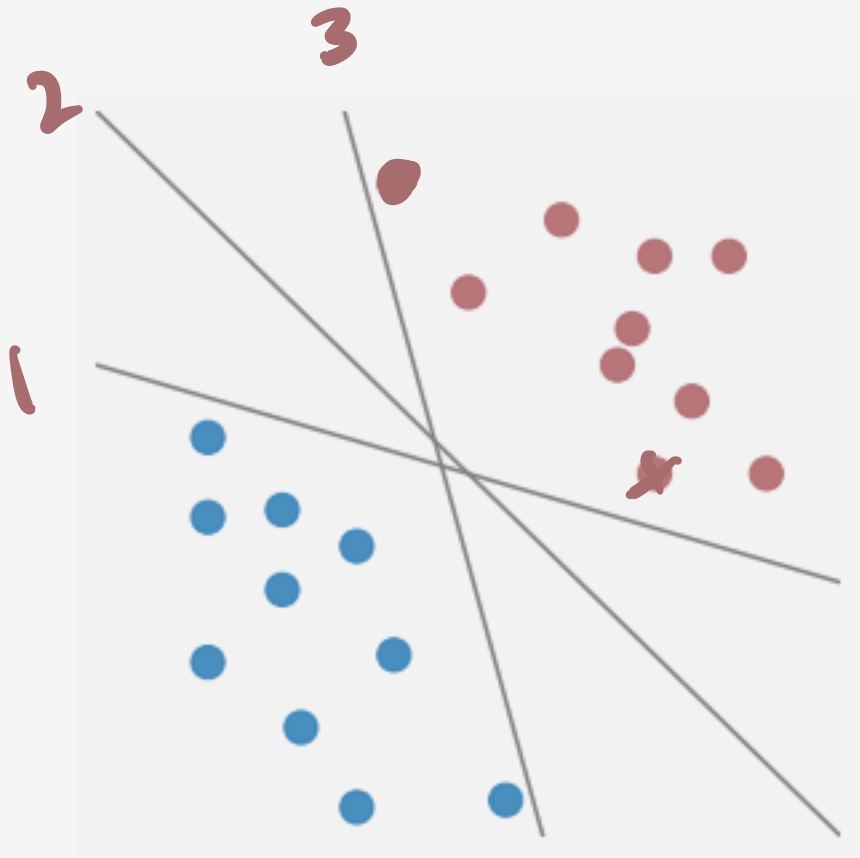
SVM Intuition

Temporary Assumption: Assume the training data is linearly separable

Consider the following training set in 2D

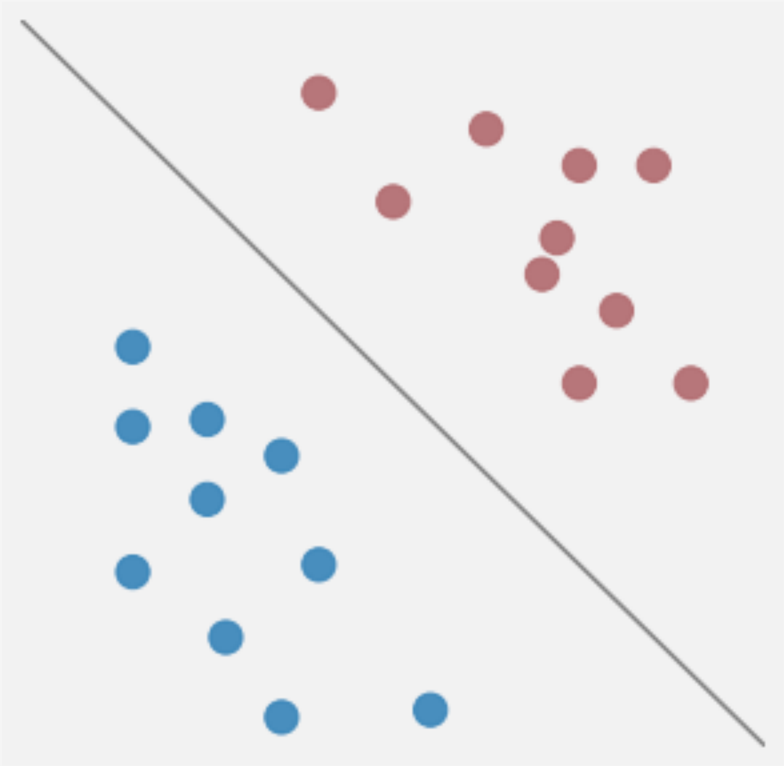
SVM Intuition

Question: Which decision boundary seems **better** to you?



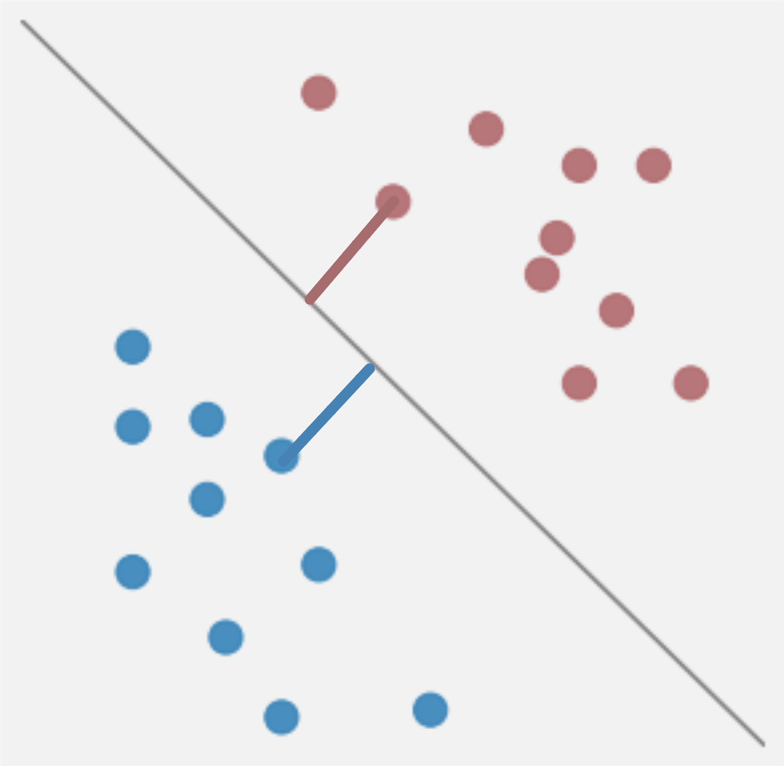
SVM Intuition

Answer: This one feels better



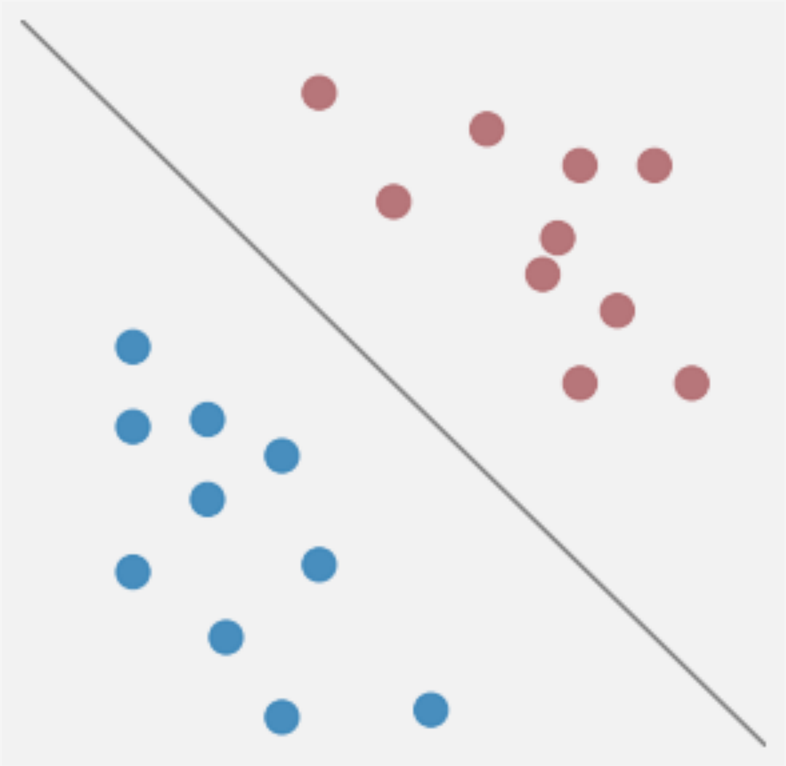
SVM Intuition

More space between DB and closest training examples



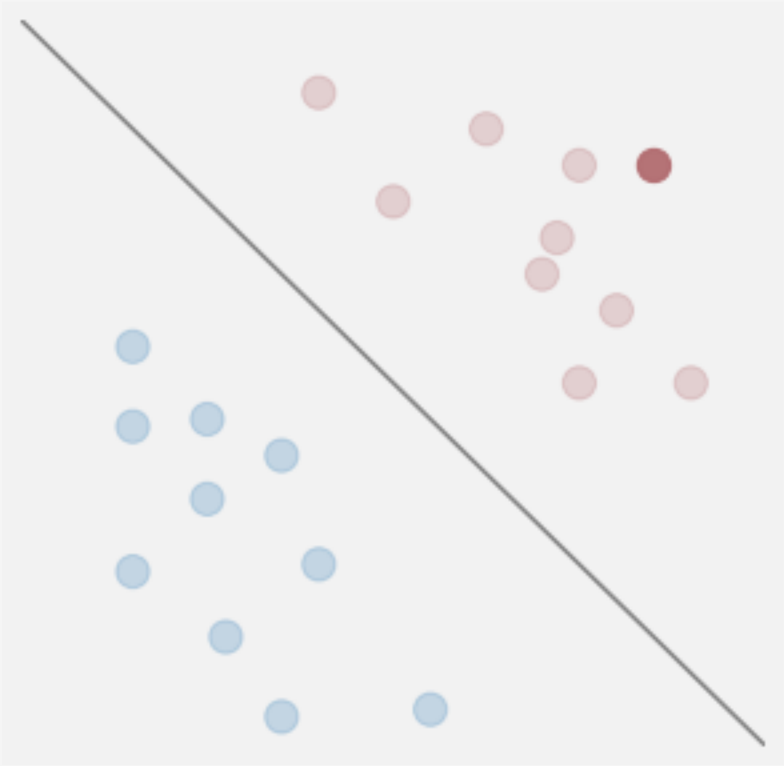
SVM Intuition

Think about the size of $\mathbf{w}^T\mathbf{x}_i + b$ as a measure of confidence in classification of example $\mathbf{x}_i$



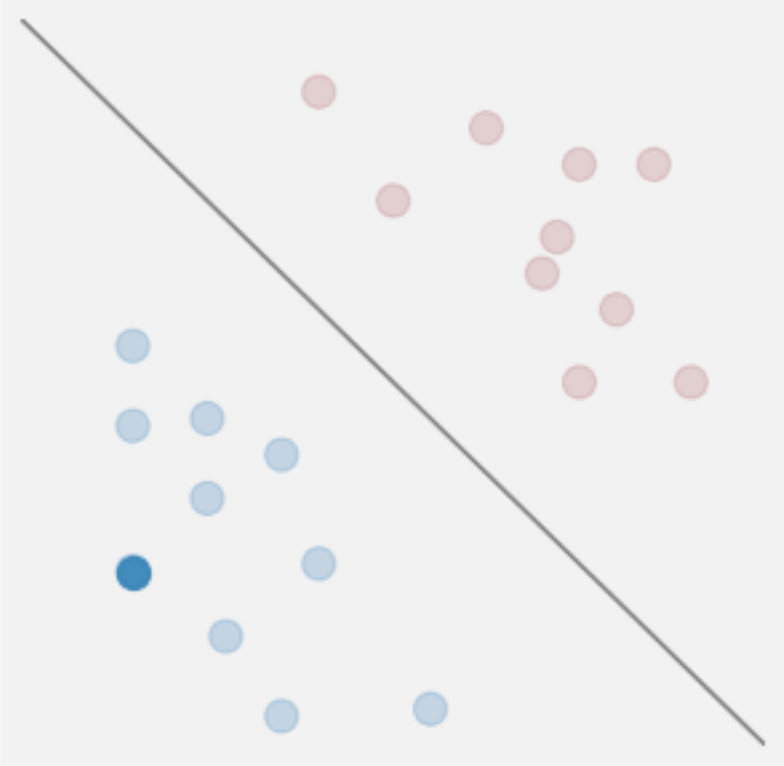
SVM Intuition

Very confident $y_i = 1$ if $\mathbf{w}^T\mathbf{x}_i + b \gg 0$



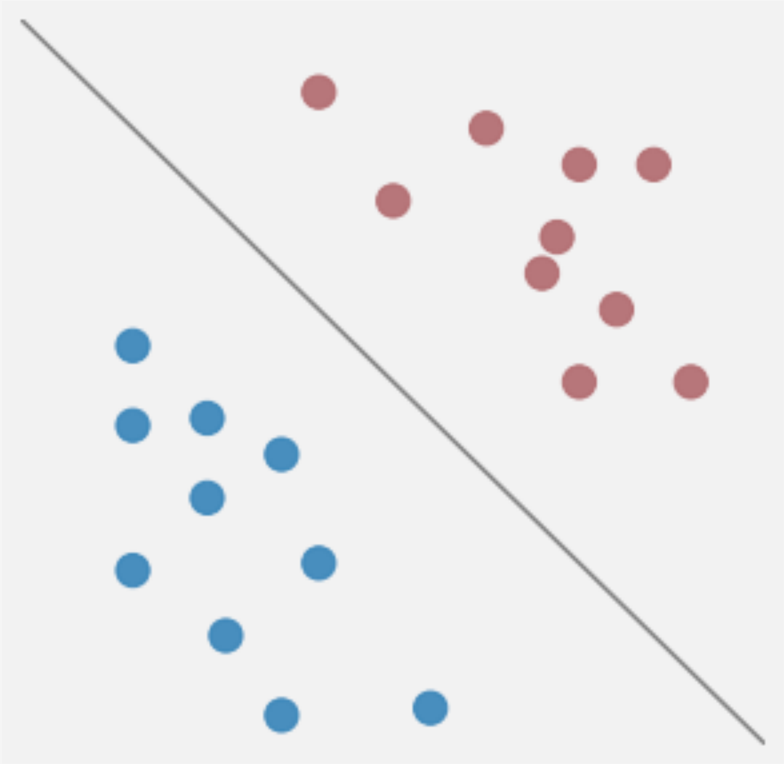
SVM Intuition

Very confident $y_i = -1$ if $\mathbf{w}^T\mathbf{x}_i + b \ll 0$



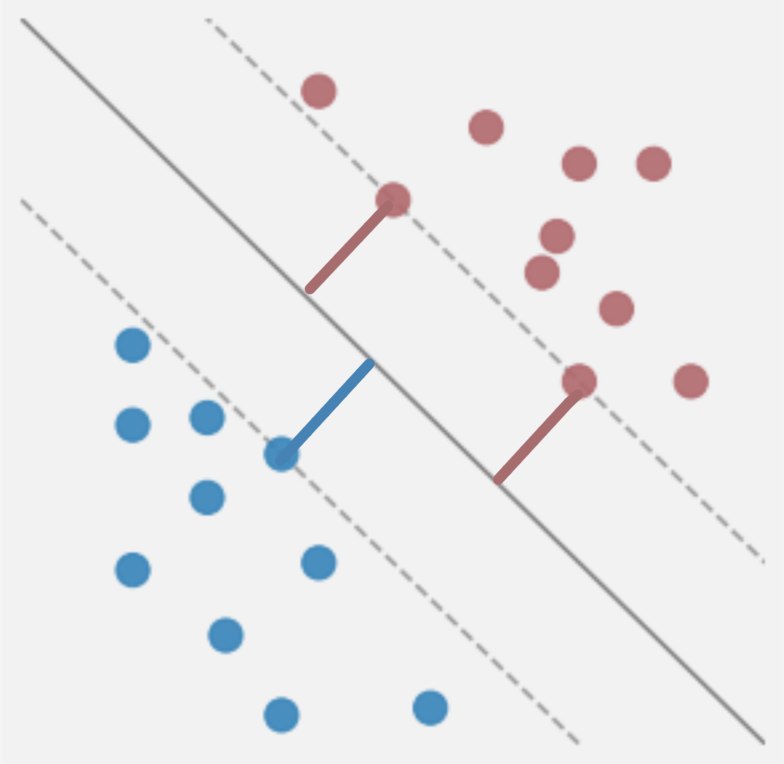
SVM Intuition

Want a DB that makes us very confident in the classification of each training example



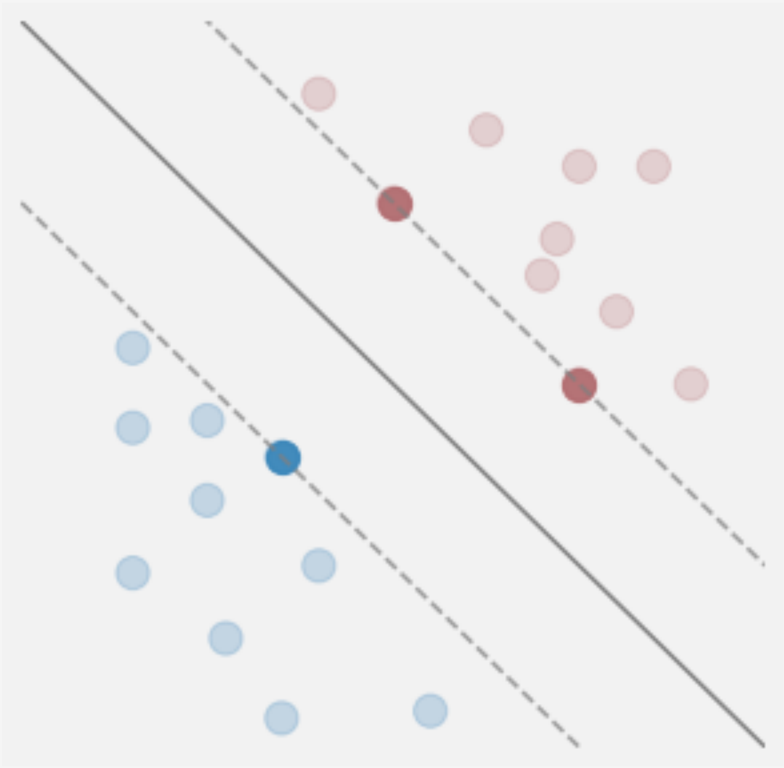
SVM Intuition

Do this by maximizing distance from DB to *closest* training examples



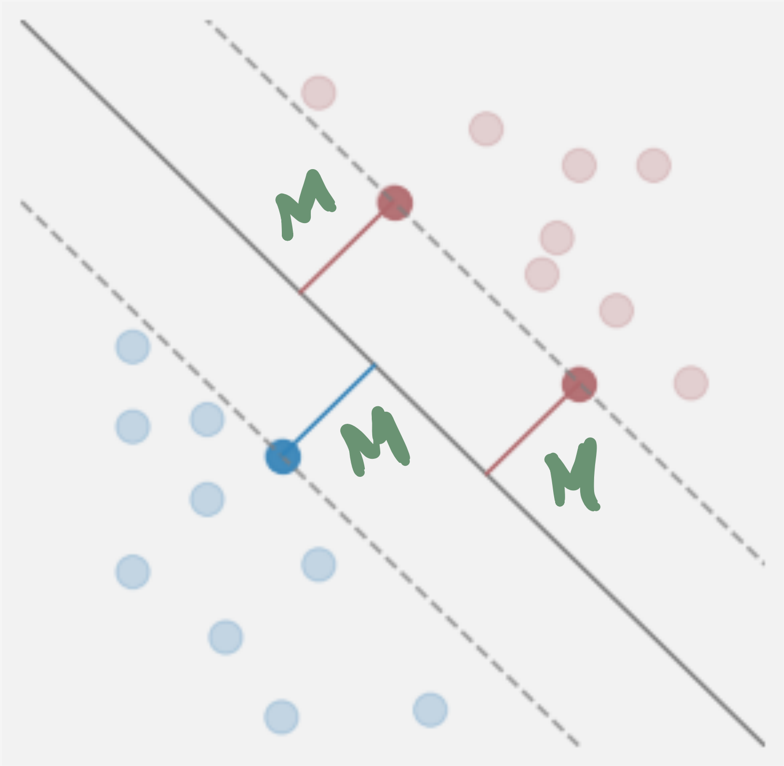
SVM Intuition

Do this by maximizing distance from DB to *closest* training examples



SVM Intuition

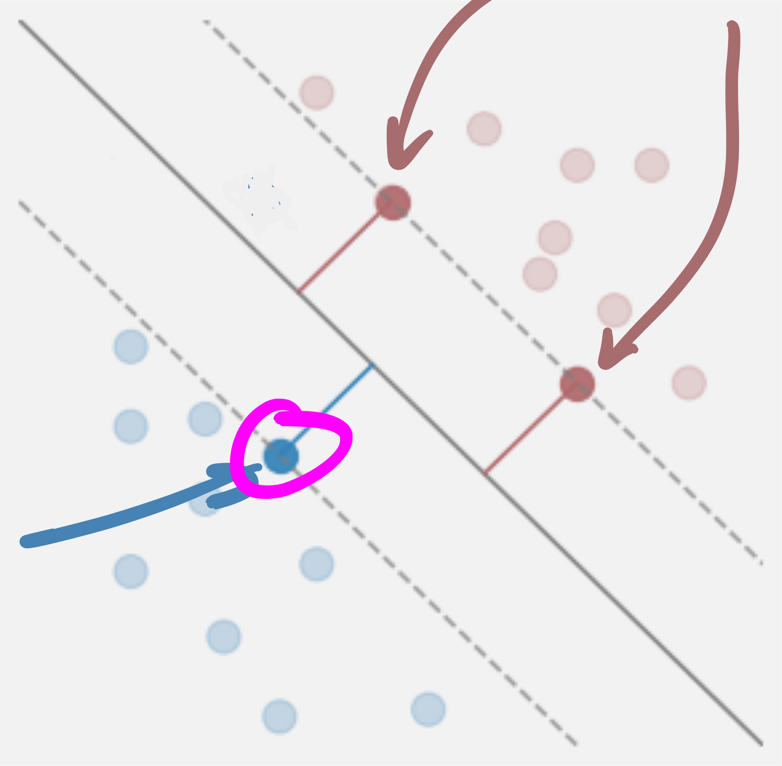
This distance from the DB to closest training examples is called the **Margin**



SVM Intuition

The points closest to the DB are called the **support vectors**

+VE SUPPORT VECTORS



NEG SUPPORT VECTOR

Maximum Margin Classifier

How do we do this **mathematically?**

Let the margin be represented by M

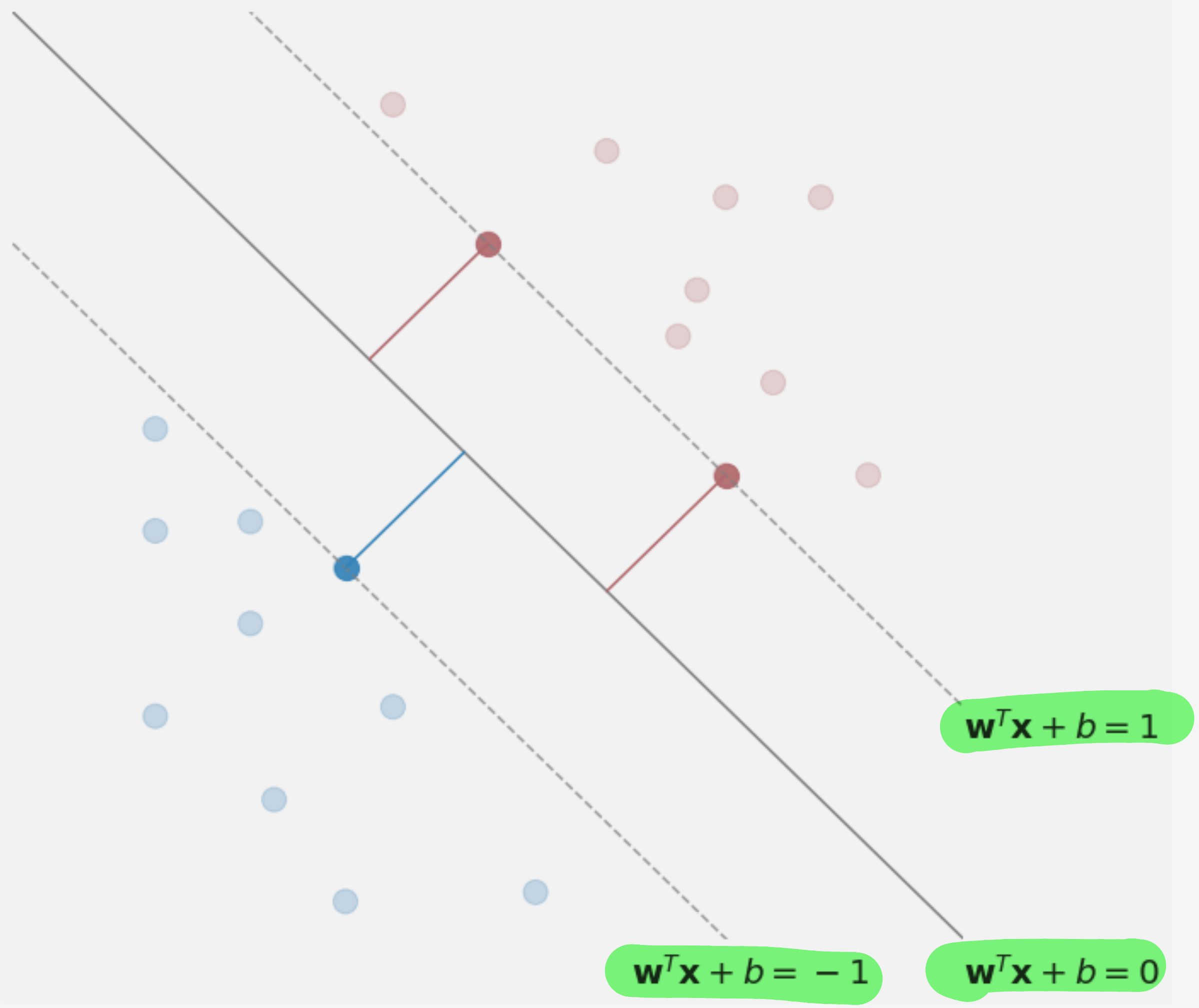
Goal: Determine $\mathbf{w}$ and b that maximize M while classifying all training points correctly

This gives the following (informal) optimization problem

$$\begin{array}{ll} \max\limits_{\mathbf{w},\ b} & M \\ \text{s.t.} & \text{All training examples correctly classified} \end{array}$$

Maximum Margin Classifier

How do we represent M **mathematically?** First, support vector boundaries as $\mathbf{w}^T\mathbf{x} + b = \pm 1$



Maximum Margin Classifier

Does this make sense?

We can probably agree that the SV boundaries should at least look like $\mathbf{w}^T\mathbf{x} + b = \pm K$

Now notice that the DB $\mathbf{w}^T\mathbf{x} + b = 0$ is unaffected by scaling of $\mathbf{w}$ and b

$$(2\mathbf{w})^T\mathbf{x} + 2b = 0 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} + b = 0$$

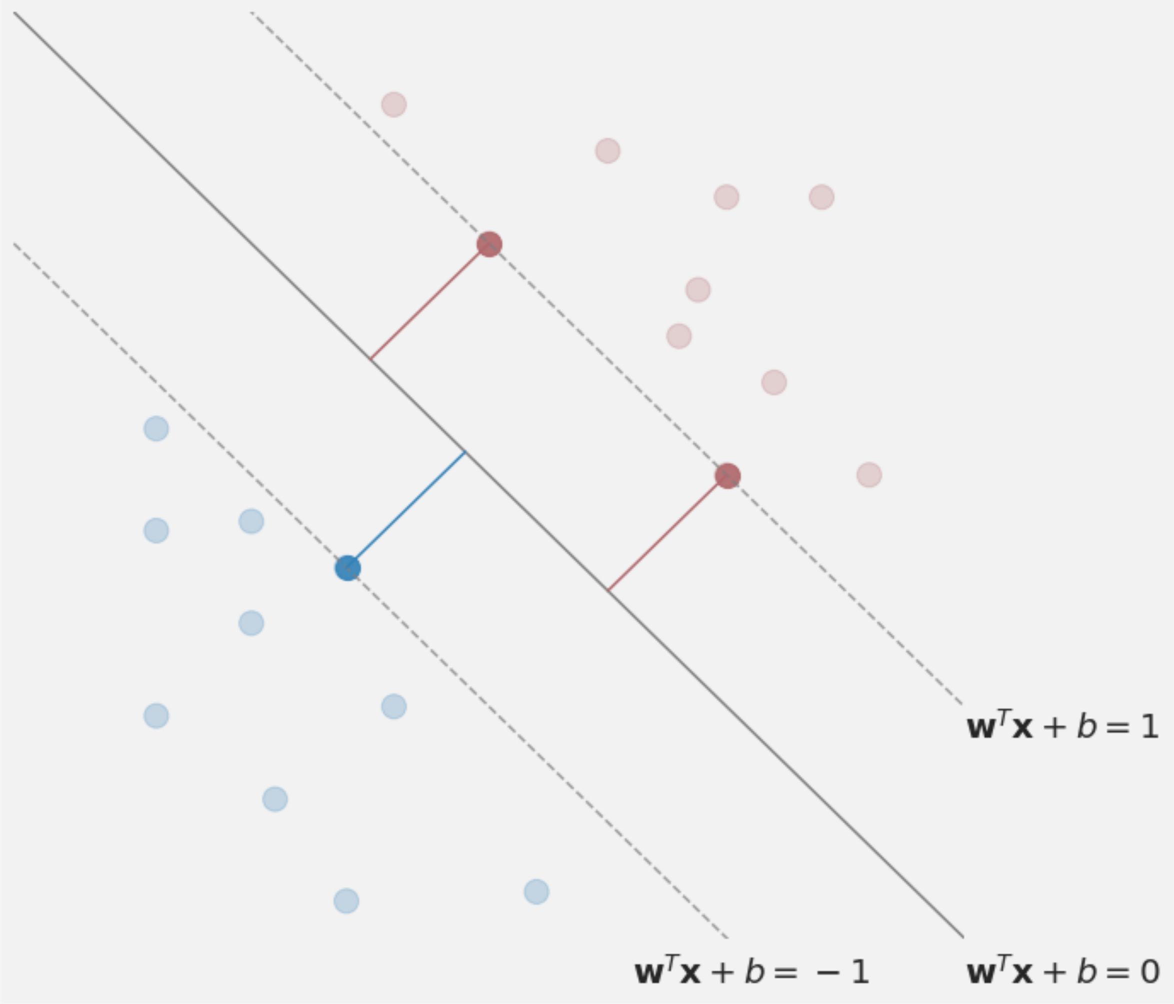
Divide both sides of $\mathbf{w}^T\mathbf{x} + b = \pm K$ by K

$$\left(\frac{\mathbf{w}}{K}\right)^T \mathbf{x} + \frac{b}{K} = \pm 1$$

Absorb K into $\mathbf{w}$ and b to obtain $\mathbf{w}^T\mathbf{x} + b = \pm 1$

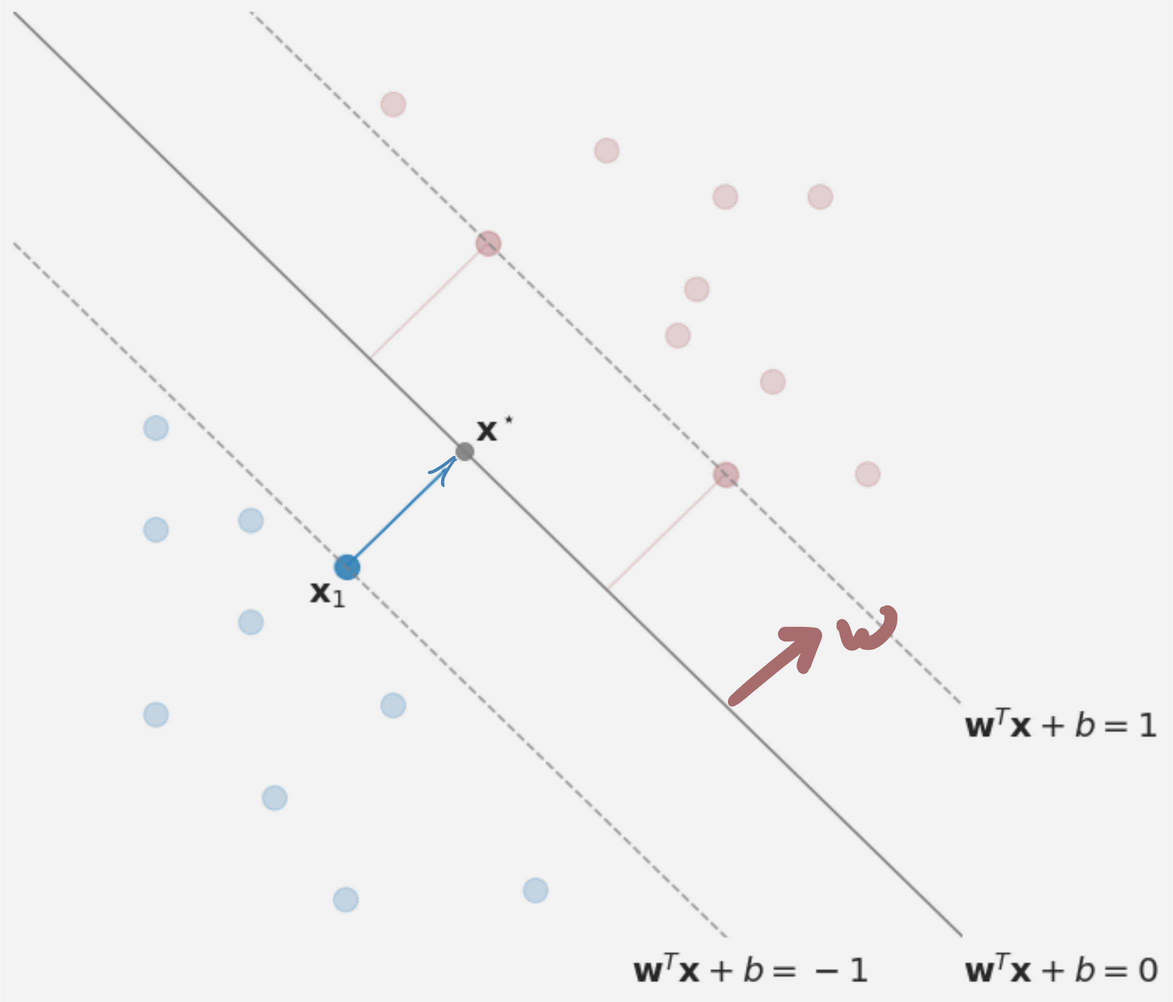
Maximum Margin Classifier

OK, so we agree on the functional form of the decision and support vector boundaries



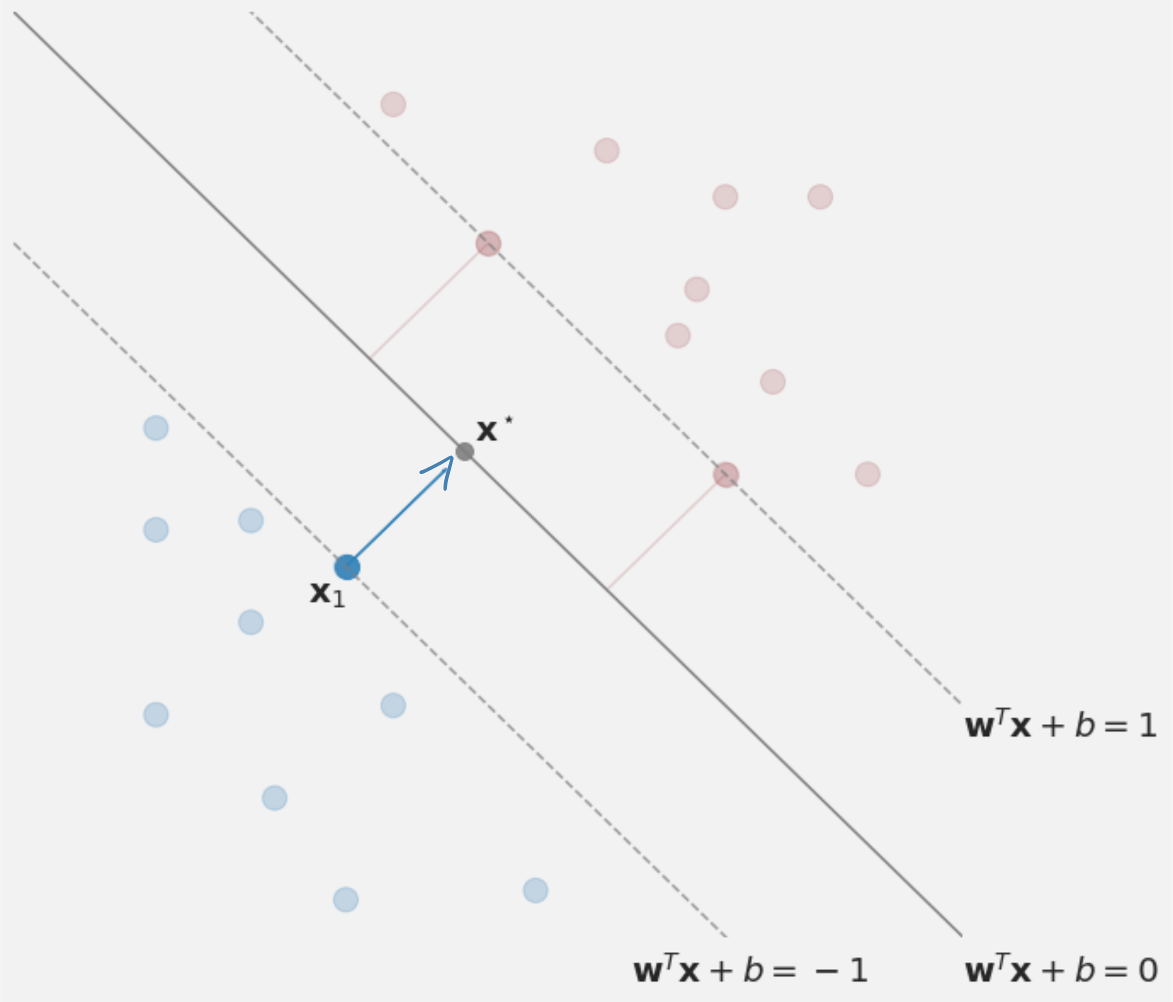
Maximum Margin Classifier

To find M we need to do some vector arithmetic



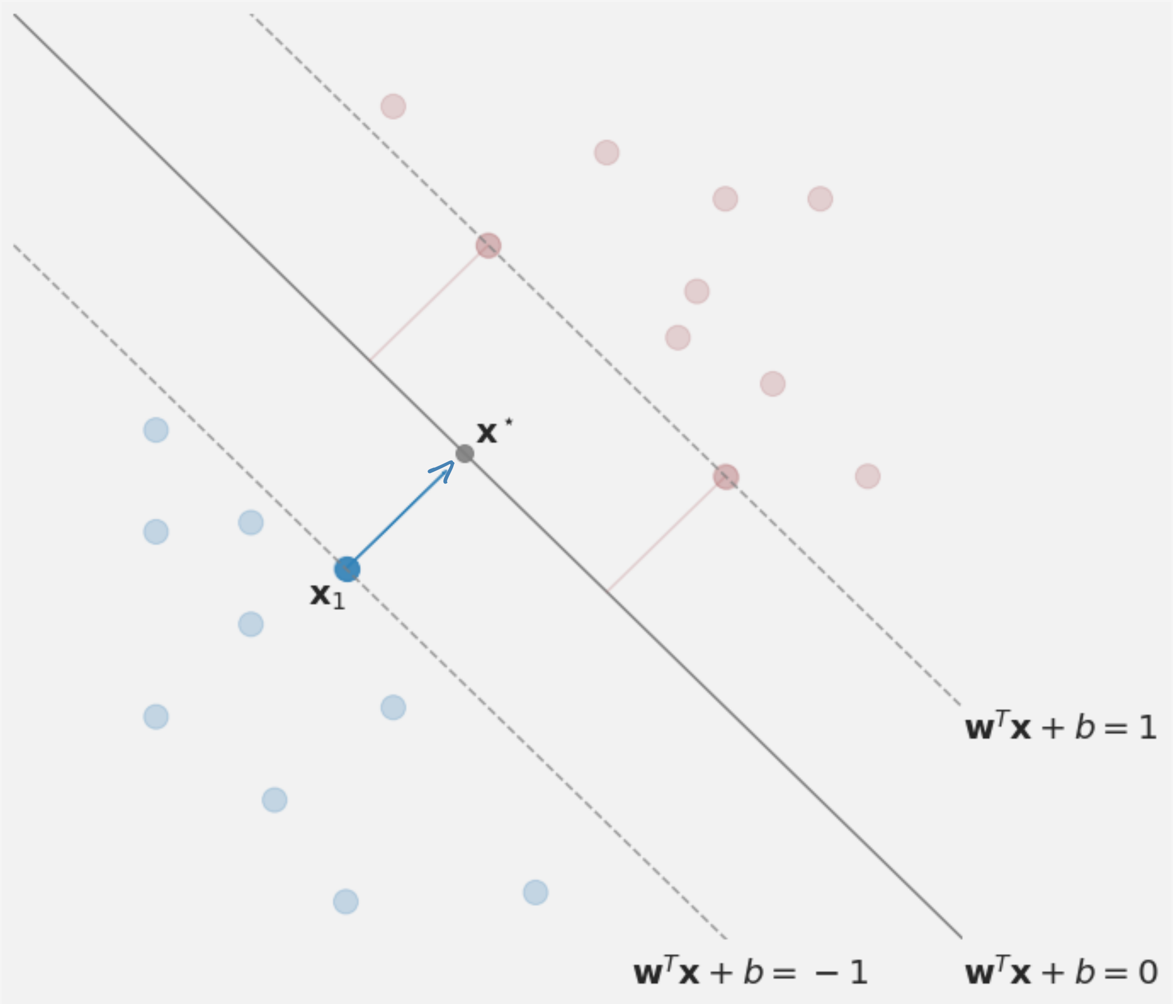
Maximum Margin Classifier

Choose support vector $\mathbf{x}_1$ and it's closest point on the decision boundary, $\mathbf{x}^\star$



Maximum Margin Classifier

Get to $\mathbf{x}^\star$ by moving from $\mathbf{x}_1$ some distance in the direction of $\mathbf{w}$: $\mathbf{x}^\star = \mathbf{x}_1 + \lambda \mathbf{w}$



Maximum Margin Classifier

Get to $\mathbf{x}^\star$ by moving from $\mathbf{x}_1$ some distance in the direction of $\mathbf{w}$: $\mathbf{x}^\star = \mathbf{x}_1 + \lambda \mathbf{w}$

The distance moved is the margin M

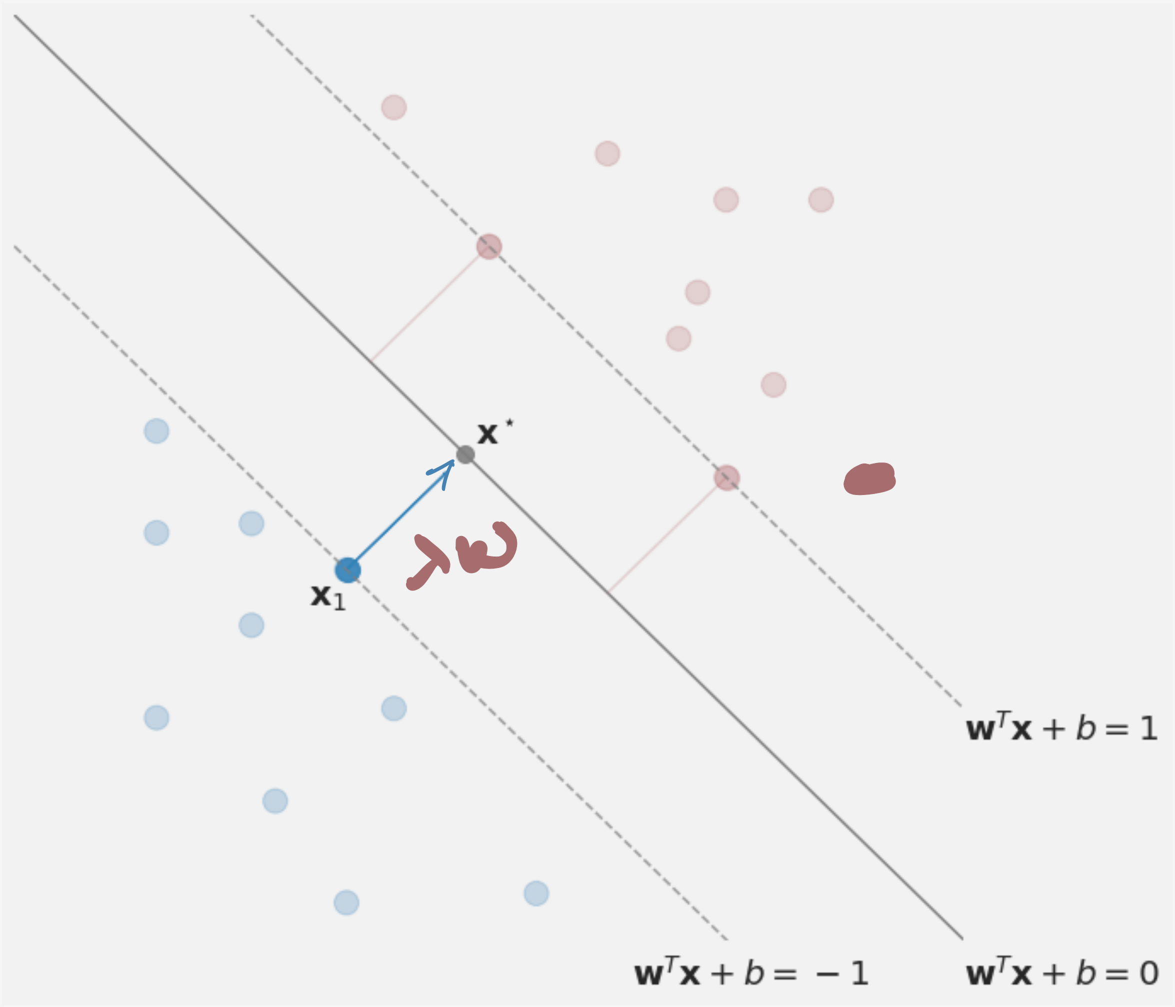
$$M = \|\lambda \mathbf{w}\|_2 = \boxed{\lambda \|\mathbf{w}\|_2 = M}$$

The point $\mathbf{x}^\star$ is on the DB

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

The SV $\mathbf{x}_1$ is on the negative SV boundary

$$\mathbf{w}^T \mathbf{x}_1 + b = -1$$



Maximum Margin Classifier

$\boxed{M = \lambda \|w\|}$

$w^T w = \|w\|_2^2$

We now have the equations:

$$
\begin{aligned}
\mathbf{x}^\star &= \mathbf{x}_1 + \lambda \mathbf{w} & (1)\\
\mathbf{w}^T\mathbf{x}^\star + b &= 0 & (2)\\
\mathbf{w}^T\mathbf{x}_1 + b &= -1 & (3)
\end{aligned}
$$

$M\|w\| = \lambda \|w\|^2$

$M\|w\| = 1$

$\Rightarrow M = \frac{1}{\|w\|_2}$

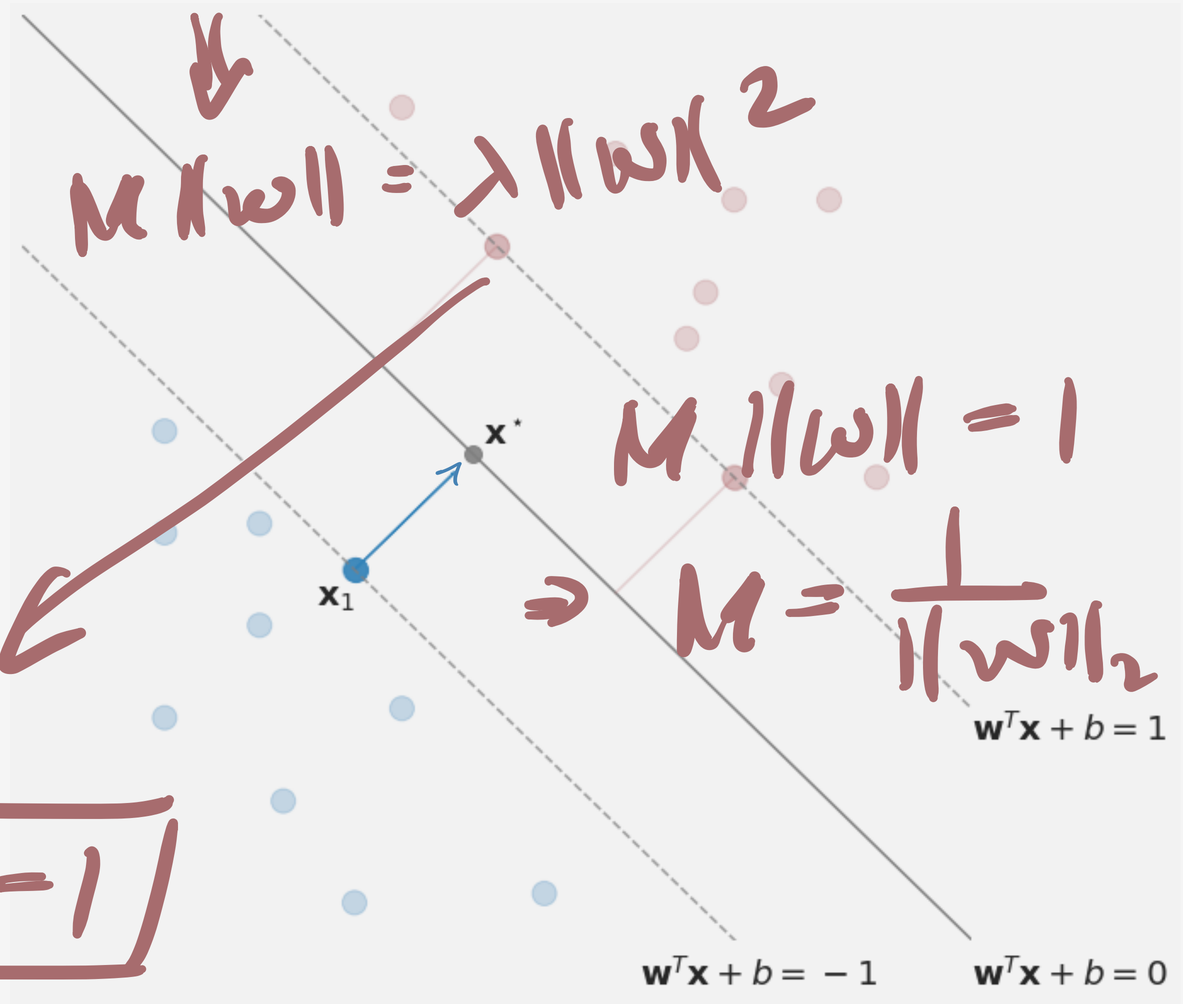
And with some algebra, we have

$w^T(x_1 + \lambda w) + b = 0$

$w^T x_1 + \lambda w^T w + b = 0$

$\lambda w^T w + (w^T x_1 + b) = 0$

$\lambda w^T w - 1 = 0 \Rightarrow \boxed{\lambda \|w\|^2 = 1}$



Maximum Margin Classifier

Mathier Goal:

$$\begin{array}{ll} \max\limits_{\mathbf{w},\ b} & 1/\|\mathbf{w}\|_2 \\ \text{s.t.} & \text{All training examples correctly classified} \end{array}$$

The constraints can be expressed as

$$\begin{array}{rcrl} \mathbf{w}^T\mathbf{x}_i + b & \geq & 1 & \text{for points with } y_i = 1 \\ \mathbf{w}^T\mathbf{x}_i + b & \leq & -1 & \text{for points with } y_i = -1 \end{array}$$

which we can combine into one constraint as

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, m$$

Maximum Margin Classifier

Mathiest Goal:

$$\begin{array}{ll} \max\limits_{\mathbf{w},\ b} & 1/\|\mathbf{w}\|_2 \\ \text{s.t.} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, m \end{array}$$

Solve this constrained optimization problem to learn weights and bias

Note: Rigid enforcement of constraints that all training examples be classified correctly is what makes this a **Hard-Margin** SVM

Problem: This optimization problem is **nasty**

Maximum Margin Classifier

Mathiest Goal:

$$\begin{aligned} \max_{\mathbf{w},\ b} \quad & 1/\|\mathbf{w}\|_2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, m \end{aligned}$$

Problem: The objective function is non-differentiable

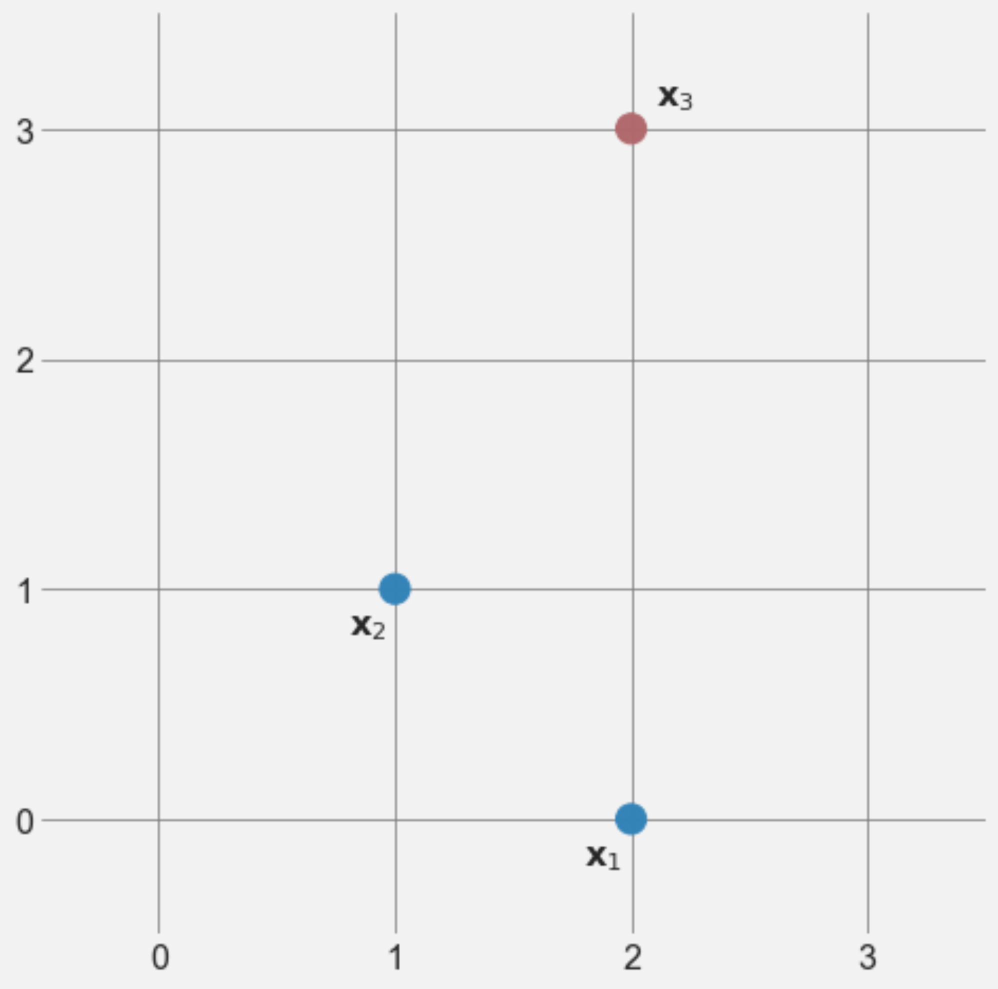
Instead, we'll formulate it as follows

$$\begin{aligned} \min_{\mathbf{w},\ b} \quad & \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \ldots, m \end{aligned}$$

This is a quadratic objective function with linear inequality constraints. Tons of canned software.

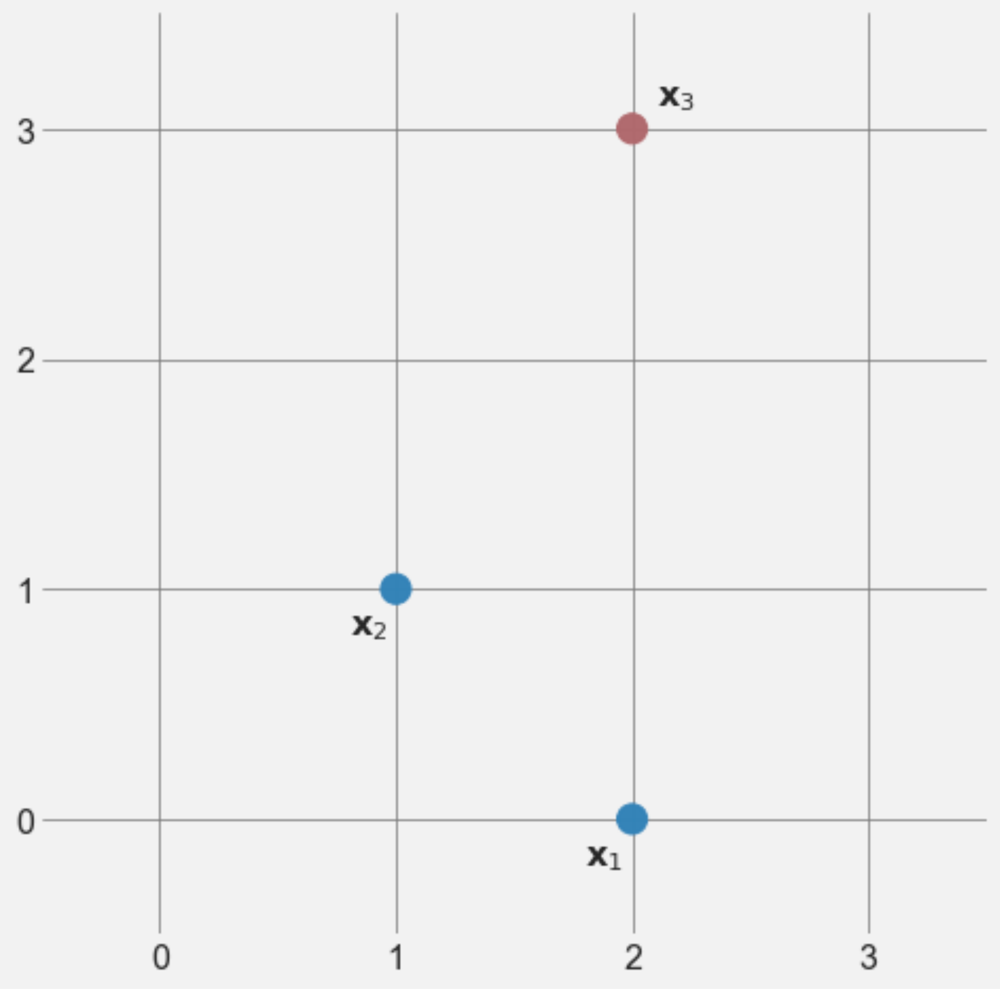
Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data



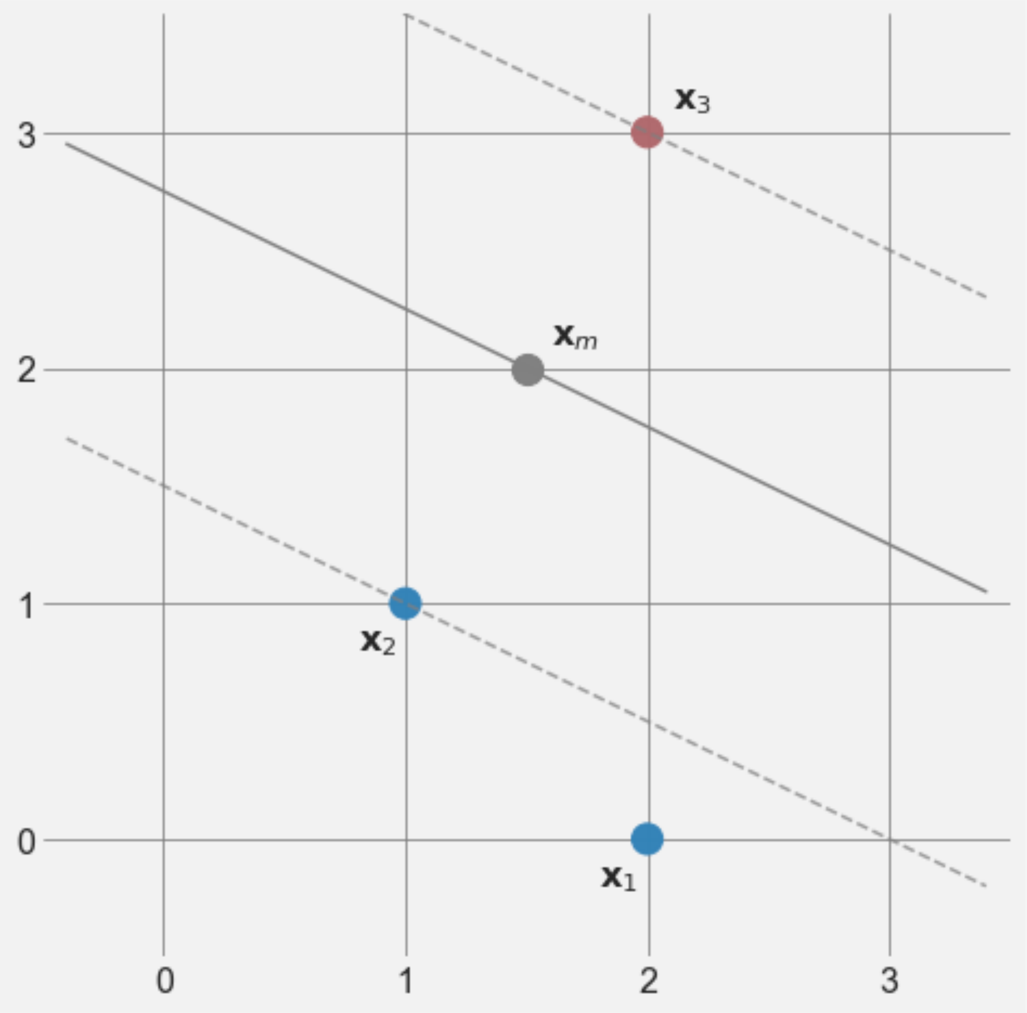
Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data



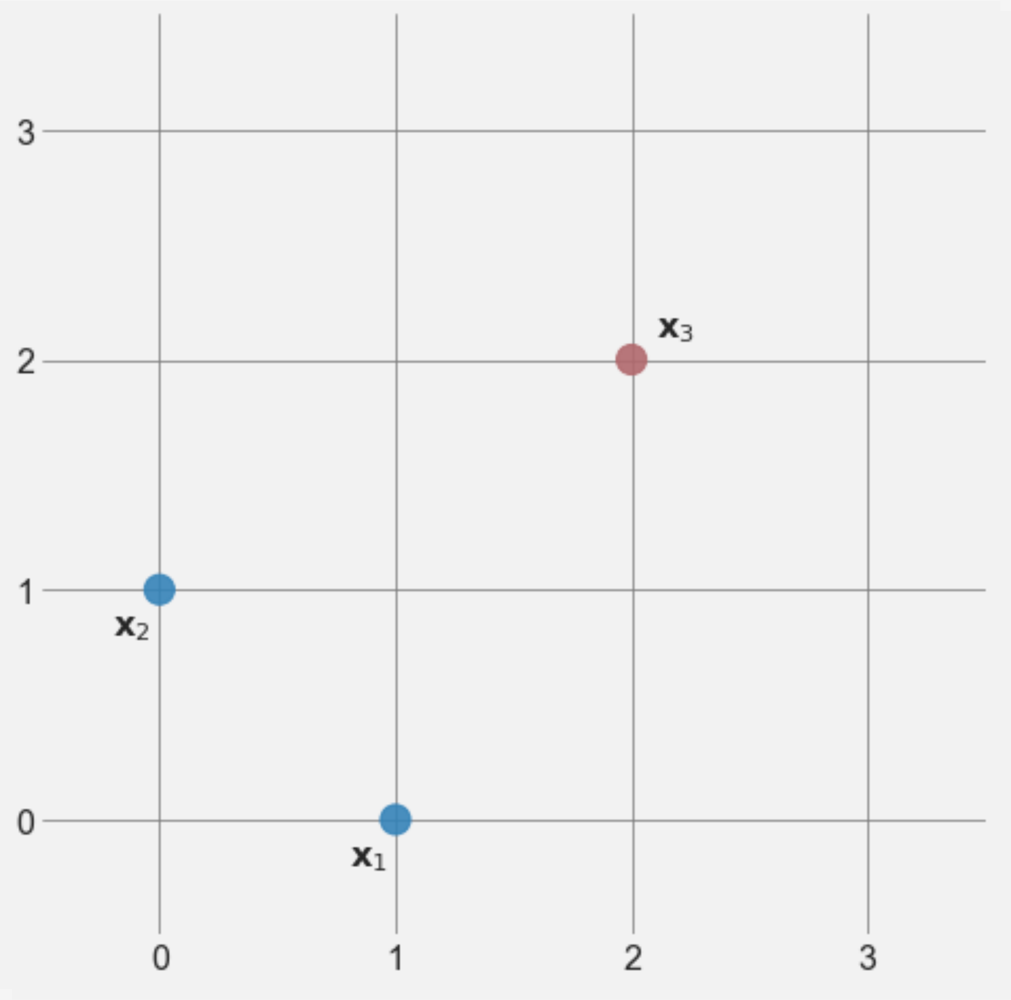
Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data



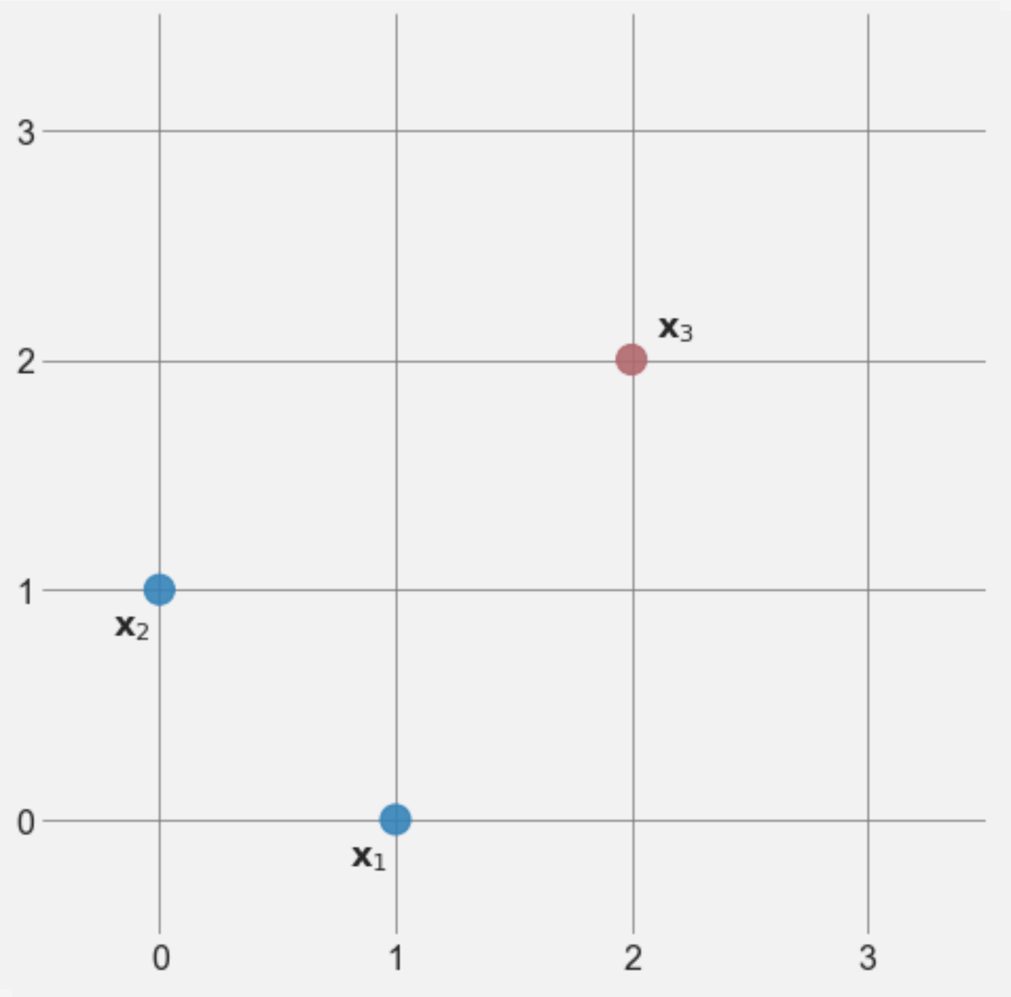
Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data



Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data



Hard-Margin SVM Example

Geometric Example: Find the Hard-Margin SVM of form $\mathbf{w}^T\mathbf{x} + b = 0$ for the following data