Artificial Intelligence II (CS4442B & CS9542B)

Classification: Support Vector Machines

Boyu Wang
Department of Computer Science
University of Western Ontario

Intuitions

Intuitions

- So far, the supervised learning algorithms we have learned (e.g., linear regression, logistic regression, Gaussian discriminant analysis) are based on probabilistic assumptions:
  - We assume that the conditional probability $p(y|x)$ follows some distributions or functions parameterized by w.
  - Then, we choose the parameters by the principle of maximum likelihood.
- Is there any other principle to choose the parameter (e.g., without any probabilistic assumption)?
- A support vector machines (SVM) aims to find the model parameters by the large margin principle.

What is a support vector machine (SVM)

A **supervised learning** technique (mostly for **classification**).

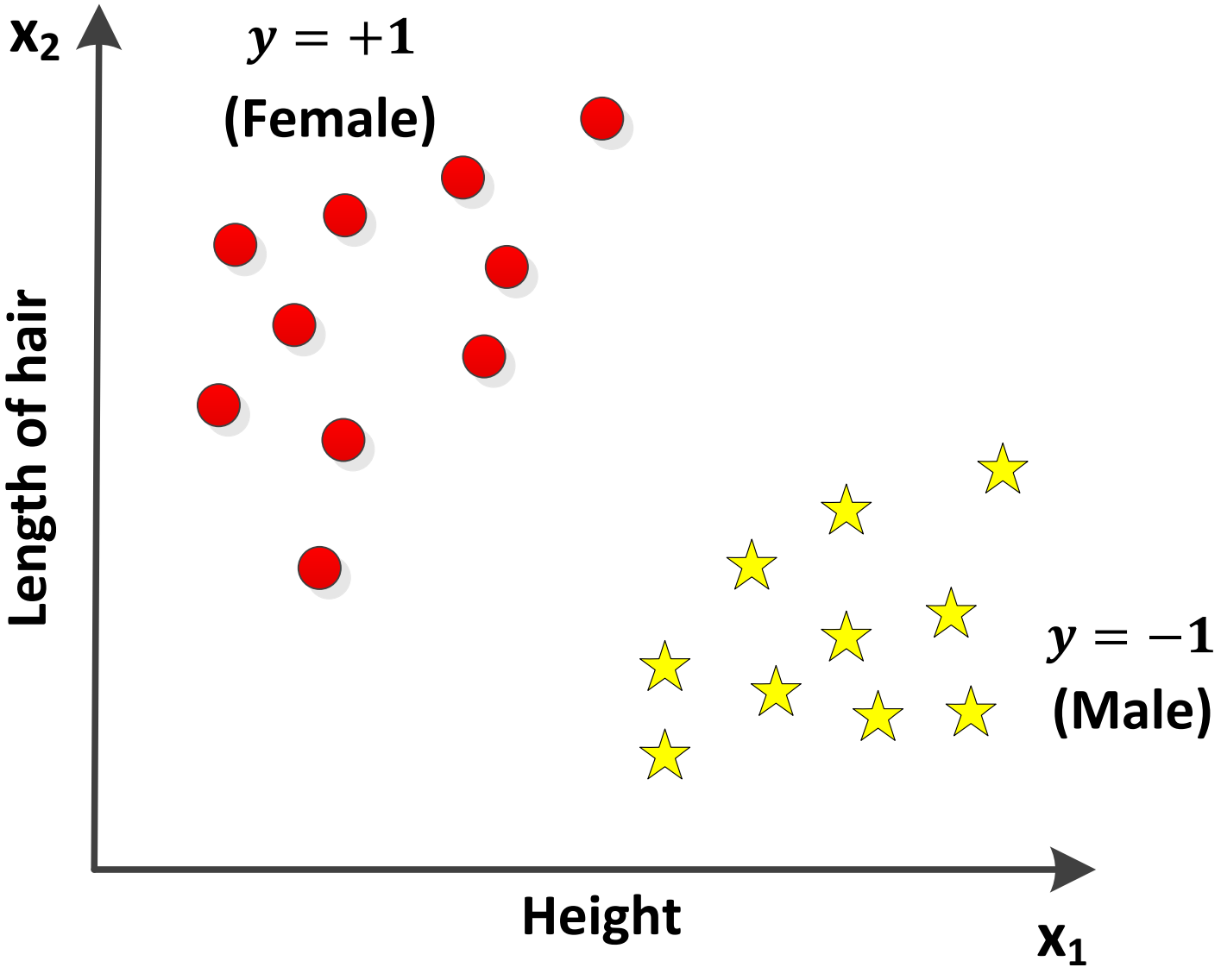
- Goal: learn a mapping $f : \mathcal{X} \to \mathcal{Y}$

f()= female

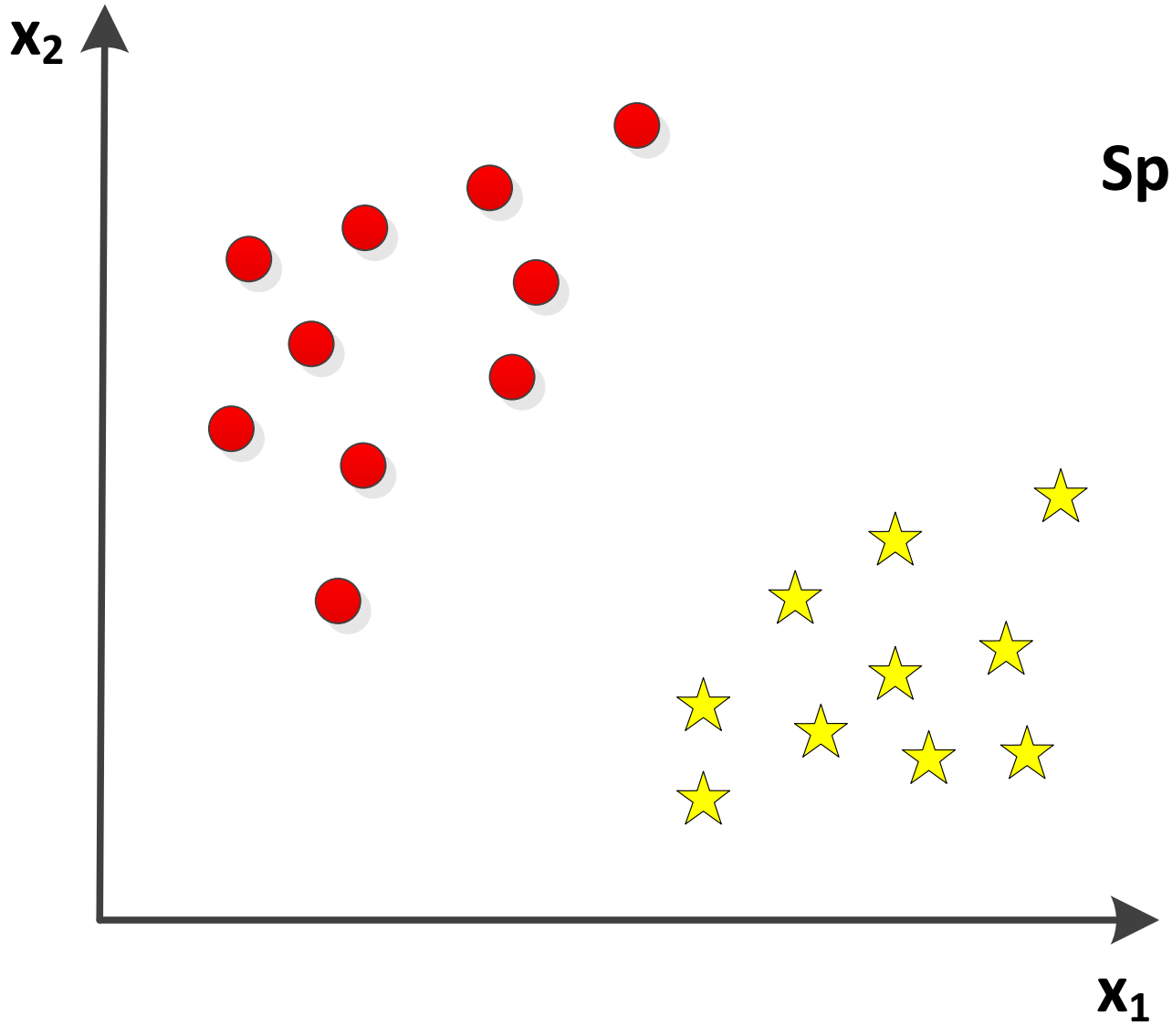
f()= male

- The learning algorithm is trained on a set of examples $(x_1, y_1), \ldots, (x_m, y_m)$, $x \in \mathbb{R}^n$, $y \in \{-1, 1\}$

Linear Separator $w^\top x + b = 0$

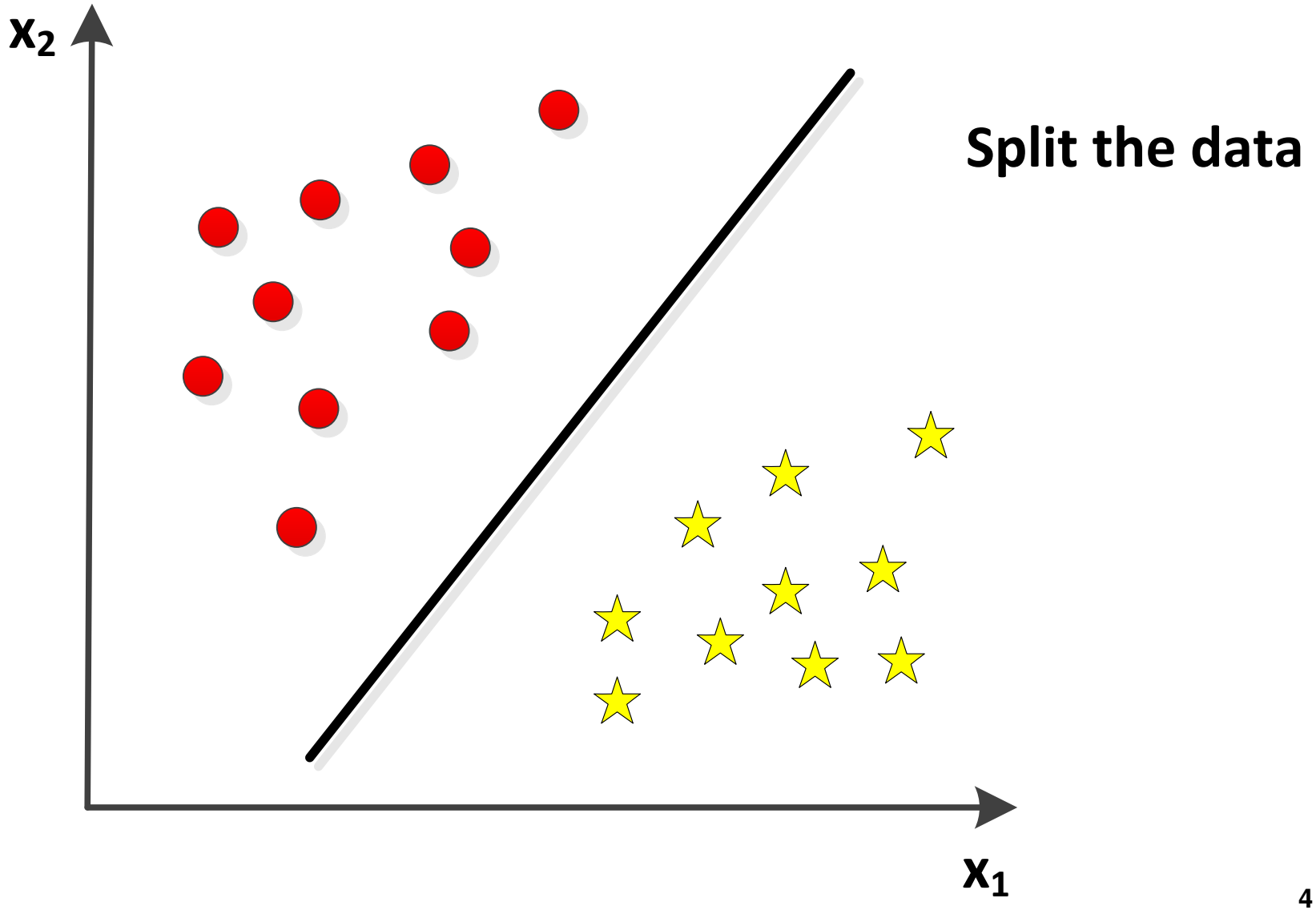


Linear Separator $w^\top x + b = 0$

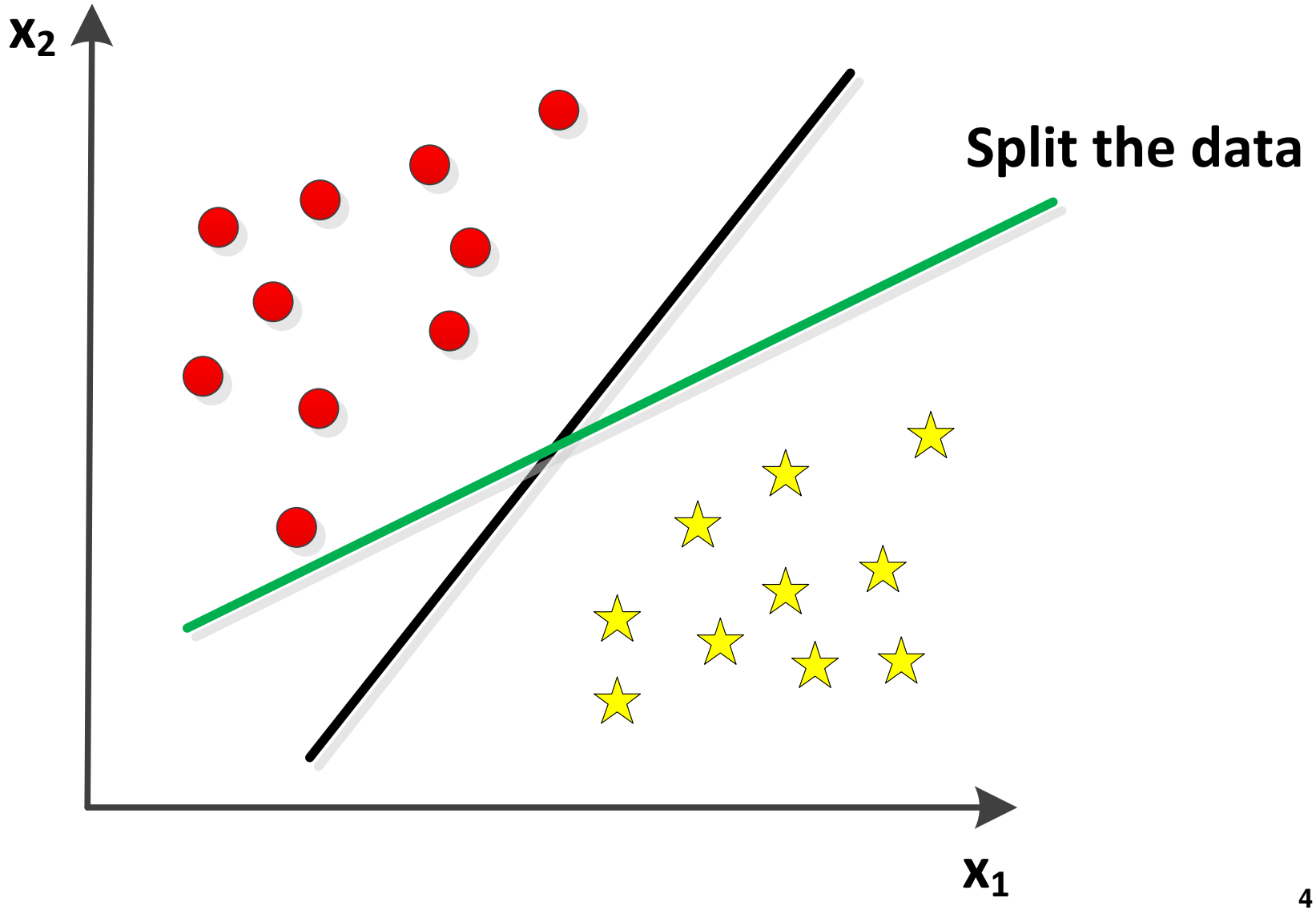


Split the data

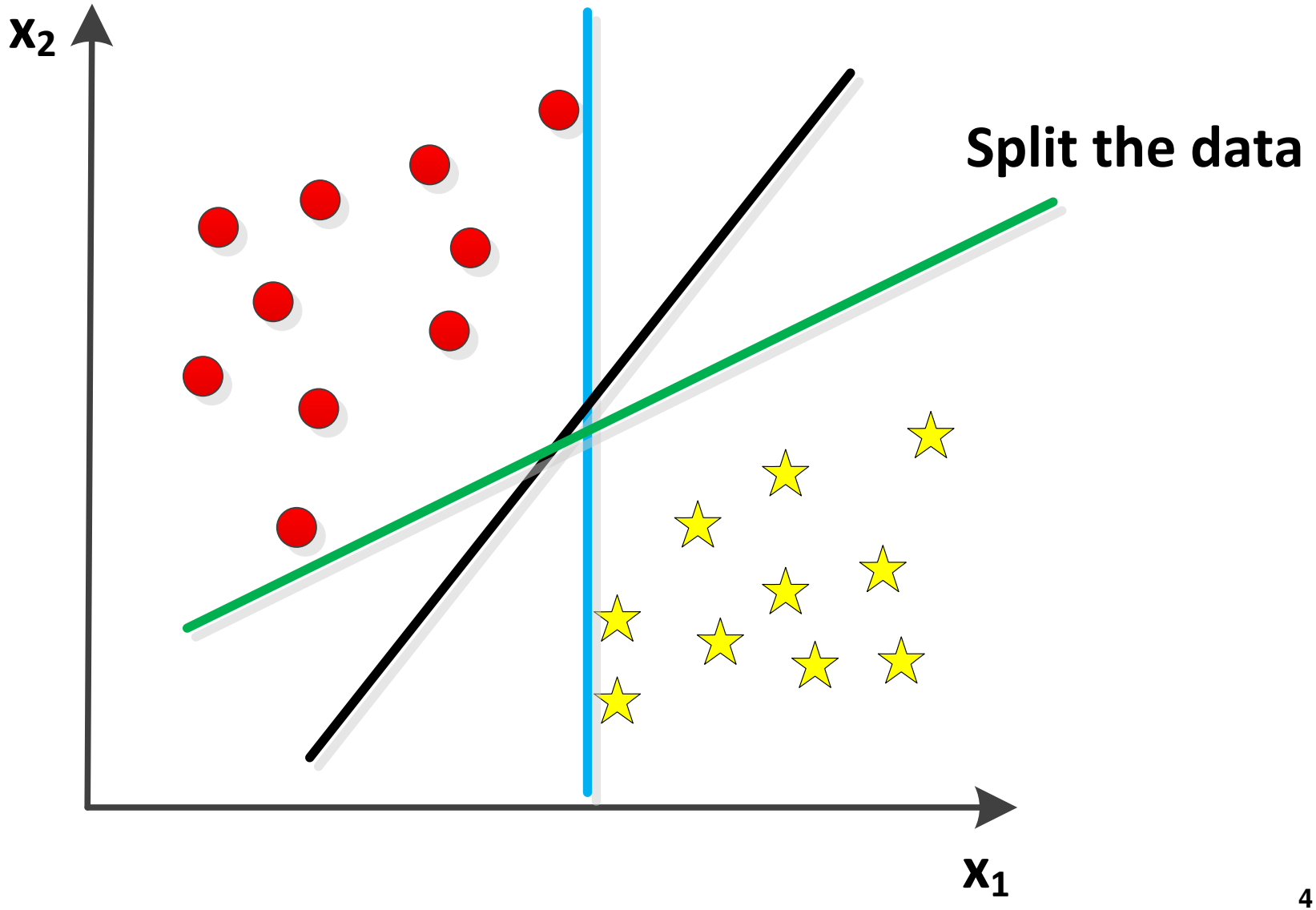
Linear Separator $w^\top x + b = 0$



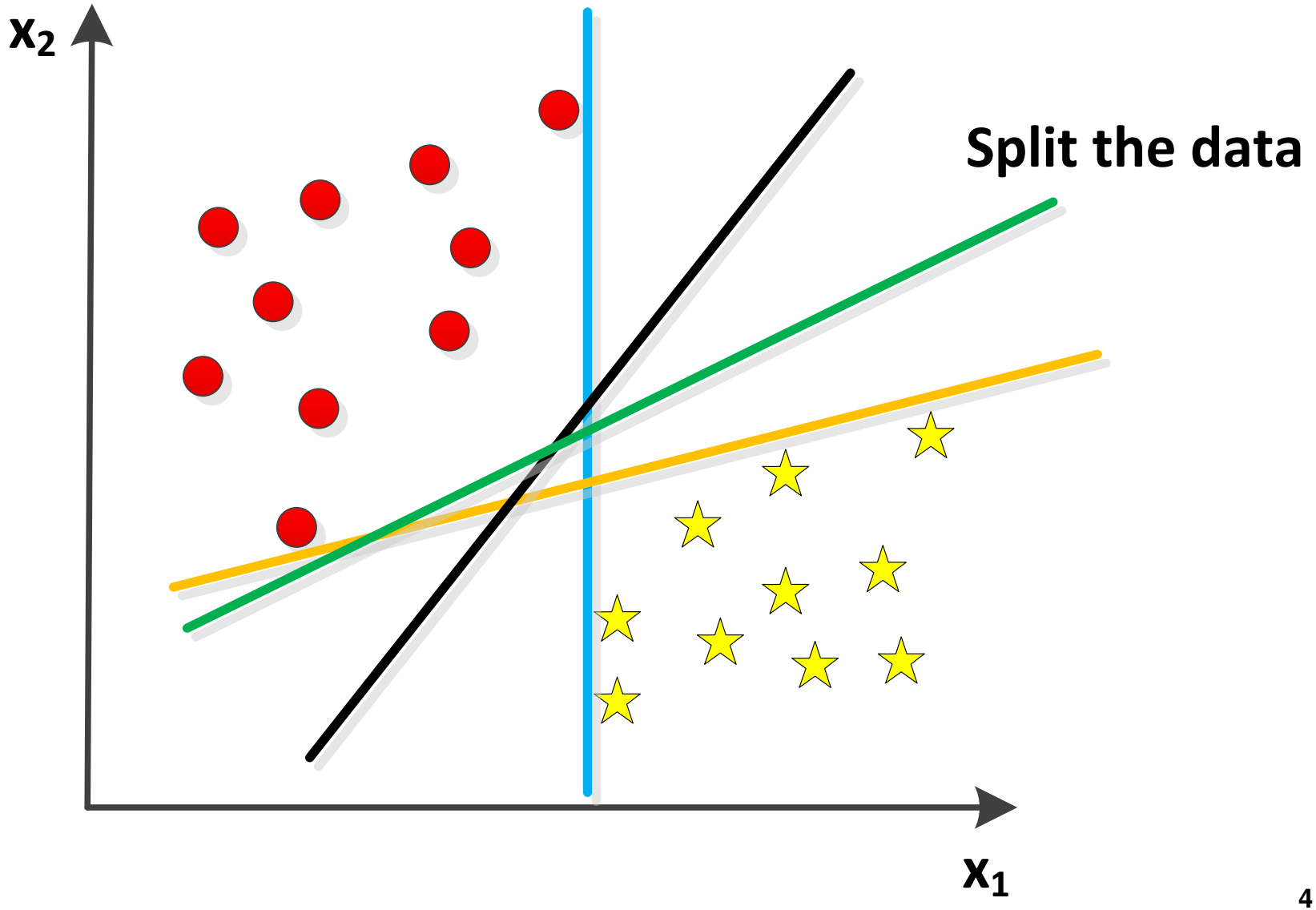
Linear Separator $w^\top x + b = 0$



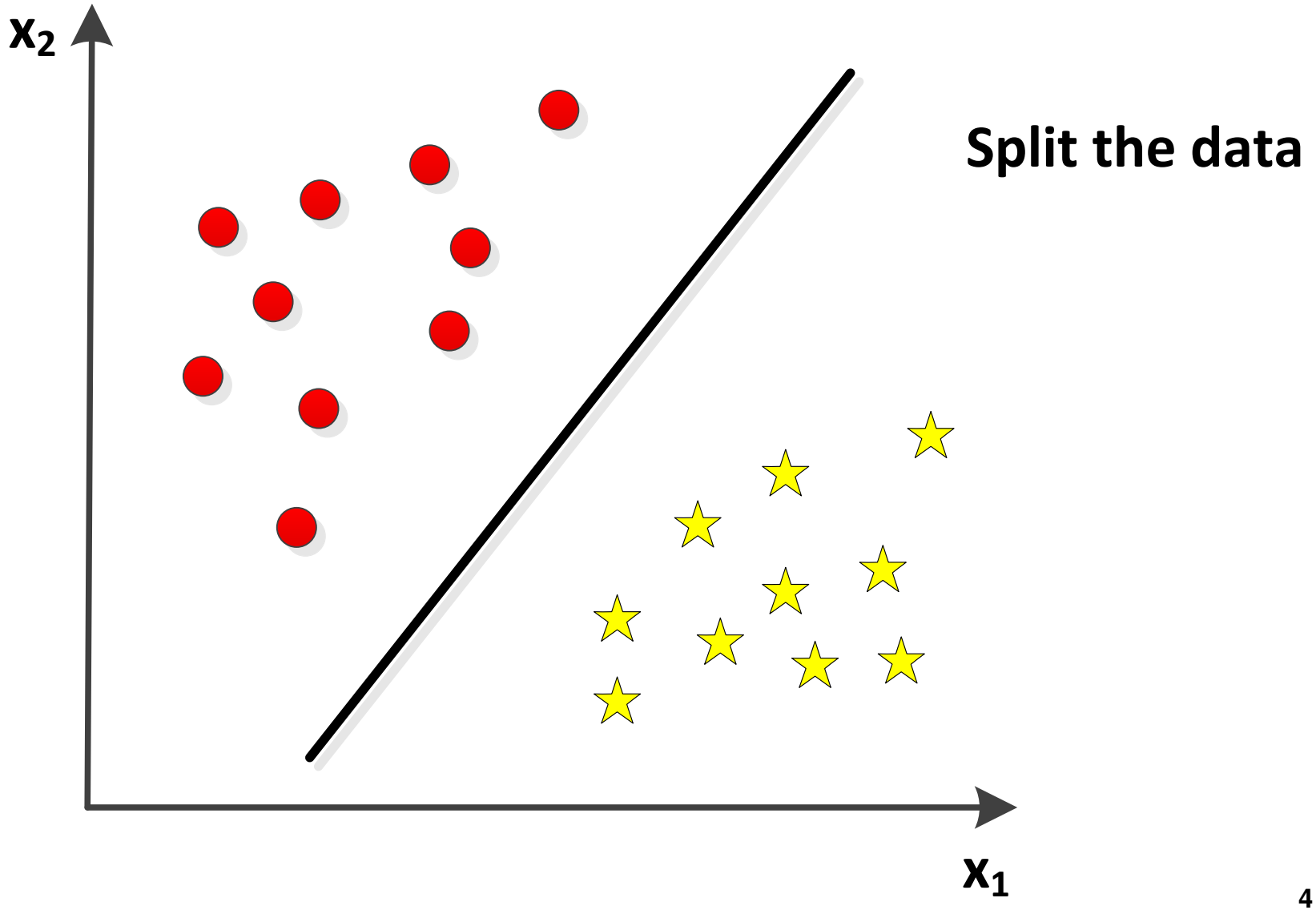
Linear Separator $w^\top x + b = 0$



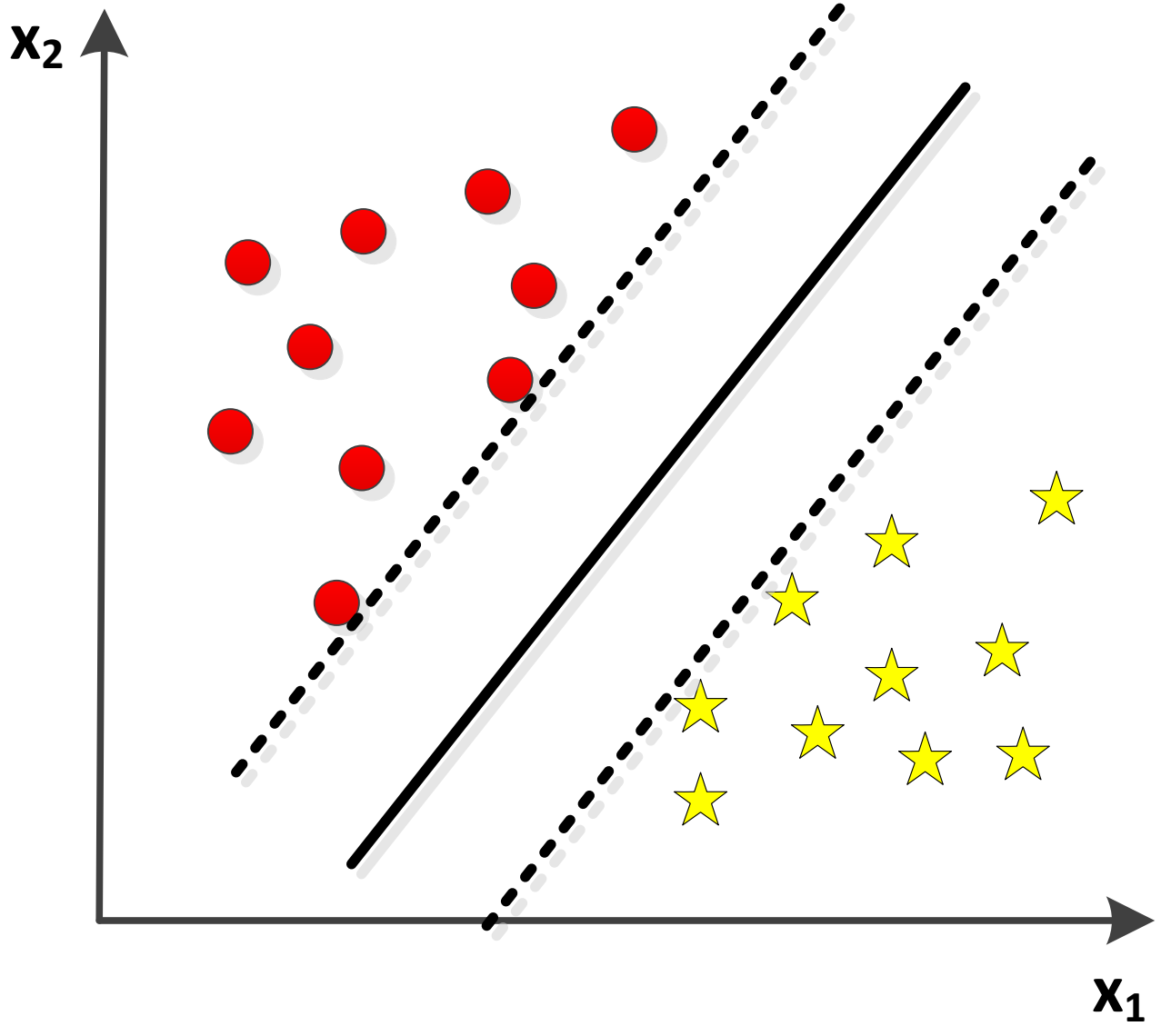
Linear Separator $w^\top x + b = 0$



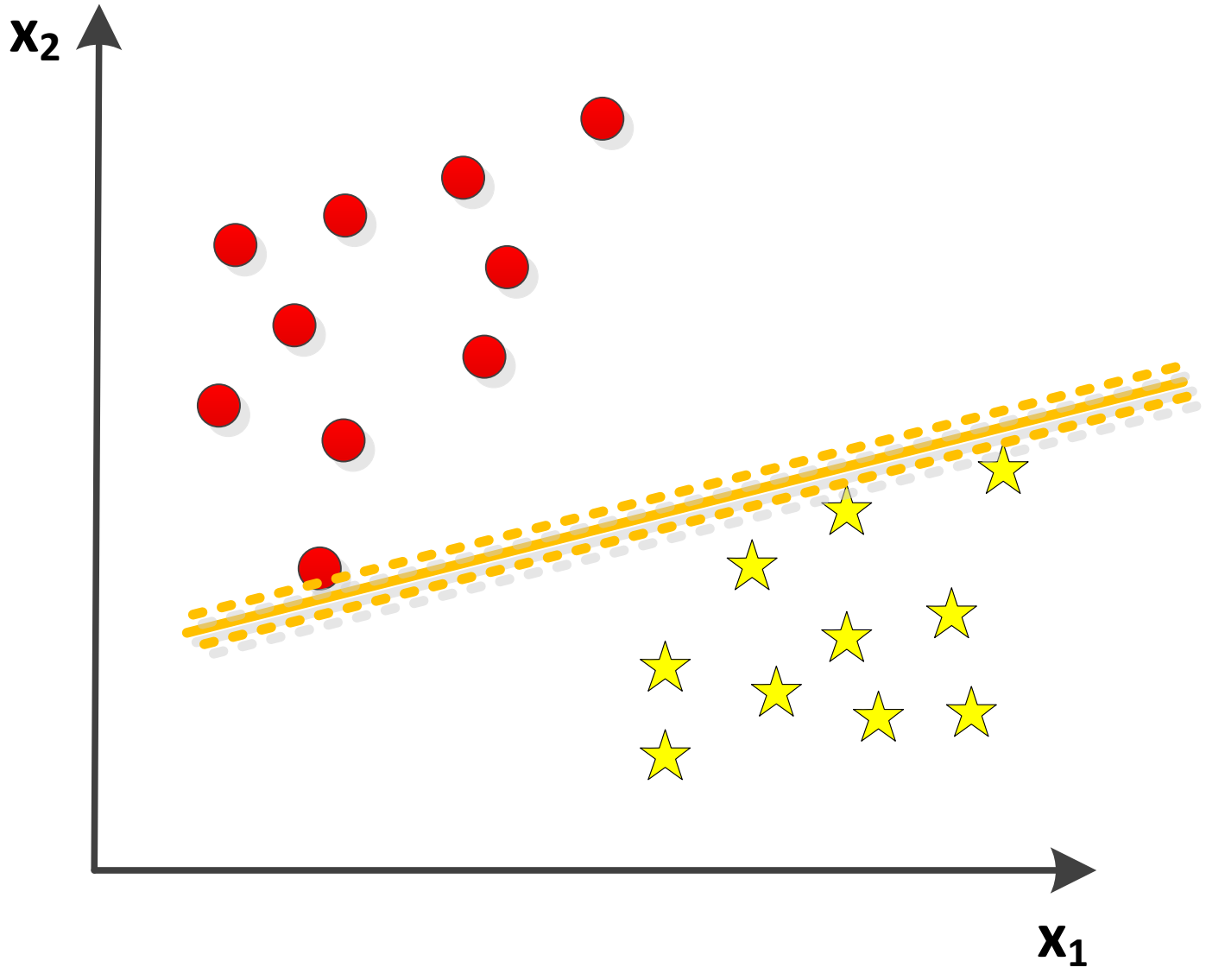
Linear Separator $w^\top x + b = 0$



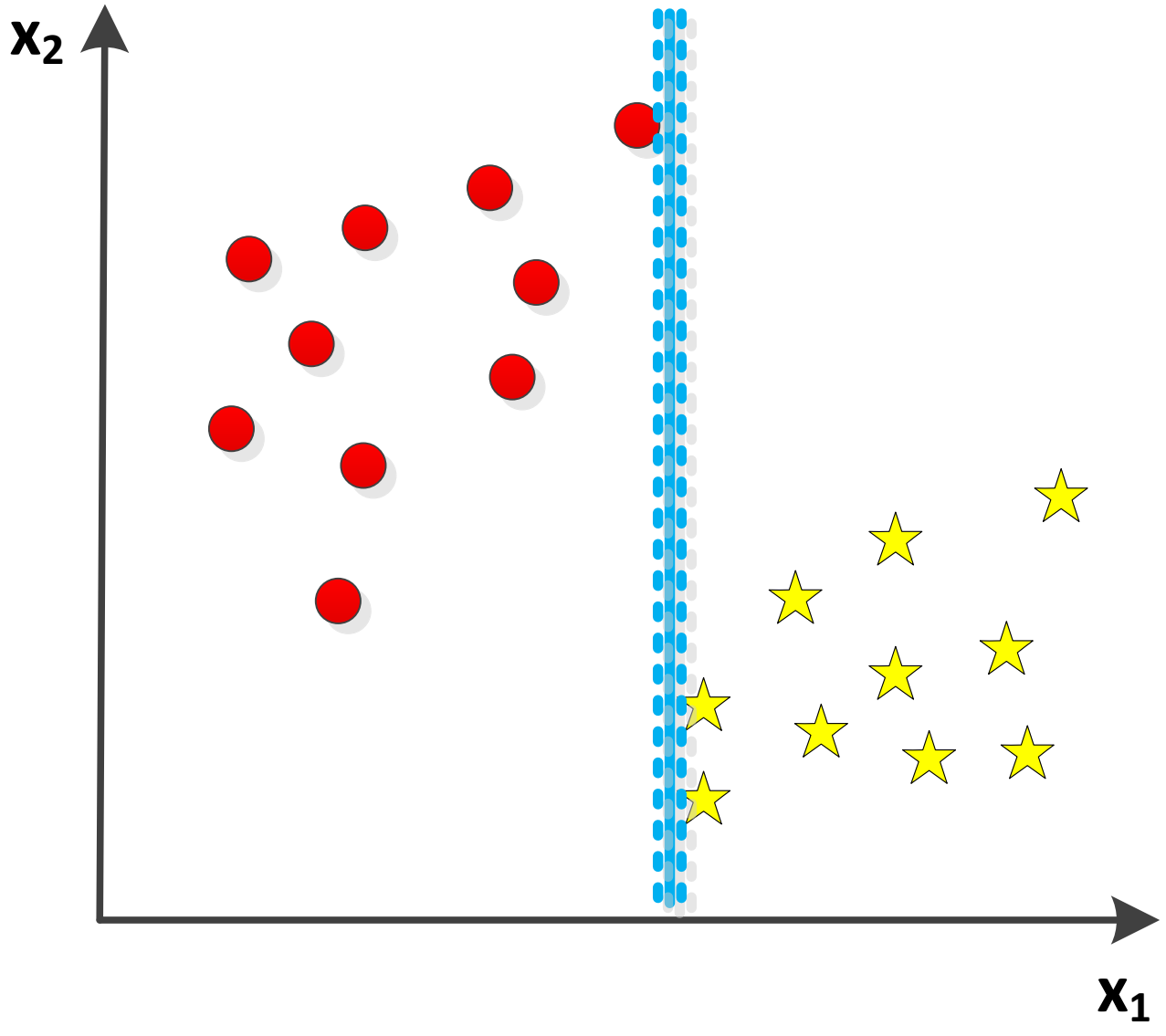
Linear Separator $w^\top x + b = 0$



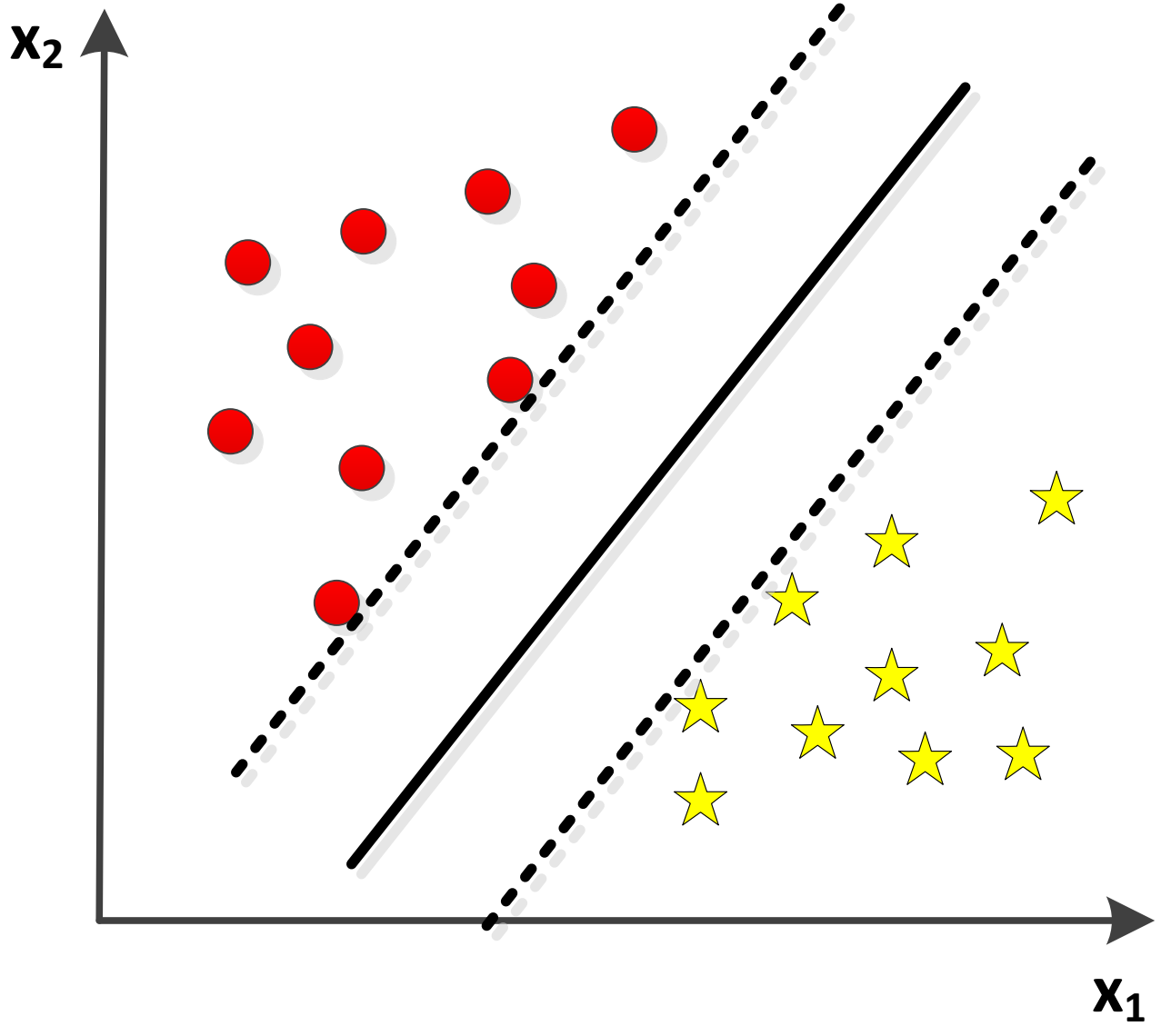
Linear Separator $w^\top x + b = 0$



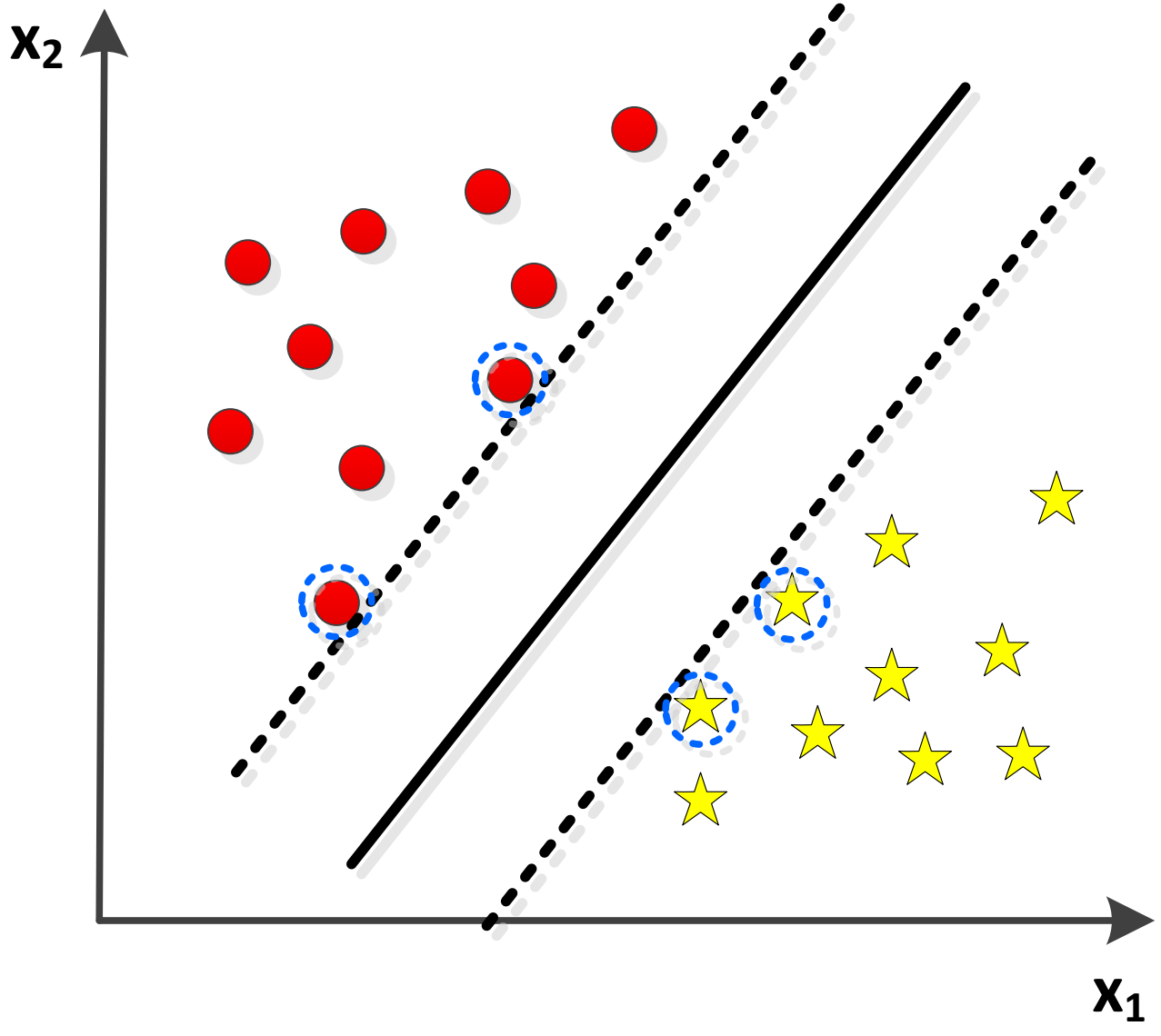
Linear Separator $w^\top x + b = 0$



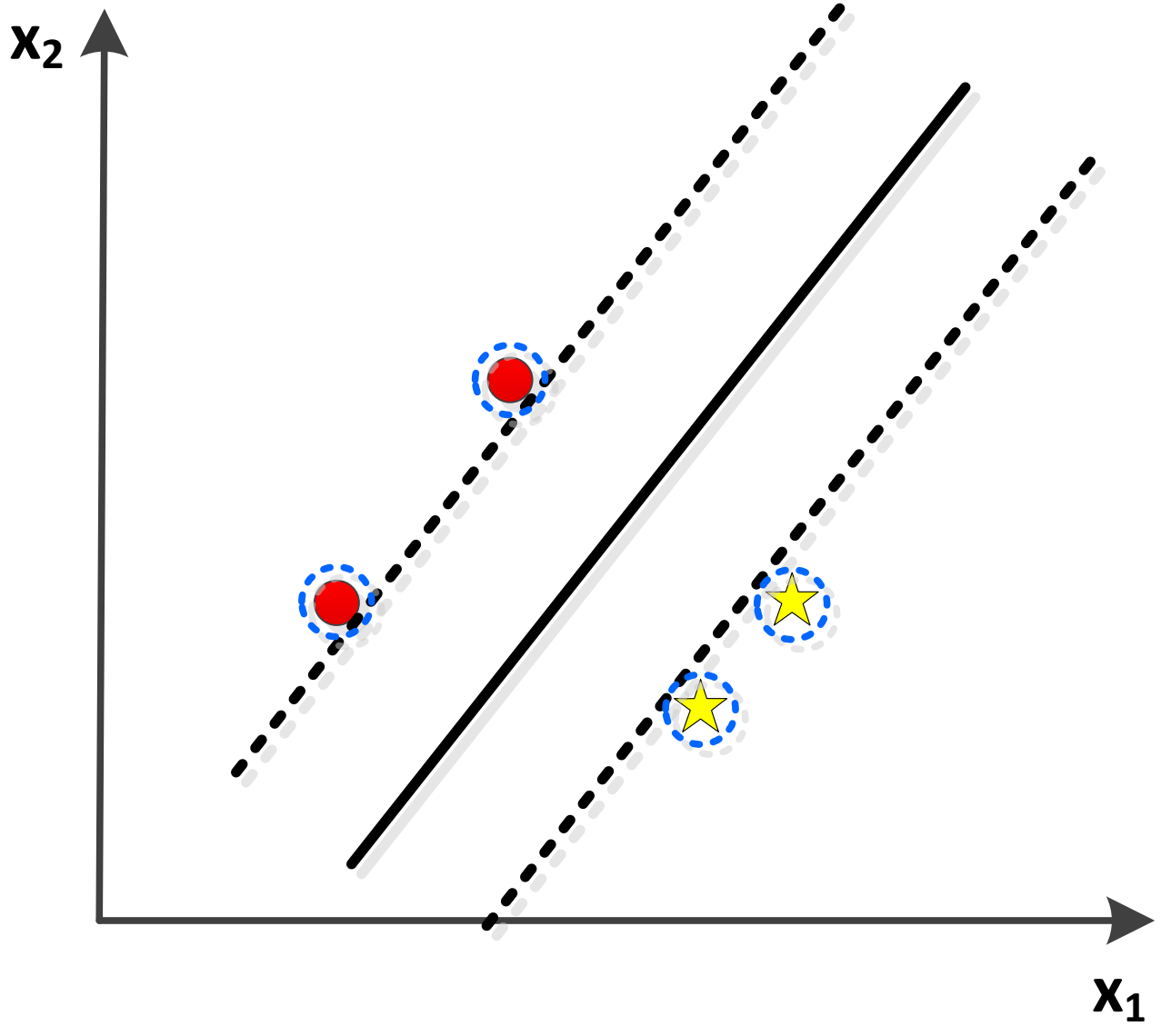
Linear Separator $w^\top x + b = 0$



Linear Separator $w^\top x + b = 0$



Linear Separator $w^\top x + b = 0$



Support vector machines (SVMs)

- A SVM classifier finds a (linear) separator (hyperplane) that **maximizes the margin** between the two classes of examples.
- The SVM is a type of **large margin classifier** (Other examples include boosting algorithms, voted-perceptron).
- Theoretical justifications: large margin leads to good generalization (i.e., high accuracy on test data).

Problem Formulation: Margin

Illustration of the geometry of hyperplane

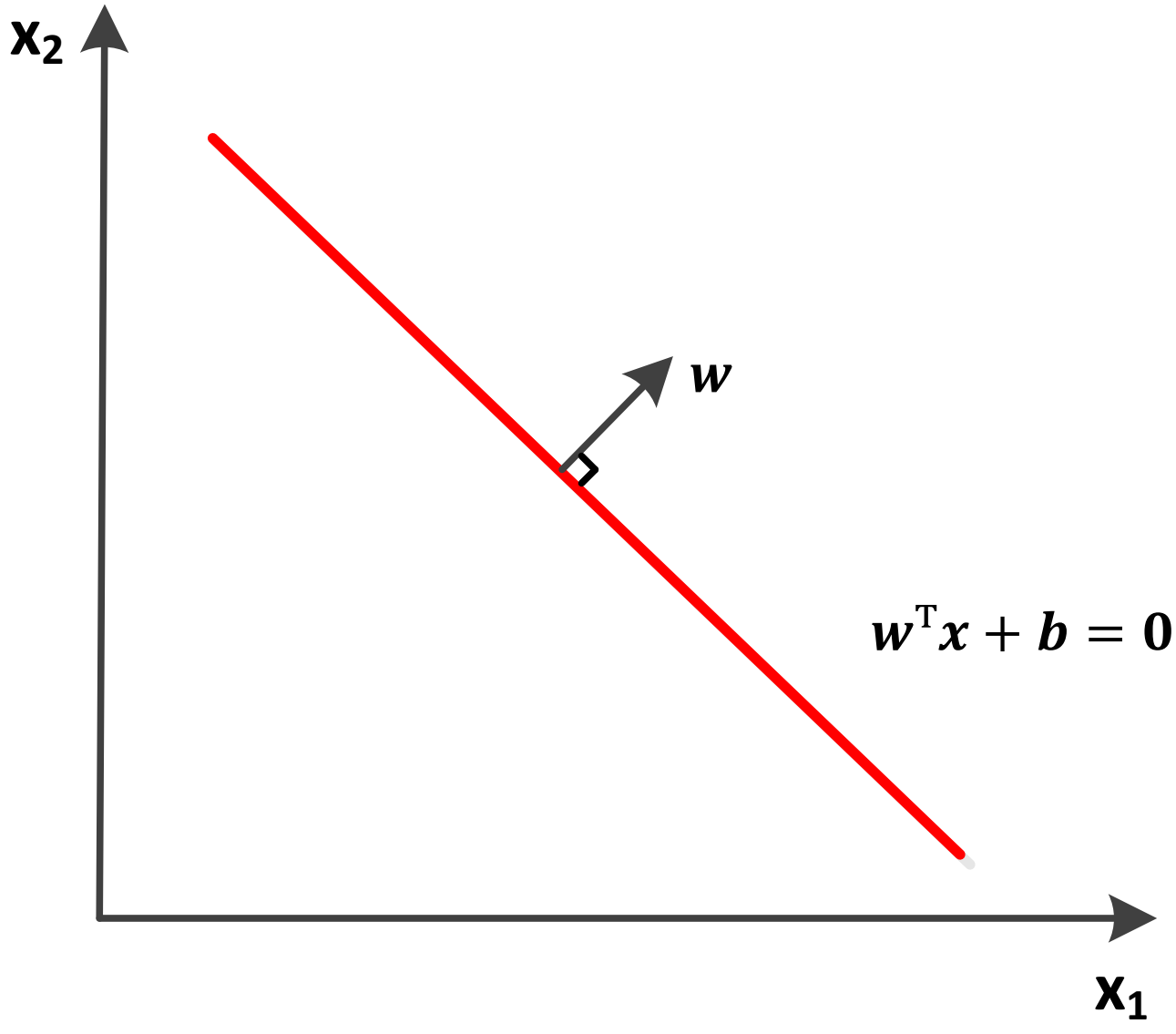


Illustration of the geometry of hyperplane

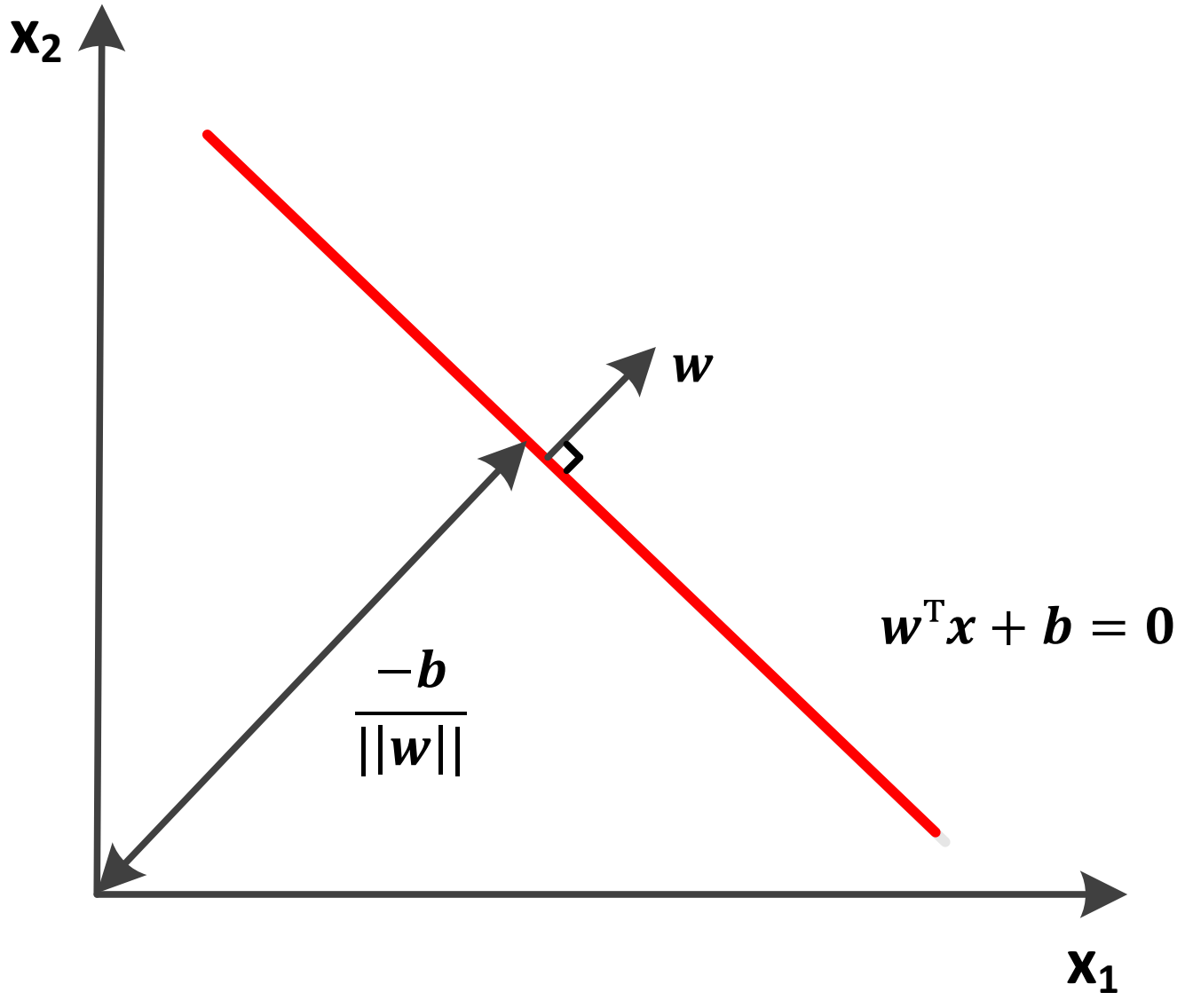


Illustration of the geometry of hyperplane

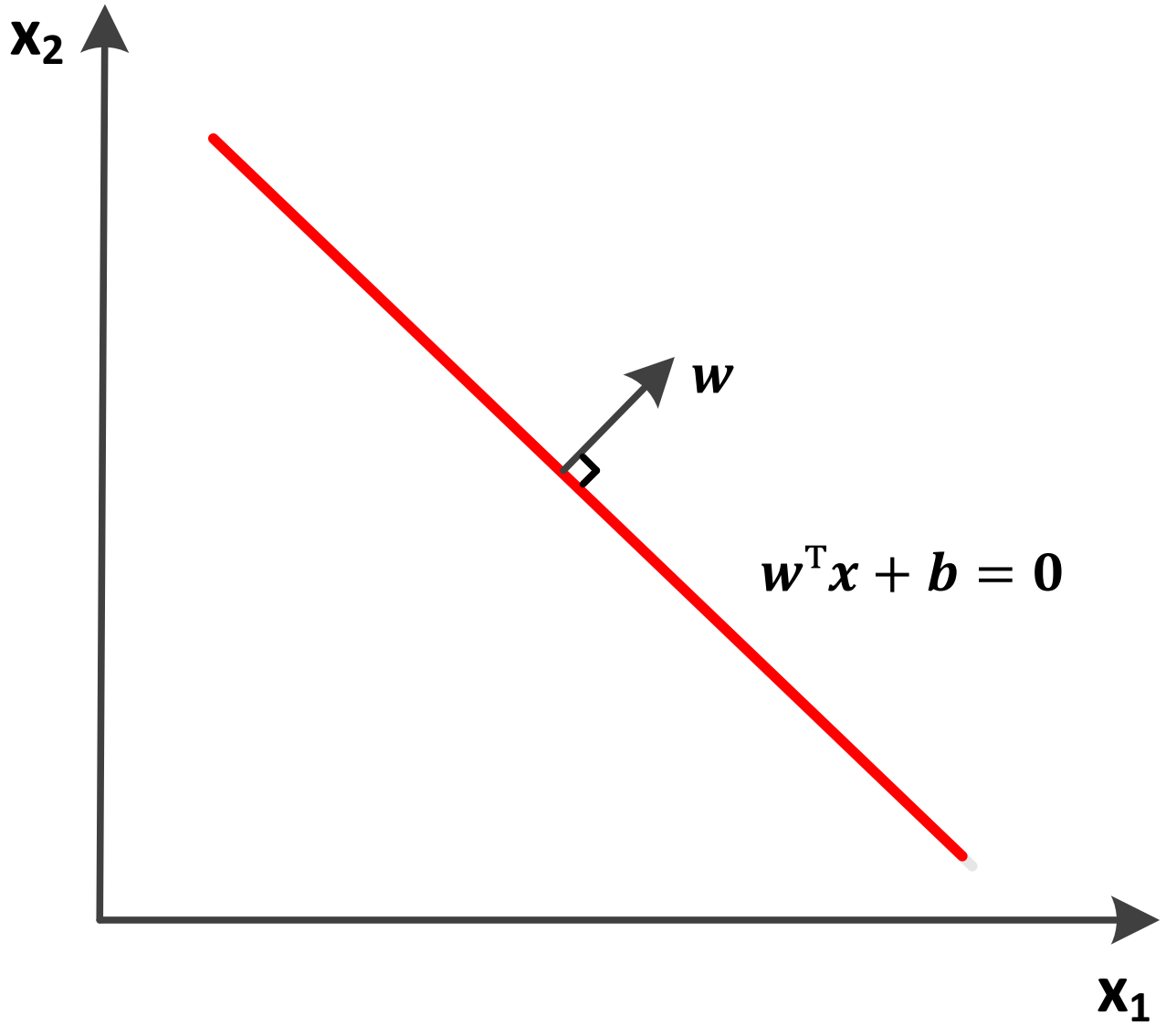


Illustration of the geometry of hyperplane

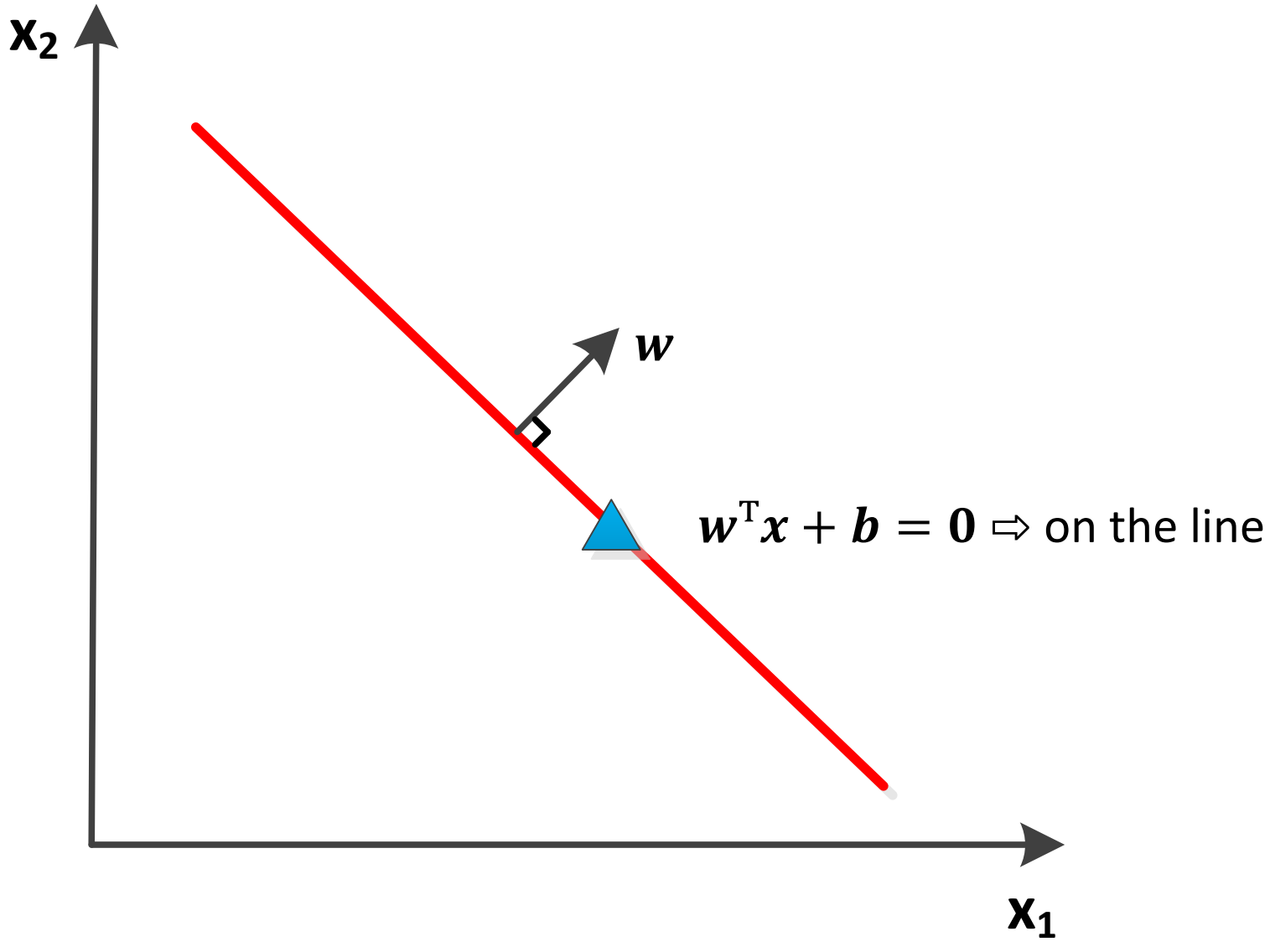


Illustration of the geometry of hyperplane

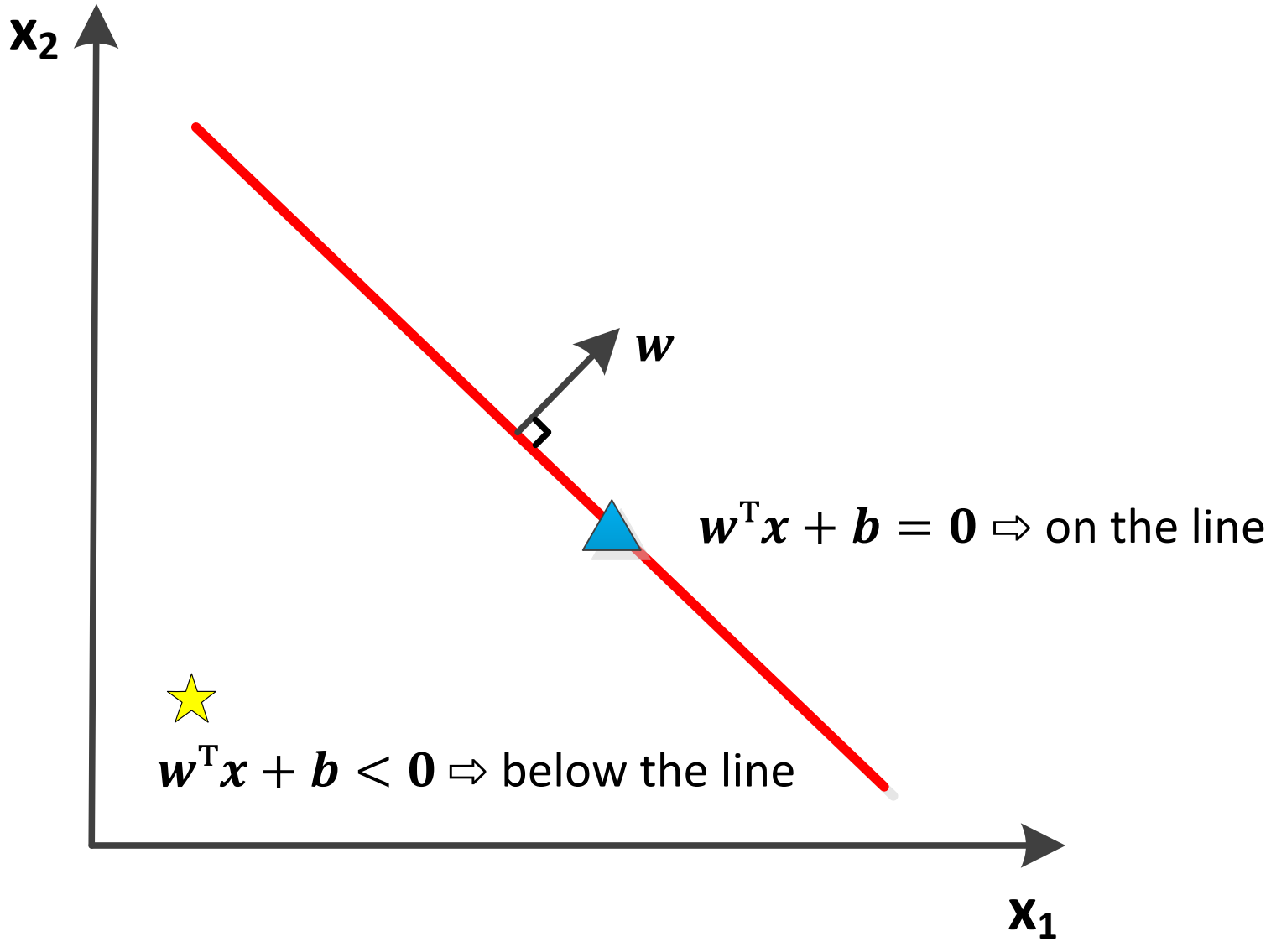


Illustration of the geometry of hyperplane

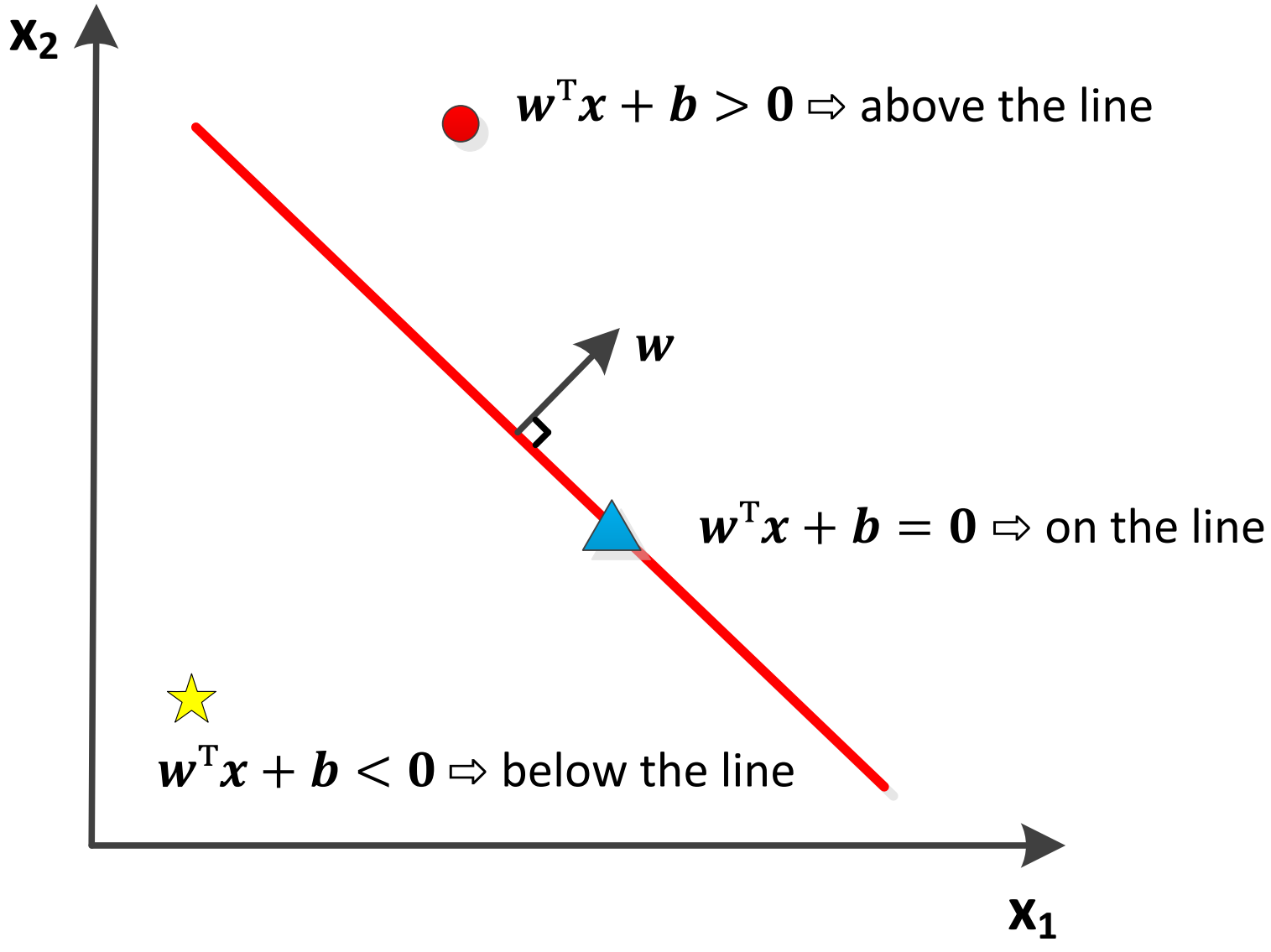


Illustration of the geometry of hyperplane

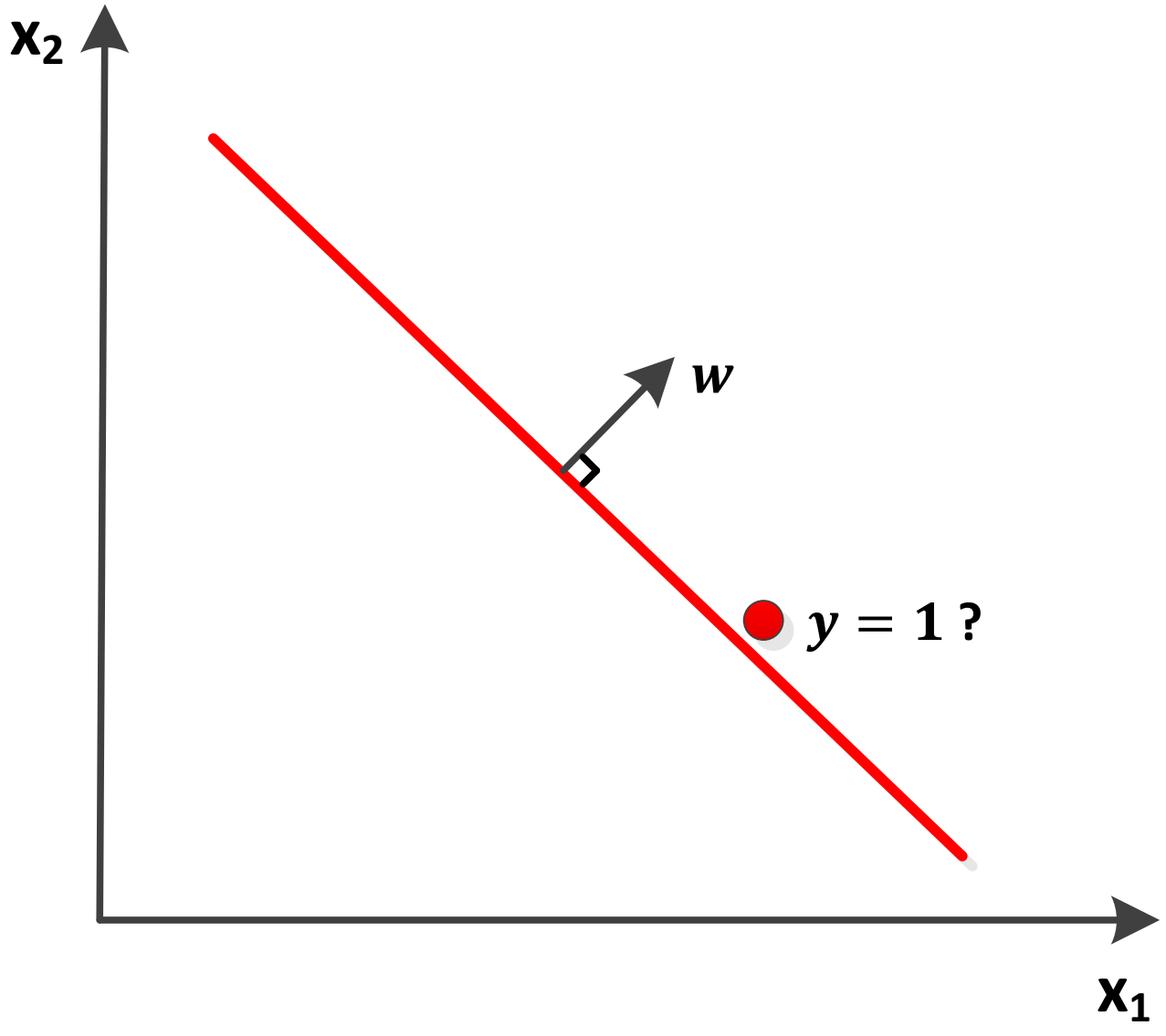


Illustration of the geometry of hyperplane

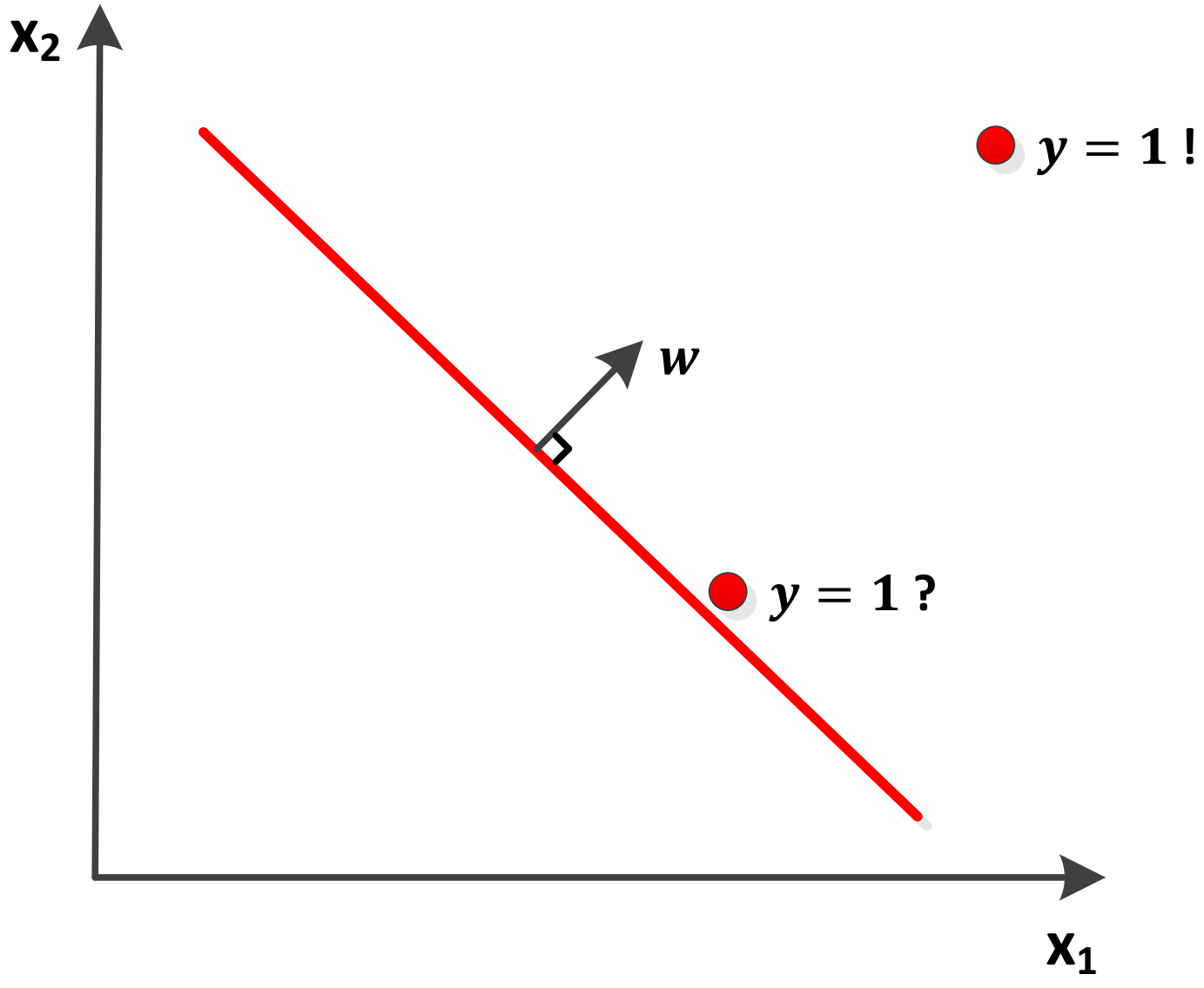


Illustration of the geometry of margin

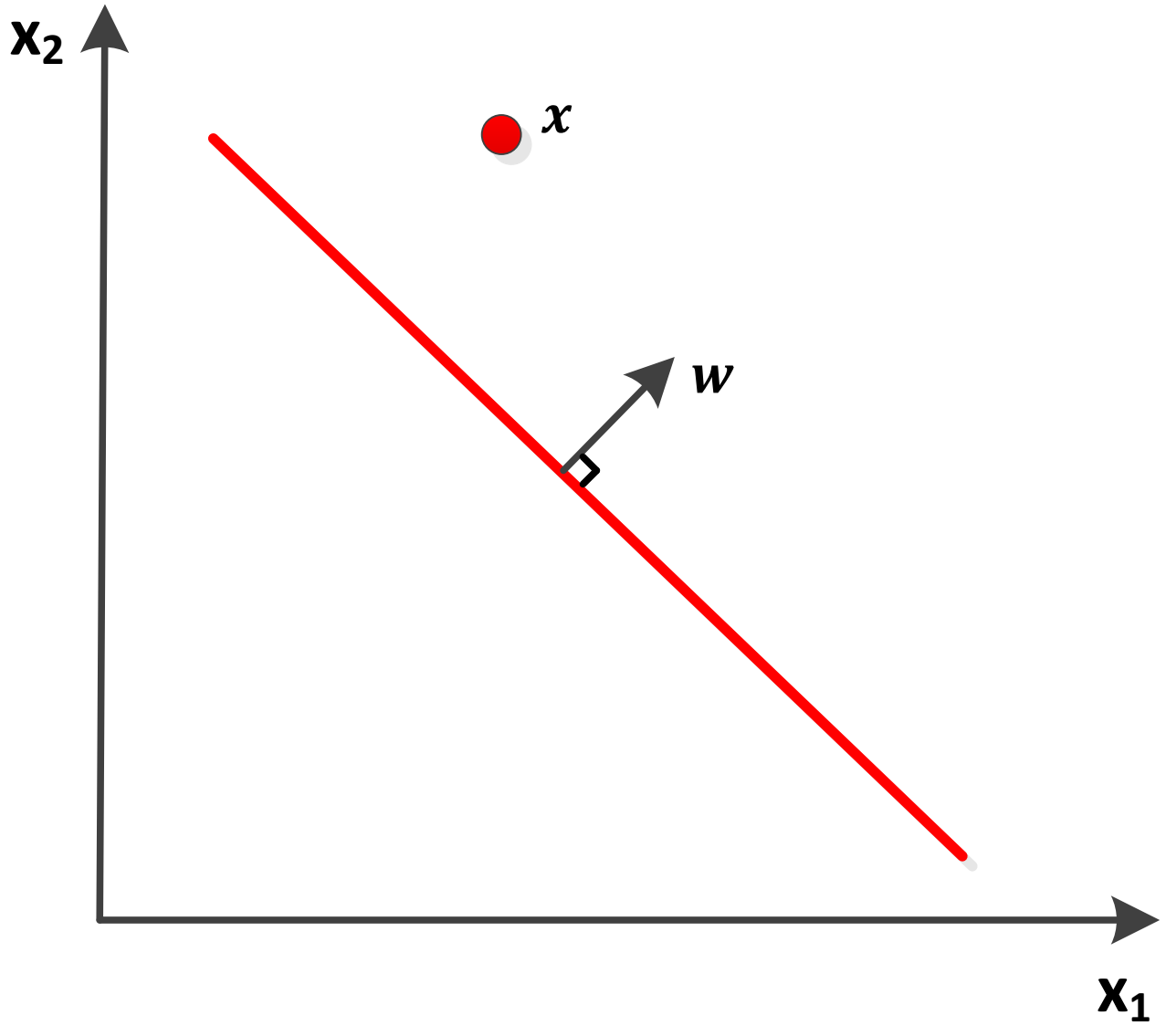


Illustration of the geometry of margin

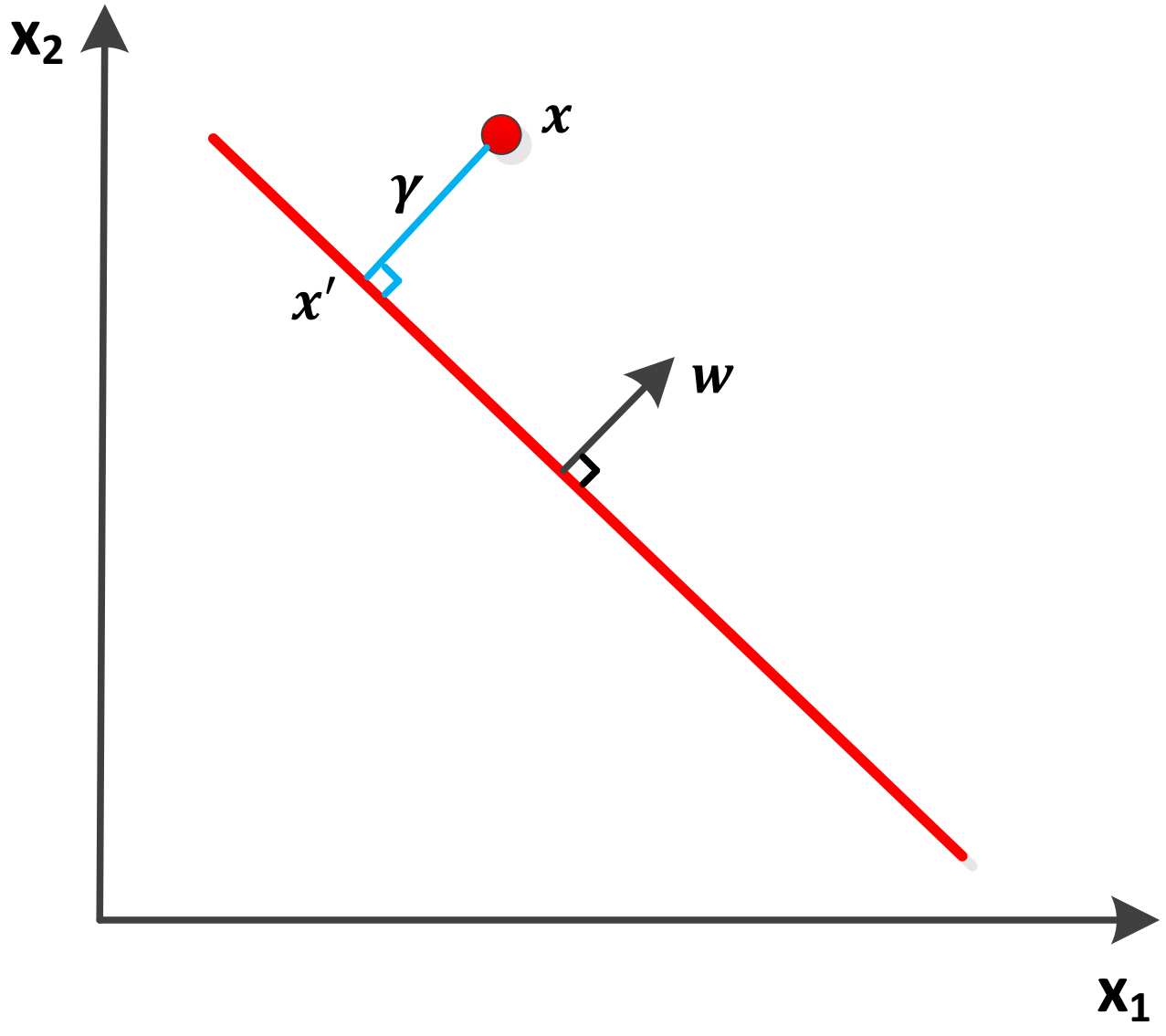


Illustration of the geometry of margin

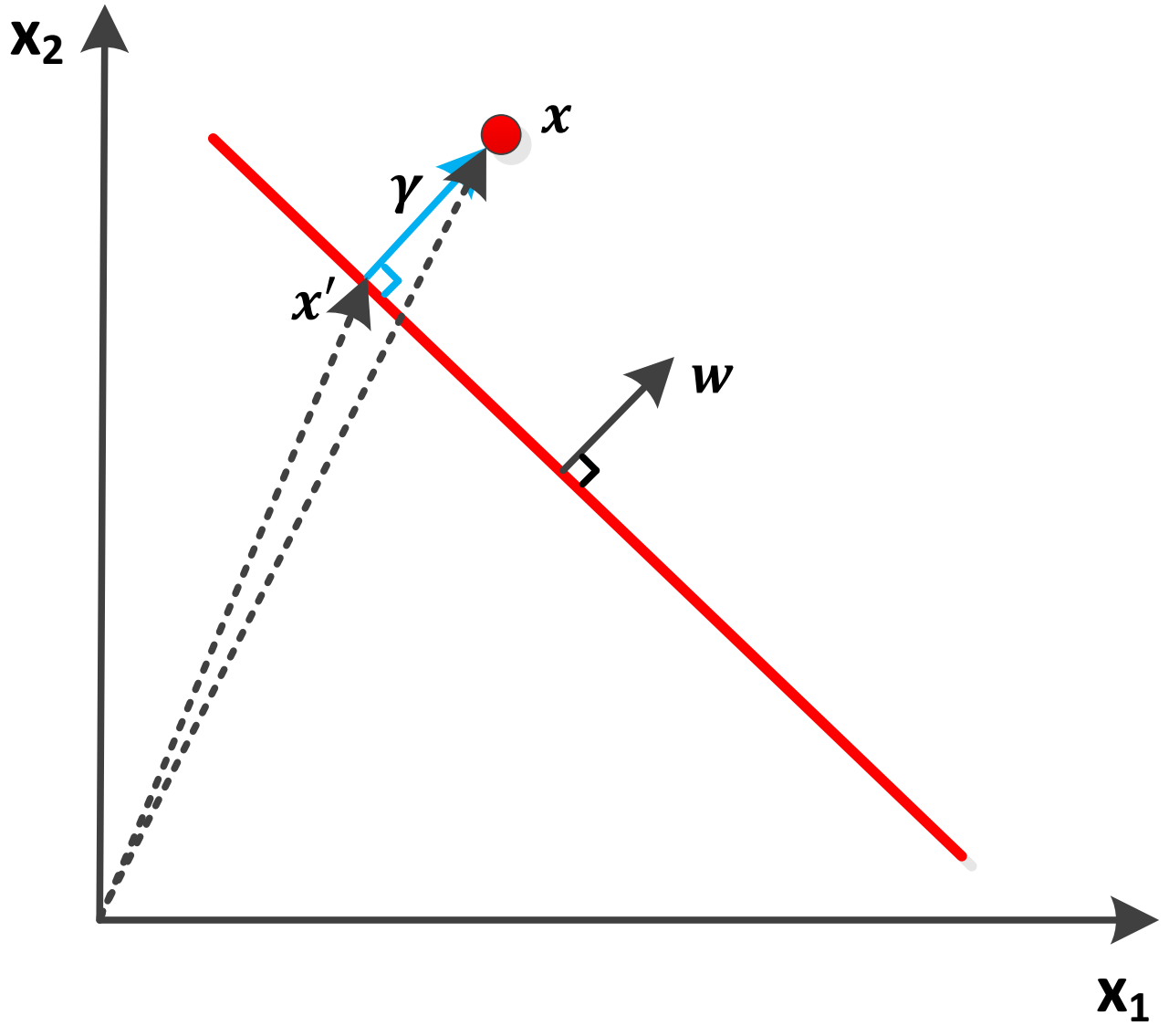


Illustration of the geometry of margin

$$x' = x - \gamma \frac{w}{||w||}$$

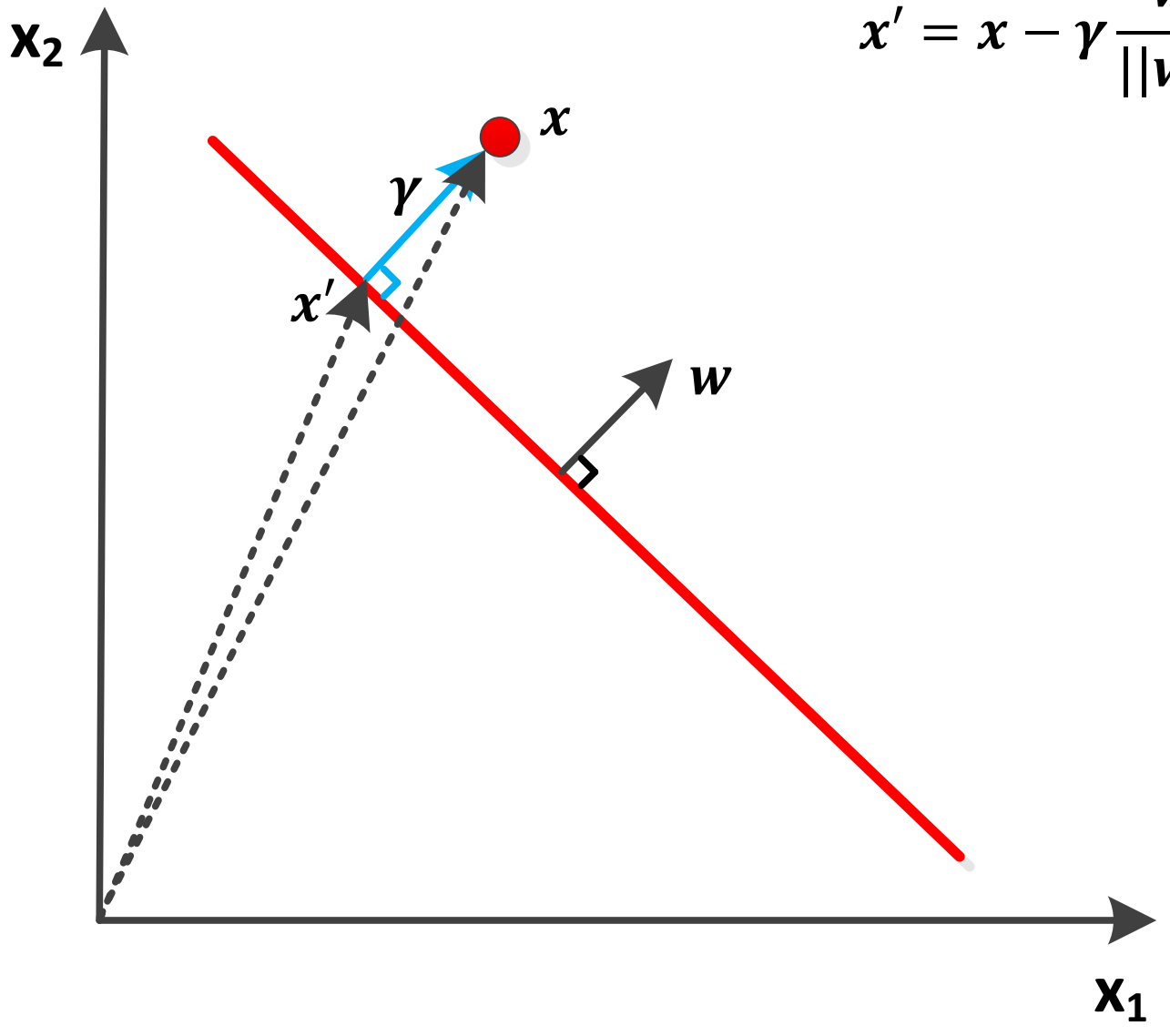
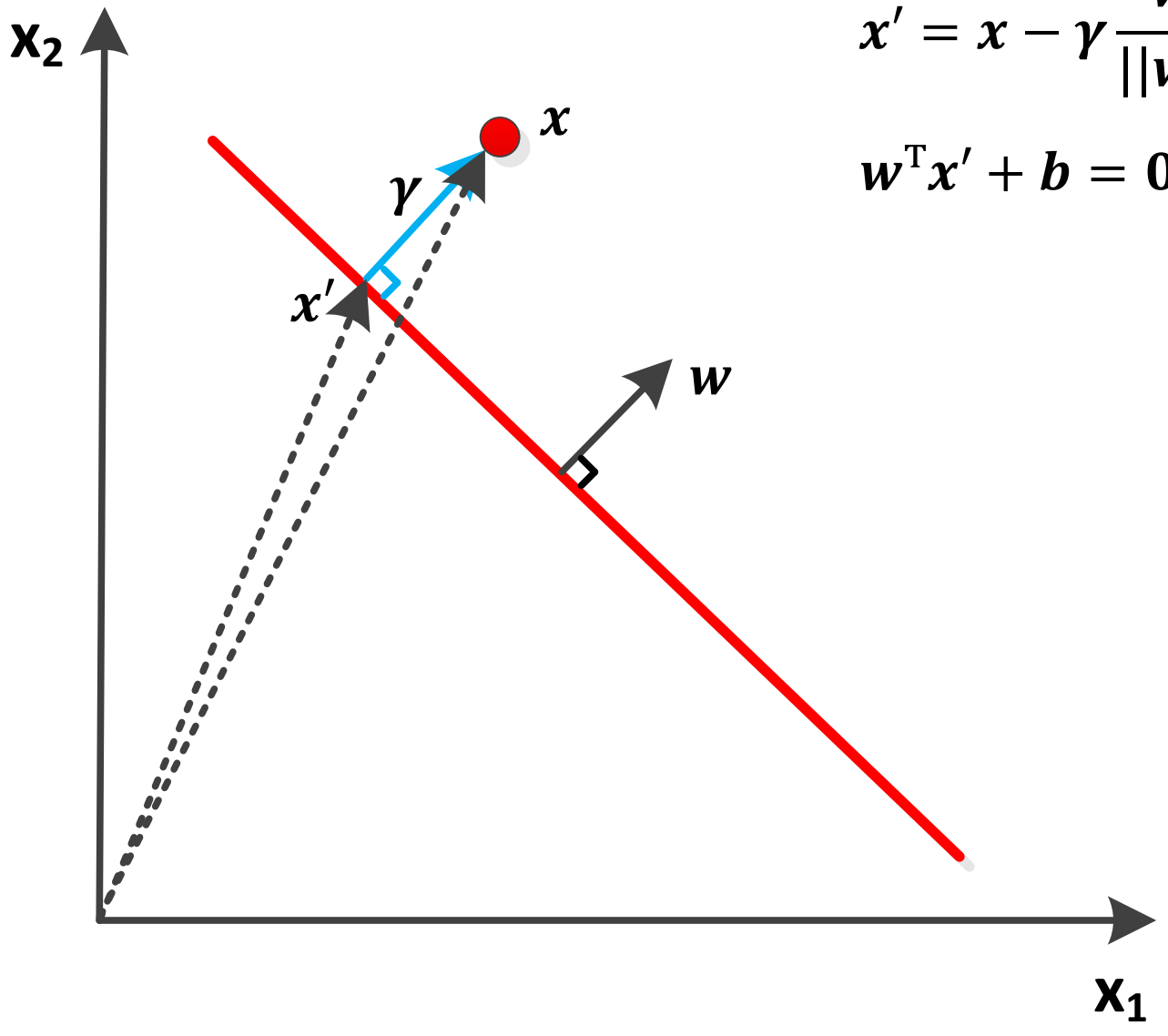


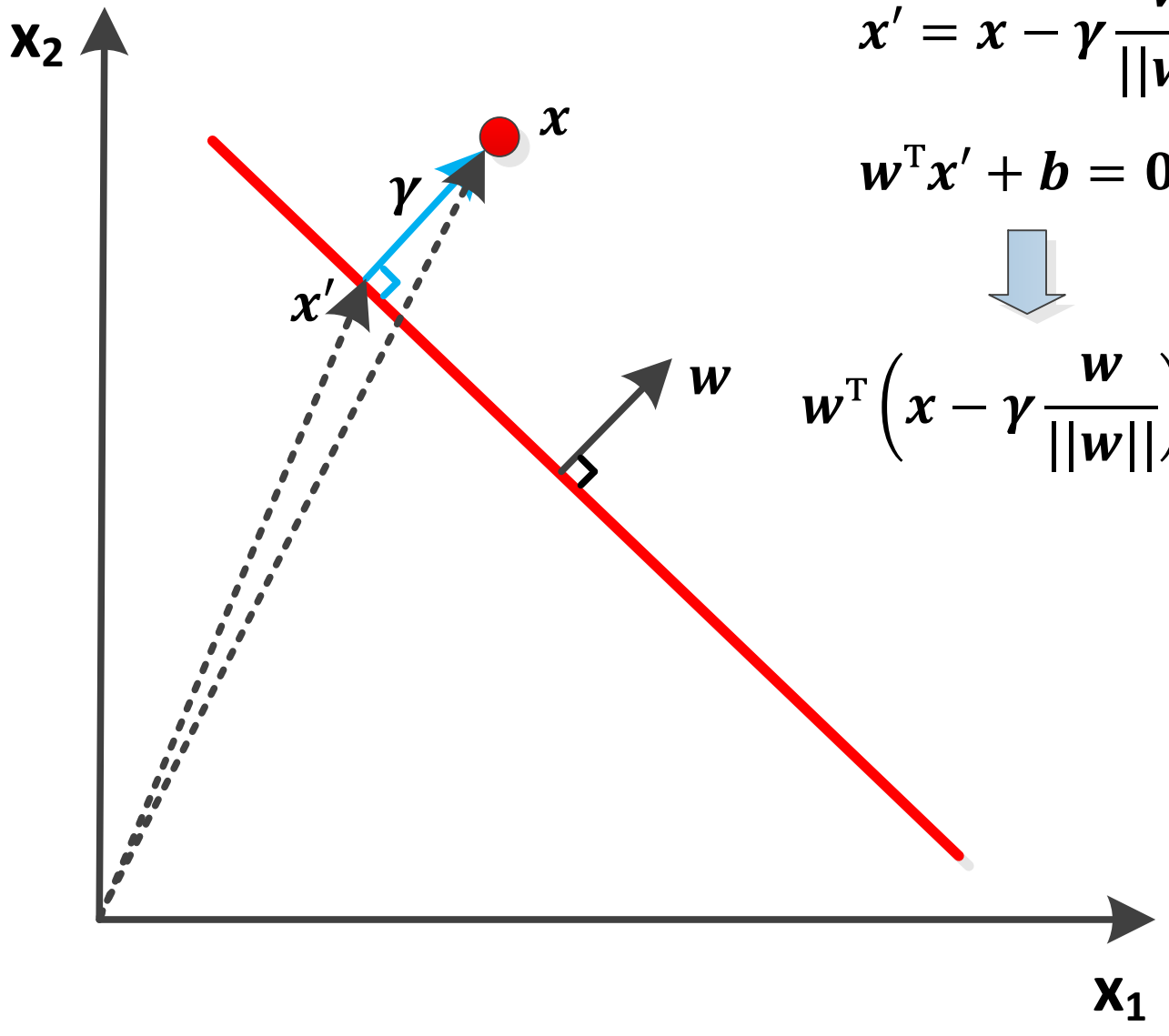
Illustration of the geometry of margin



$$x' = x - \gamma \frac{w}{||w||}$$

$$w^{\mathrm{T}} x' + b = 0$$

Illustration of the geometry of margin

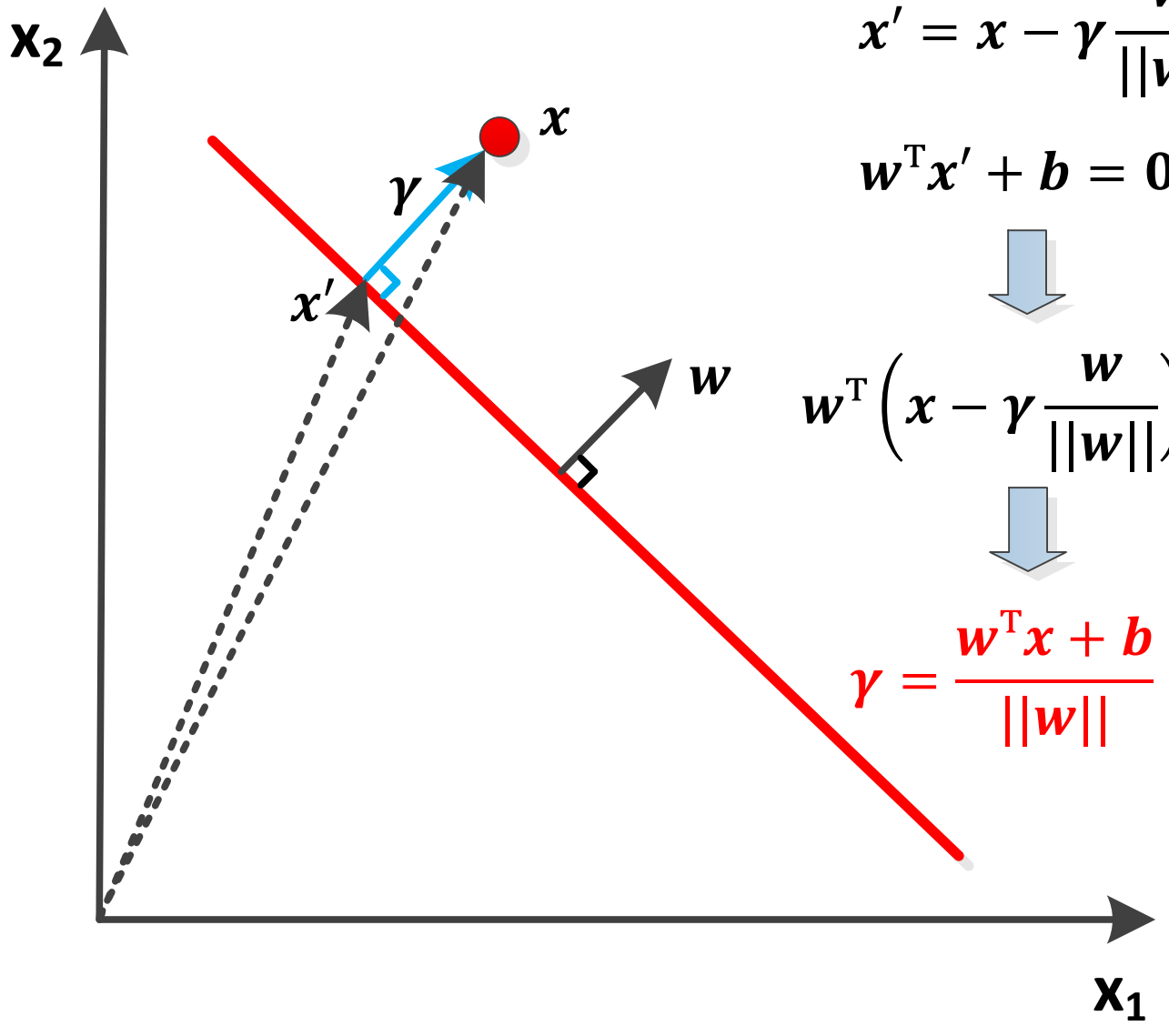


$$x' = x - \gamma \frac{w}{||w||}$$

$$w^{\mathrm{T}} x' + b = 0$$

$$w^{\mathrm{T}} \left(x - \gamma \frac{w}{||w||} \right) + b = 0$$

Illustration of the geometry of margin



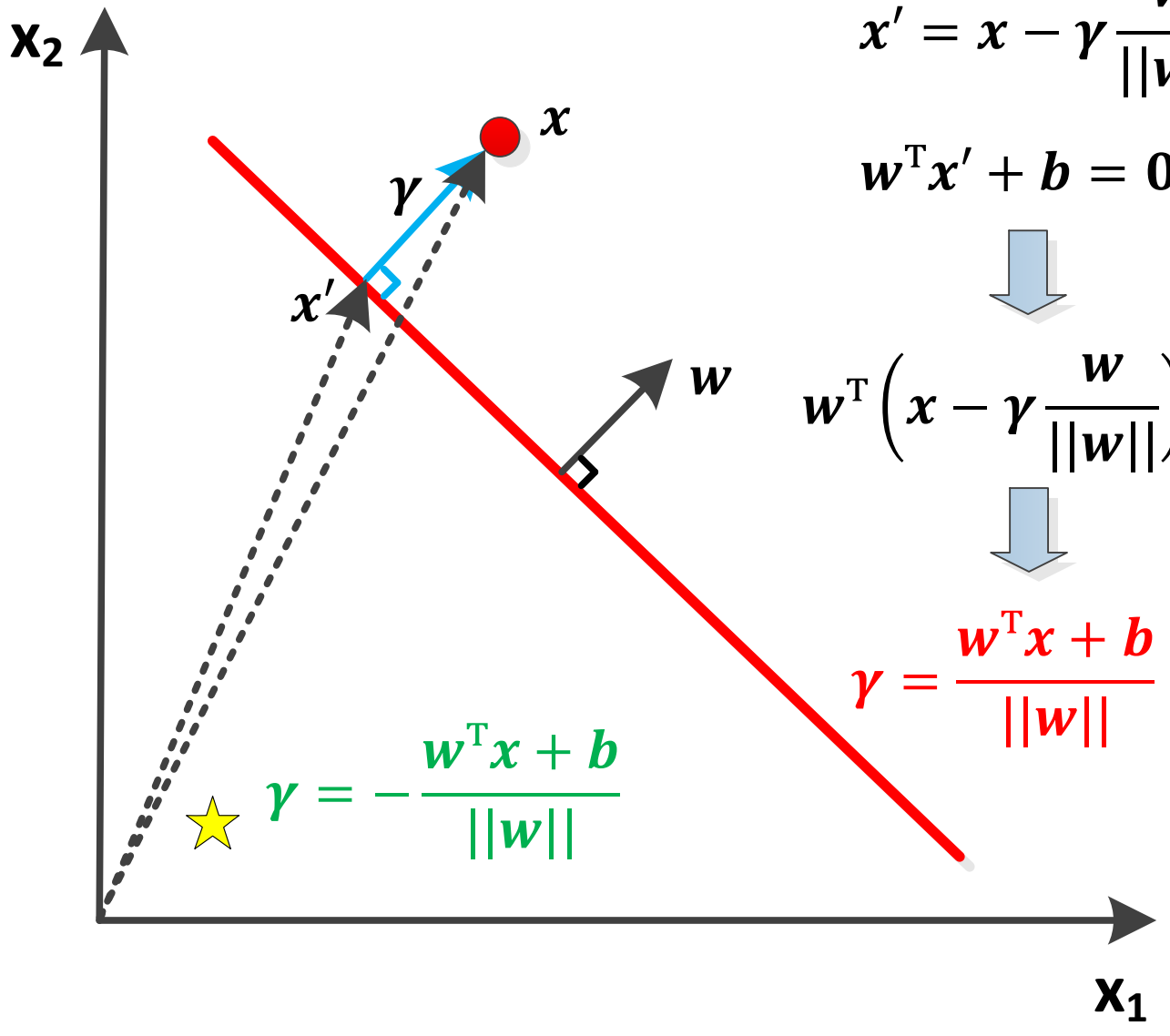
$$x' = x - \gamma \frac{w}{||w||}$$

$$w^{\mathrm{T}} x' + b = 0$$

$$w^{\mathrm{T}} \left(x - \gamma \frac{w}{||w||} \right) + b = 0$$

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

Illustration of the geometry of margin



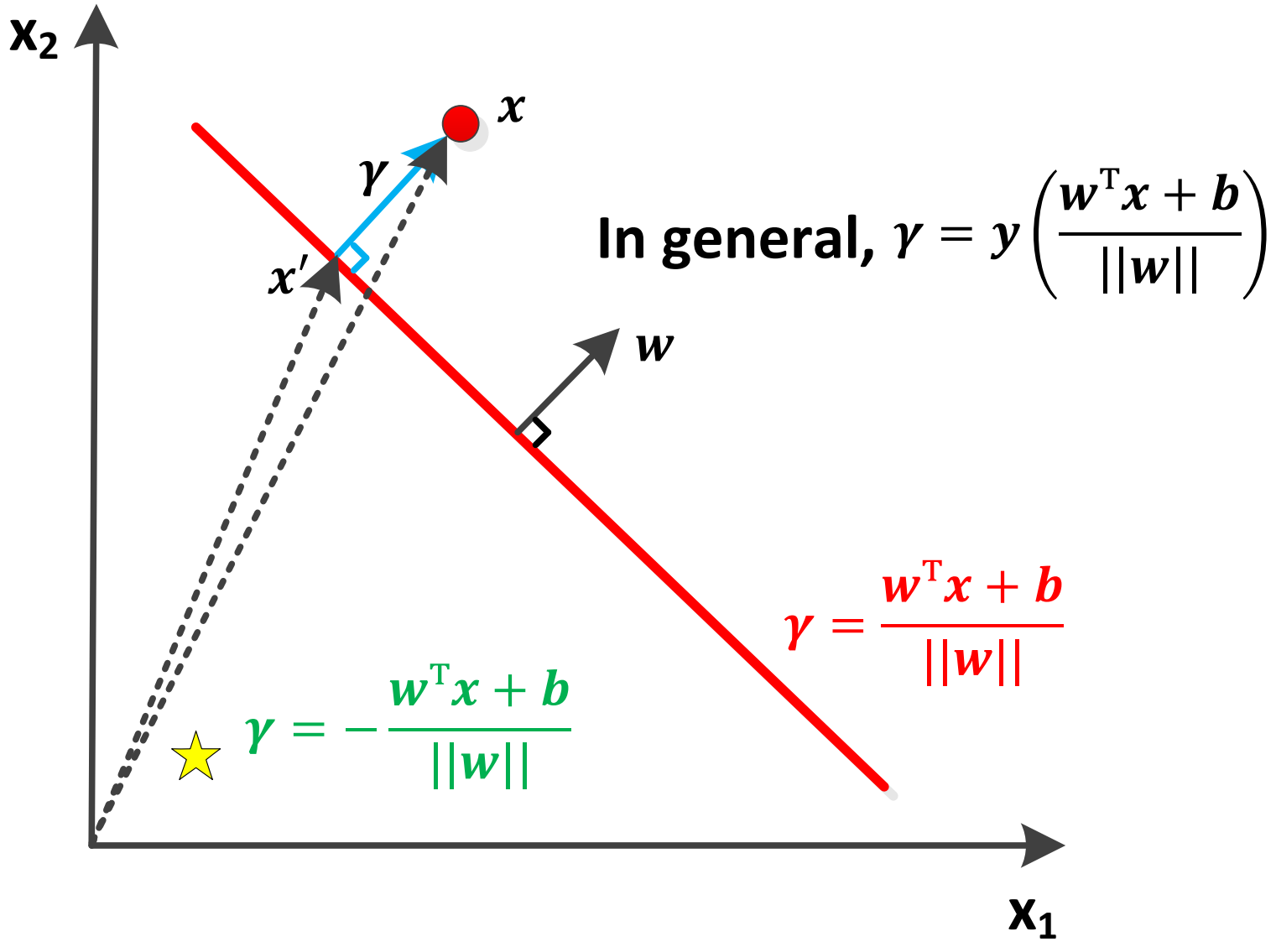
$$x' = x - \gamma \frac{w}{||w||}$$

$$w^T x' + b = 0$$

$$w^T \left(x - \gamma \frac{w}{||w||} \right) + b = 0$$

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

Illustration of the geometry of margin

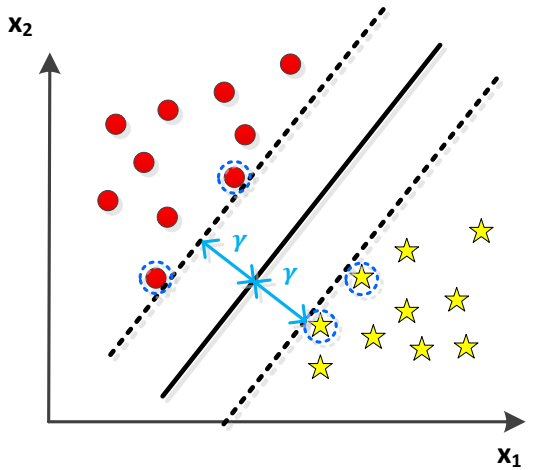


Large margin intuition

- **Margin** is the distance between an example and the decision hyperplane, denoted by γ.
- Maximize the margin of the data points which are closest to the hyperplane.

- Given a training set $\mathcal{S} = \{(x_i, y_i); i = 1, \ldots, m\}$, learn a hyperplane to maximize the margin γ, where

$$\gamma = \min_{i=1,\ldots,m} \gamma_i = \min_{i=1,\ldots,m} \frac{y_i(w^\top x_i + b)}{||w||}$$



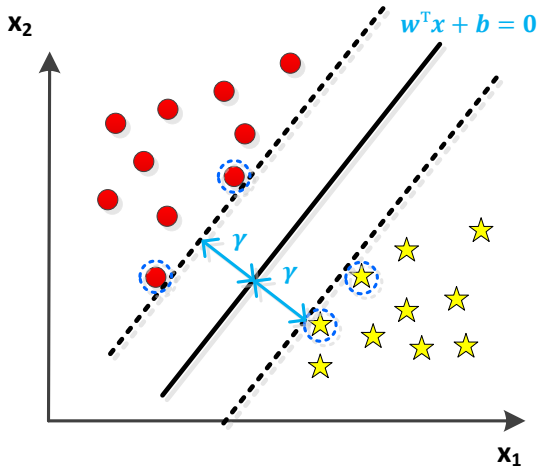
Large margin formulation

- Objective function:

$$\max_{w,b} \ \gamma$$

$$\text{s.t.} \ \frac{y_i(w^\top x_i + b)}{||w||} \geq \gamma, \ \forall i = 1, \ldots, m$$

- The decision hyperplane and γ are invariant to the rescaling of the parameters – need additional constraints.



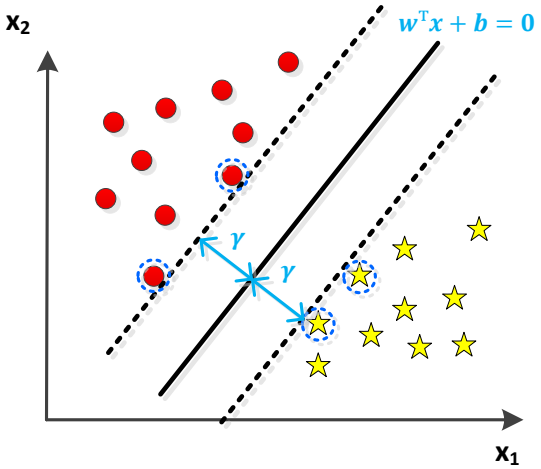
Large margin formulation

- Option 1:

$$\max_{w,b} \ \gamma$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq \gamma, \ \forall i = 1, \ldots, m$$

$$||w|| = 1$$

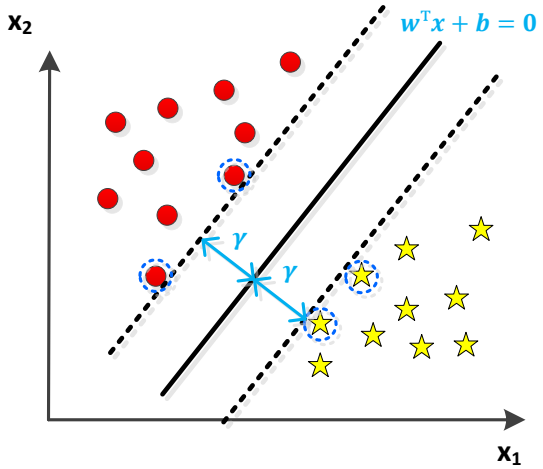


Large margin formulation

- Option 1:

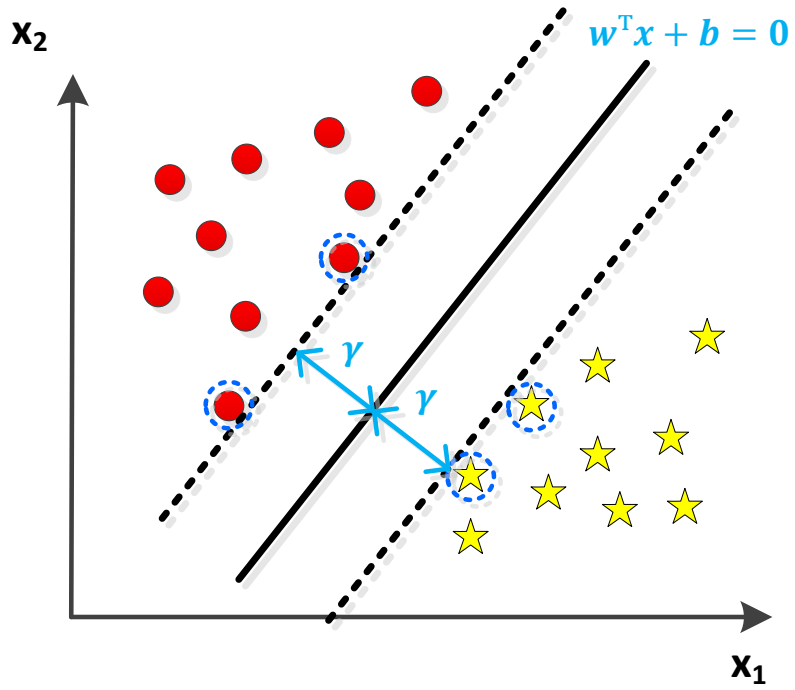
$$\max_{w,b} \ \gamma$$
$$\text{s.t. } y_i(w^\top x_i + b) \geq \gamma, \ \forall i = 1, \ldots, m$$
$$||w|| = 1$$

Non-convex optimization problem, hard to solve!



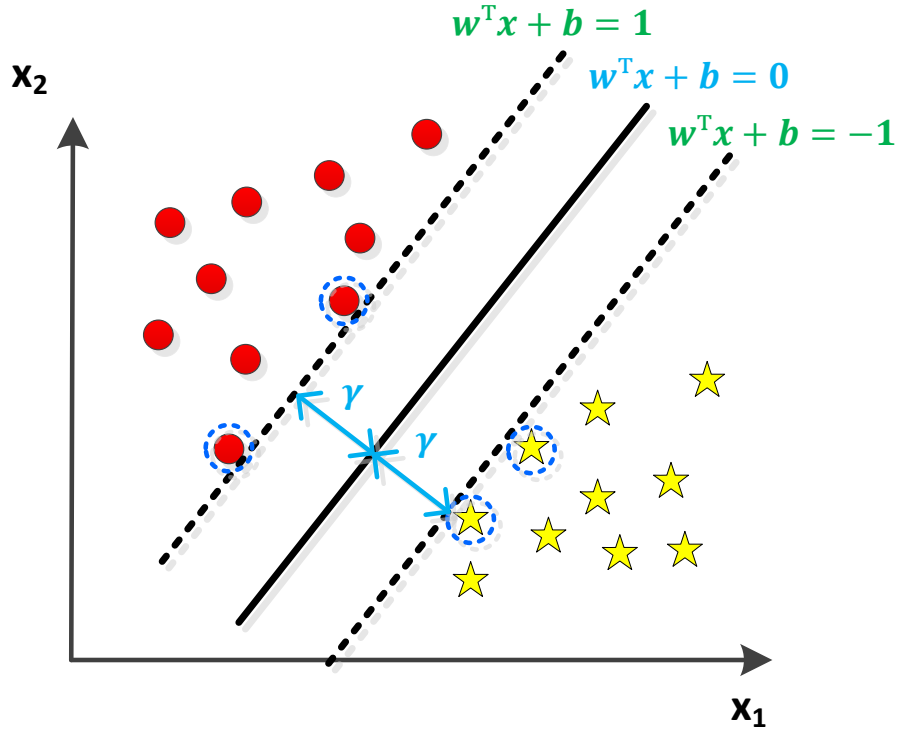
Large margin formulation

- Option 2:



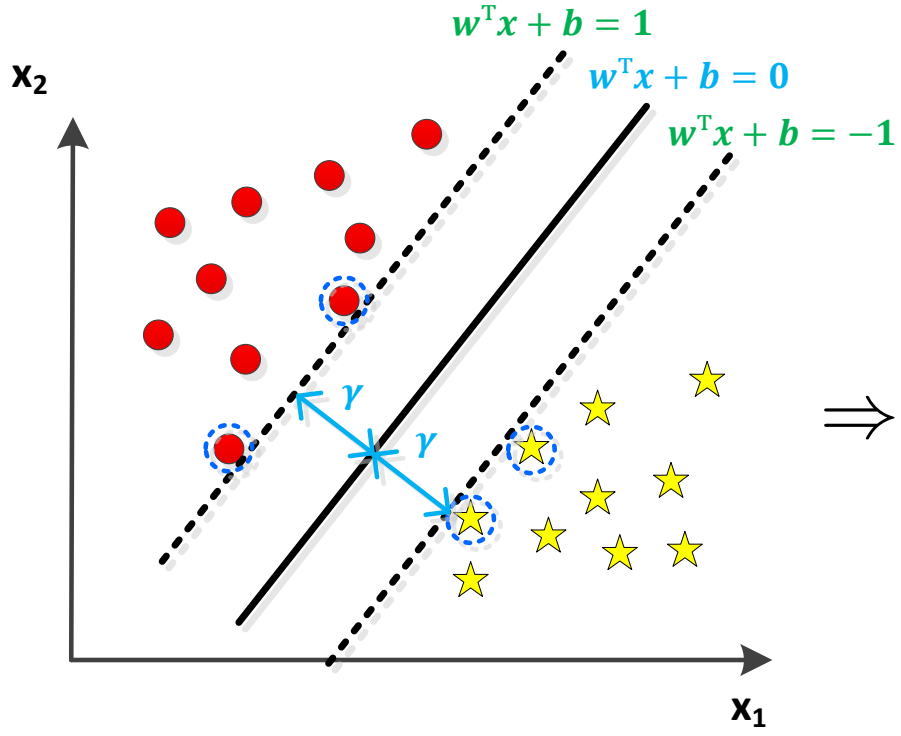
Large margin formulation

- Option 2:



Large margin formulation

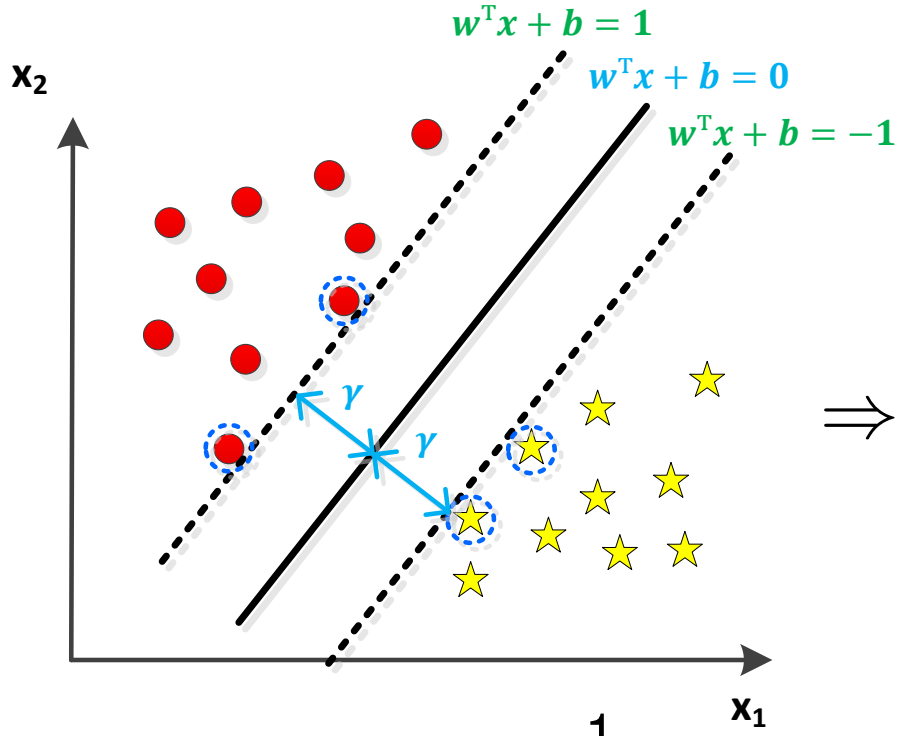
- Option 2:



$$\Rightarrow \gamma = \frac{y(w^\top x + b)}{||w||} = \frac{1}{||w||}$$

Large margin formulation

- Option 2:



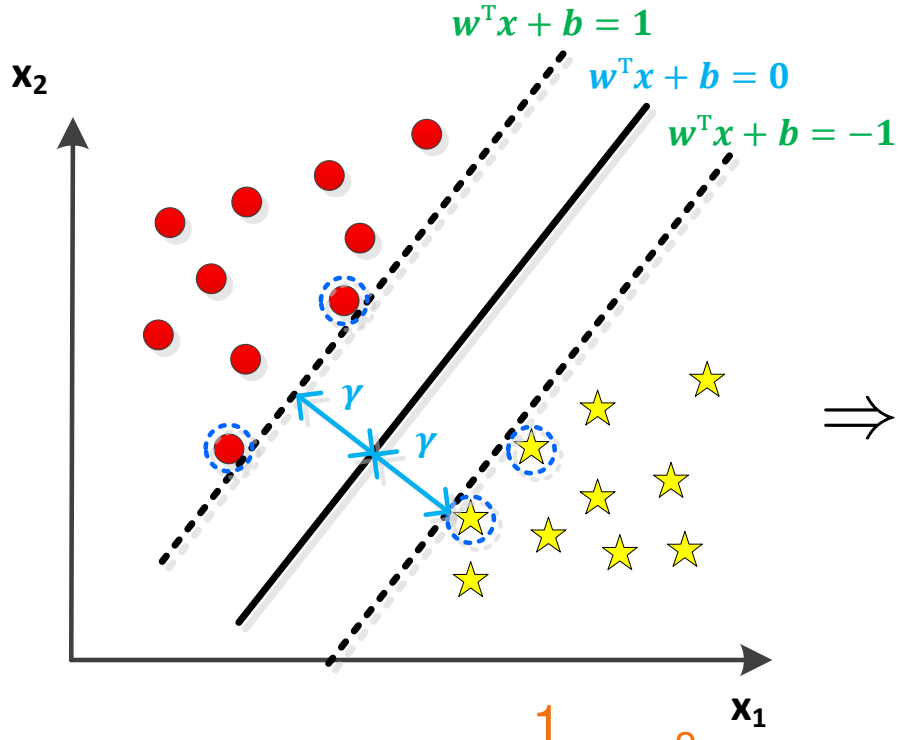
$$\Rightarrow \gamma = \frac{y(w^\top x + b)}{||w||} = \frac{1}{||w||}$$

$$\max_{w,b} \frac{1}{||w||}$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, m$$

Large margin formulation

- Option 2:



$$\Rightarrow \gamma = \frac{y(w^\top x + b)}{||w||} = \frac{1}{||w||}$$

$$\min_{w,b} \frac{1}{2}||w||_2^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, m$$

Support vector machines (SVMs)

Optimization Formulation

$$\min_{w,b} \ \frac{1}{2}||w||_2^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, m$$

- Maximize the margin while correctly classifying all examples correctly.
- Maximize the margin ($= \frac{1}{||w||}$) $\Leftrightarrow$ minimize $||w||_2^2 \Rightarrow$ prefer simple model.

Optimization

Optimization formulation

Primal optimization problem

$$\min_{w,b} \ \frac{1}{2}||w||_2^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, m$$

- An instance of convex optimization – admits a unique solution.
- Minimize a quadratic function with m linear constraints.
- An instance of quadratic programming (QP) – 'off-the-shelf" packages are available (e.g., quadprog (MATLAB), CVXOPT).

Digression: constrained optimization

Optimization with inequality constraints

$$\min_w \ f(w)$$
$$\text{s.t.} \quad g_i(w) \leq 0, \ \forall i = 1, \ldots, m$$

Method of Lagrange multipliers:

Lagrangian: $\mathcal{L}(w, \alpha) = f(w) + \sum_{i=1}^{m} \alpha_i g_i(w)$

s.t. $\alpha \succeq 0$

Digression: constrained optimization

Optimization with inequality constraints

$$\begin{aligned} &\min_w \ f(w) \\ &\text{s.t.} \quad g_i(w) \leq 0, \ \forall i = 1, \ldots, m \end{aligned}$$

Method of Lagrange multipliers:

Lagrangian: $\mathcal{L}(w, \alpha) = f(w) + \sum_{i=1}^{m} \alpha_i g_i(w)$

s.t. $\alpha \succeq 0$

Let $\theta_{\mathcal{P}}(w)$ be defined as

$$\begin{aligned} \theta_{\mathcal{P}}(w) &= \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha) \\ &= \begin{cases} f(w), & \text{if } w \text{ satisfies primal constraints} \\ \infty, & \text{otherwise.} \end{cases} \end{aligned}$$

Lagrangian primal problem

Original (primal) problem

$$\begin{aligned} &\min_w \ f(w) && (1)\\ &\text{s.t.} \quad g_i(w) \leq 0, \ \forall i = 1, \ldots, m \end{aligned}$$

Lagrangian primal problem:

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha) \qquad (2)$$

Lagrangian primal problem

Original (primal) problem

$$\begin{aligned} &\min_w \ f(w) && (1)\\ &\text{s.t.} \quad g_i(w) \leq 0, \ \forall i = 1, \ldots, m \end{aligned}$$

Lagrangian primal problem:

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha) \qquad (2)$$

Claim: Problem (1) is equivalent to problem (2).

Lagrangian dual problem

- Consider a slightly different problem:

$$\theta_{\mathcal{D}}(\alpha) = \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

- Consider a slightly different problem:

$$\theta_{\mathcal{D}}(\alpha) = \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

$$\max_{\alpha \succeq 0} \theta_{\mathcal{D}}(\alpha) = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

- Consider a slightly different problem:

$$\theta_{\mathcal{D}}(\alpha) = \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

$$\max_{\alpha \succeq 0} \theta_{\mathcal{D}}(\alpha) = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian primal problem

$$\min_{w} \theta_{\mathcal{P}}(w) = \min_{w} \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

- Consider a slightly different problem:

$$\theta_{\mathcal{D}}(\alpha) = \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

$$\max_{\alpha \succeq 0} \theta_{\mathcal{D}}(\alpha) = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian primal problem

$$\min_{w} \theta_{\mathcal{P}}(w) = \min_{w} \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha)$$

$$d^* = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha) \leq \min_{w} \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha) = p^*$$

Lagrangian dual problem

- Consider a slightly different problem:

$$\theta_{\mathcal{D}}(\alpha) = \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian dual problem

$$\max_{\alpha \succeq 0} \theta_{\mathcal{D}}(\alpha) = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha)$$

Lagrangian primal problem

$$\min_{w} \theta_{\mathcal{P}}(w) = \min_{w} \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha)$$

$$d^* = \max_{\alpha \succeq 0} \min_{w} \mathcal{L}(w, \alpha) \leq \min_{w} \max_{\alpha \succeq 0} \mathcal{L}(w, \alpha) = p^*$$

For SVM, $d^* = p^*$.

Relation between primal and dual

- For $d^* = p^* = \mathcal{L}(w^*, \alpha^*)$, certain conditions should be satisfied, known as **Karush-Kuhn-Tucker (KKT)** conditions:

$$\frac{\partial}{\partial w}\mathcal{L}(w^*, \alpha^*) = 0 \quad \text{(C1)}$$

$$\alpha_i^* g_i(w^*) = 0, \ i = 1, \ldots, m \quad \text{(C2)}$$

$$g_i(w^*) \leq 0, \ i = 1, \ldots, m \quad \text{(C3)}$$

$$\alpha_i^* \geq 0 \ i = 1, \ldots, m \quad \text{(C4)}$$

- Eq. (C2) is called the KKT dual complementarity condition – if $\alpha_i^* > 0$, then $g_i(w^*) = 0$ (the constraint is active).

- Roadmap: Primal problem $\Rightarrow$ Lagrangian function $\mathcal{L}(w, \alpha)$ $\Rightarrow \min_w \mathcal{L}(w, \alpha) \Rightarrow$ dual problem.

Back to SVM

Original (primal) optimization problem

$$\min_{w,b} \ \frac{1}{2}||w||_2^2$$
$$\text{s.t. } 1 - y_i(w^\top x_i + b) \leq 0, \ \forall i = 1, \ldots, m$$

- Lagrangian function:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||_2^2 + \sum_{i=1}^{m} \alpha_i [1 - y_i(w^\top x_i + b)]$$

- KKT conditions:

$$\frac{\partial}{\partial w}\mathcal{L} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i$$
$$\frac{\partial}{\partial b}\mathcal{L} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0$$

Back to SVM

Plugging the KKT conditions back into $\mathcal{L}$:

Dual optimization problem

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \ i = 1, \ldots, n$$

Having found α

- By KKT condition

$$w = \sum_{i=1}^m \alpha_i y_i x_i$$

- For any support vector x_i, we have $y_i(w^\top x_i + b) = 1$:

$$b = y_i - w^\top x_i, \ \forall \alpha_i > 0 \quad \text{(i.e., for any support vector } x_i\text{)}$$

Two key observations from dual

Observation 1: Most α_i's are 0

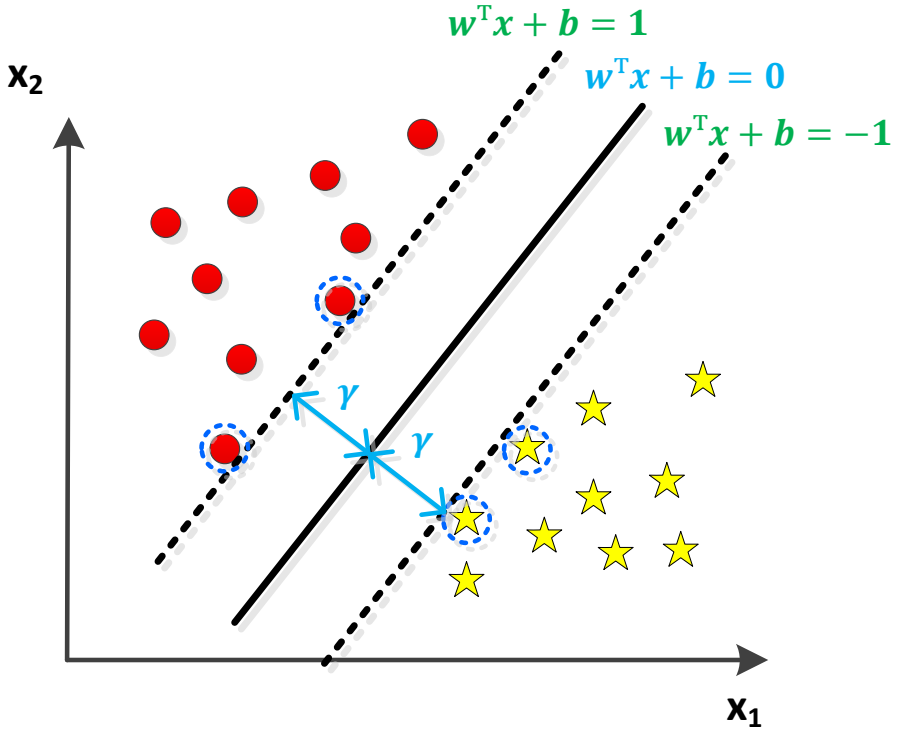
- KKT condition (C2):

$$\alpha_i[y_i(w^\top x_i + b) - 1] = 0$$

- If x_i is **not** on margin:

$$y_i(w^\top x_i + b) > 1$$
$$\Rightarrow \alpha_i = 0$$

- $\alpha_i > 0$ only for x_i on margin.
- These are **support vectors.**
- Only support vectors matter.



Two key observations from dual

Observation 2: Only inner product matters

- Dual optimization problem

$$\max_{\alpha} \ \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t. } \sum_{i=1}^{m} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \ i = 1, \ldots, m$$

- Primal solution: $w = \sum_{i=1}^{m} \alpha_i y_i x_i$
- For prediction:

$$f(x) = \text{sgn}\left(w^\top x + b\right) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \left(x_i^\top x\right) + b \right)$$

- Replace the inner products with kernel functions

$$x_i^\top x_j \Rightarrow k(x_i, x_j)$$

to produce **non-linear SVMs.**

Kernel Trick

Kernel functions

- Whenever a learning algorithm can be written in terms of inner products, there is a kernel version of this algorithm.
- Given a (nonlinear) feature mapping $\phi : x \to \phi(x)$, its kernel is the function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:

$$k(x_i, x_j) = \phi(x_i)^\top \phi(x_j)$$

- Conversely, by choosing the kernel function k, we implicitly choose a feature mapping ϕ.
- A kernel function is the inner product in the space defined by ϕ – a notion of similarity.

Dual formulation of SVMs

Training

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \ i = 1, \ldots, n$$

Prediction

$$f(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i (x_i^\top x) + b \right)$$

Dual formulation of SVMs

Training

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \ i = 1, \ldots, n$$

Prediction

$$f(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i \left(x_i^\top x\right) + b \right)$$

Kernel trick: a feature mapping $\phi : x \to \phi(x)$

$$x_i^\top x_j \to \phi(x_i)^\top \phi(x_j) \Leftrightarrow k(x_i, x_j)$$

In the dual formulation, feature mapping ϕ is not needed either to learn or to make predictions!

The kernel trick

Training

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \ i = 1, \ldots, n$$

Prediction

$$f(x) = \text{sgn}\left(\sum_{i=1}^m \alpha_i y_i k(x_i, x) + b \right)$$

Kernel trick: a feature mapping $\phi : x \to \phi(x)$

$$x_i^\top x_j \to \phi(x_i)^\top \phi(x_j) \Leftrightarrow k(x_i, x_j)$$

In the dual formulation, feature mapping ϕ is not needed either to learn or to make predictions!

Commonly used kernels

- Linear kernels – linear SVMs.

$$k(x_i, x_j) = x_i^\top x_j$$

- Gaussian kernels

$$k(x_i, x_j) = e^{-\frac{||x_i - x_j||_2^2}{2\sigma^2}}$$

- Polynomial kernels

$$k(x_i, x_j) = \left(1 + x_i^\top x_j\right)^d$$

Commonly used kernels

- Linear kernels – linear SVMs.

$$k(x_i, x_j) = x_i^\top x_j$$

- Gaussian kernels

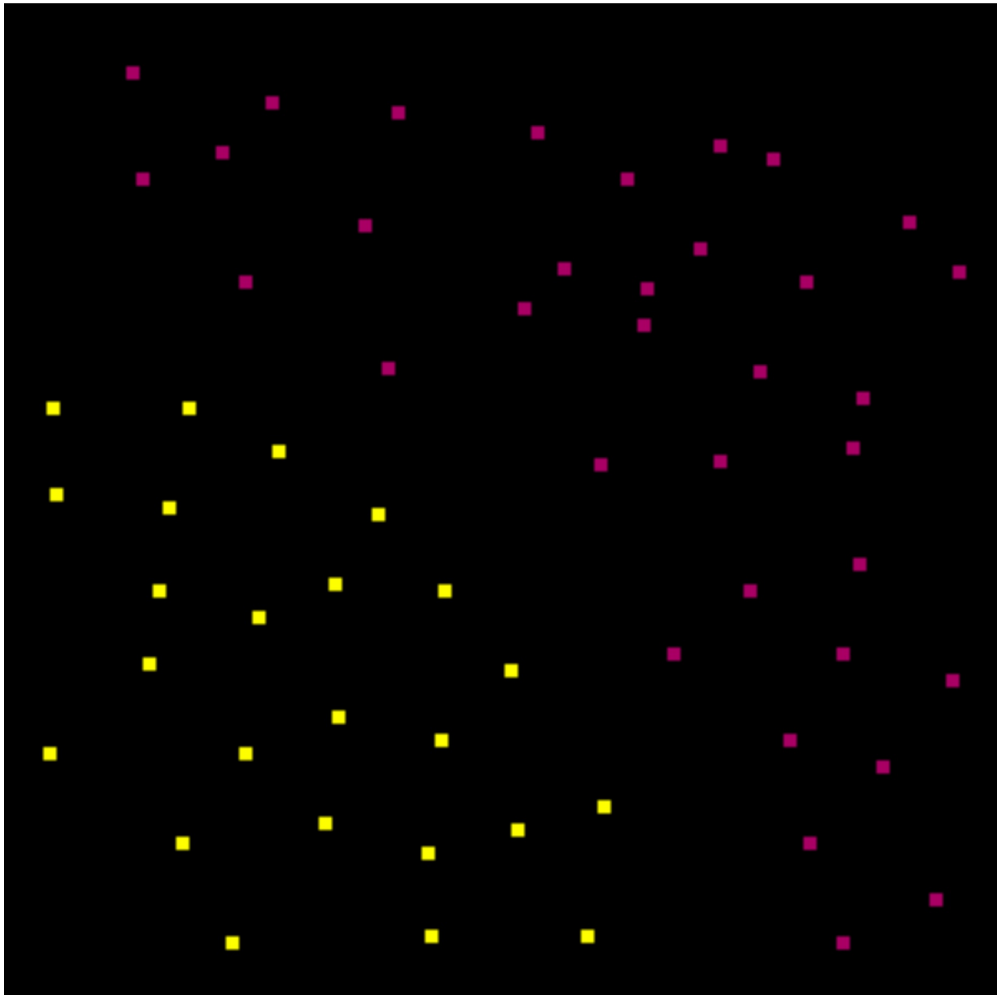
$$k(x_i, x_j) = e^{-\frac{||x_i - x_j||_2^2}{2\sigma^2}}$$

- Polynomial kernels

$$k(x_i, x_j) = \left(1 + x_i^\top x_j\right)^d$$

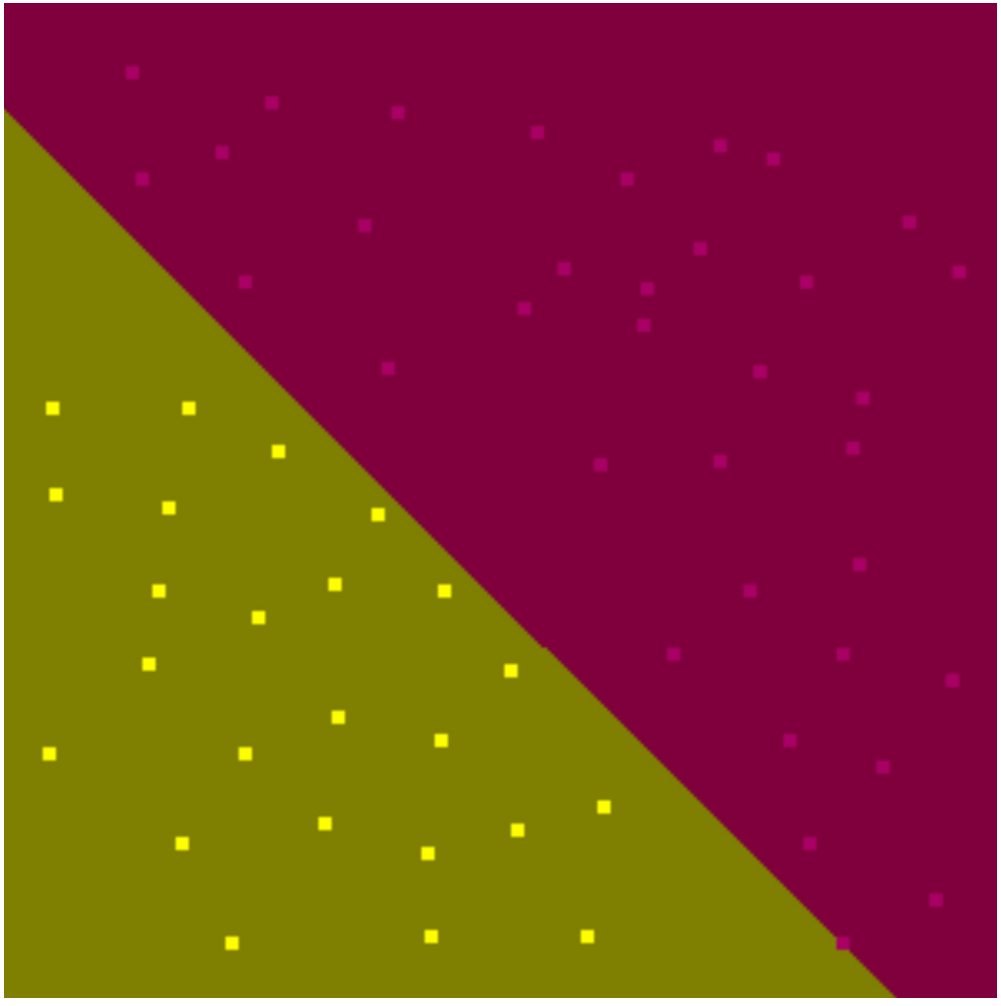
- We can impose prior knowledge by designing problem-dependent kernels (e.g., $k(x_i, x_j) = \exp(-\frac{\sin ||x_i - x_j||_2^2 / T}{2\sigma^2})$ for periodic time series)!

Effect of kernel functions



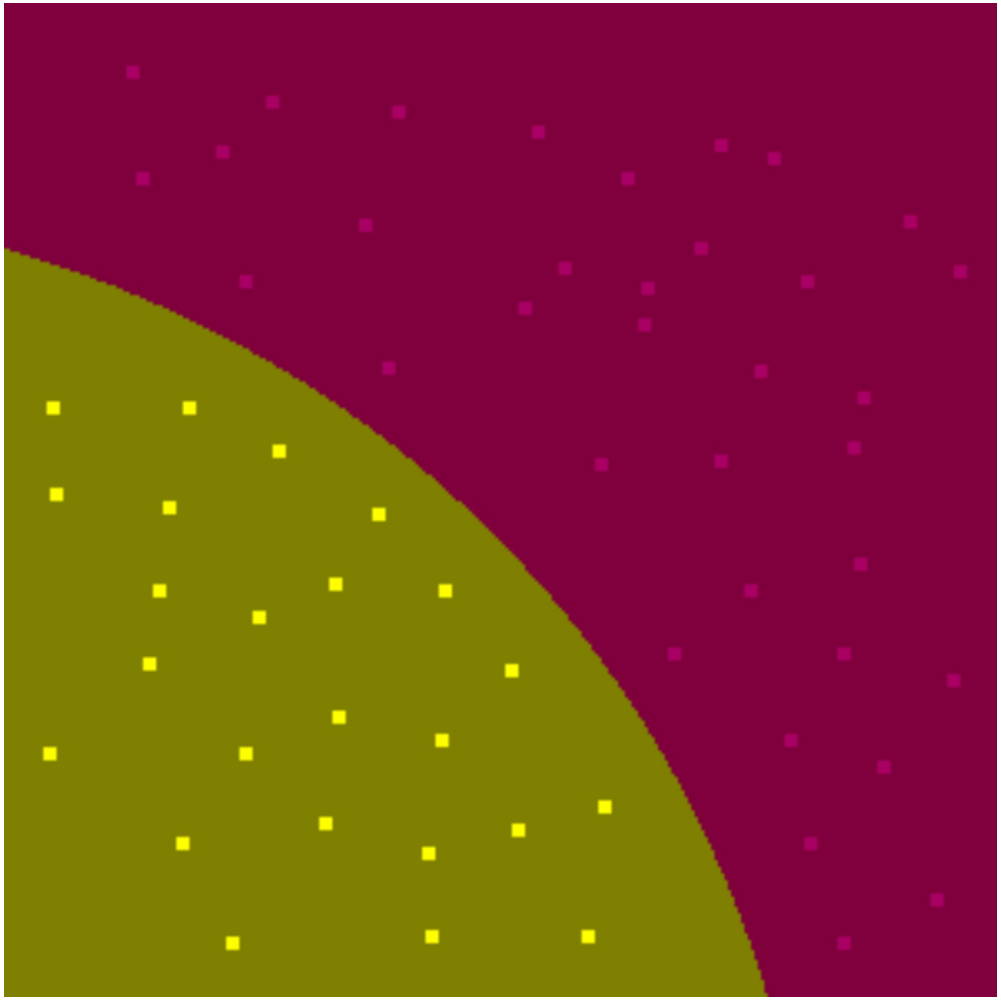
Training data

Effect of kernel functions



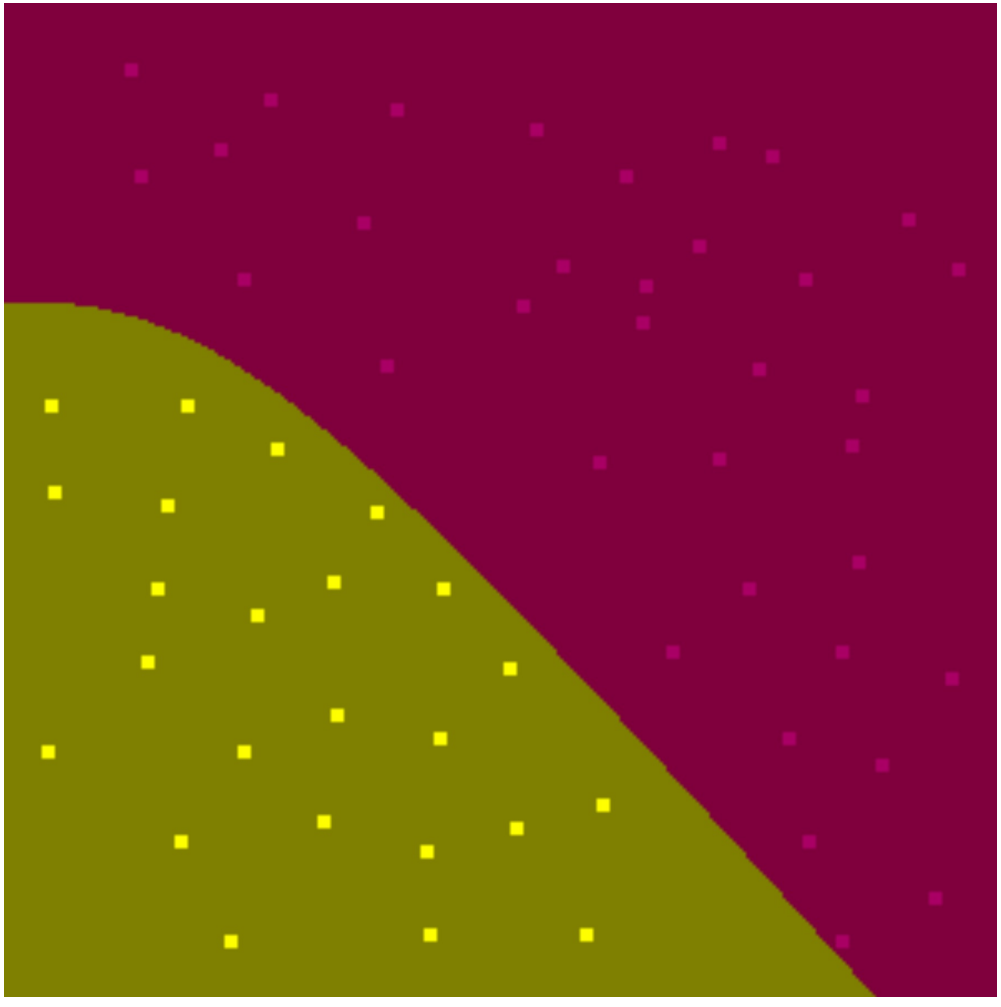
Linear kernel

Effect of kernel functions



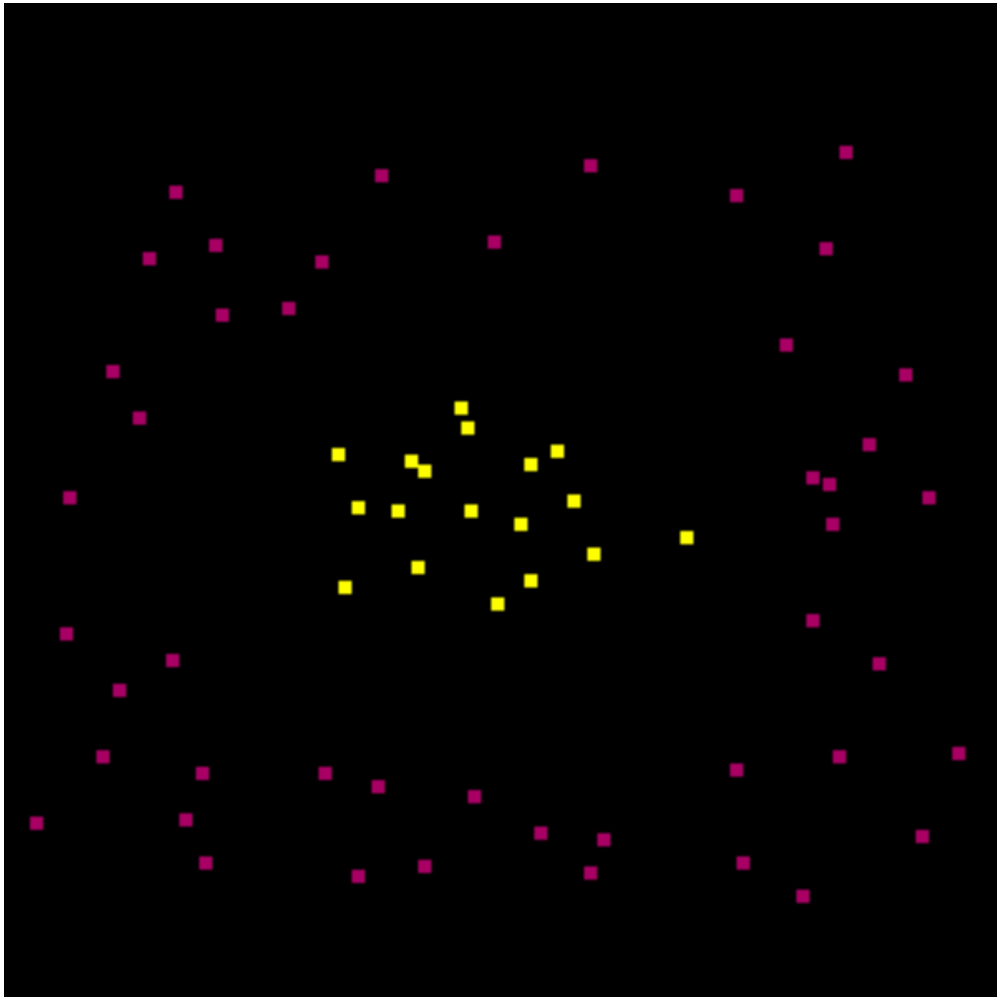
Gaussian kernel

Effect of kernel functions



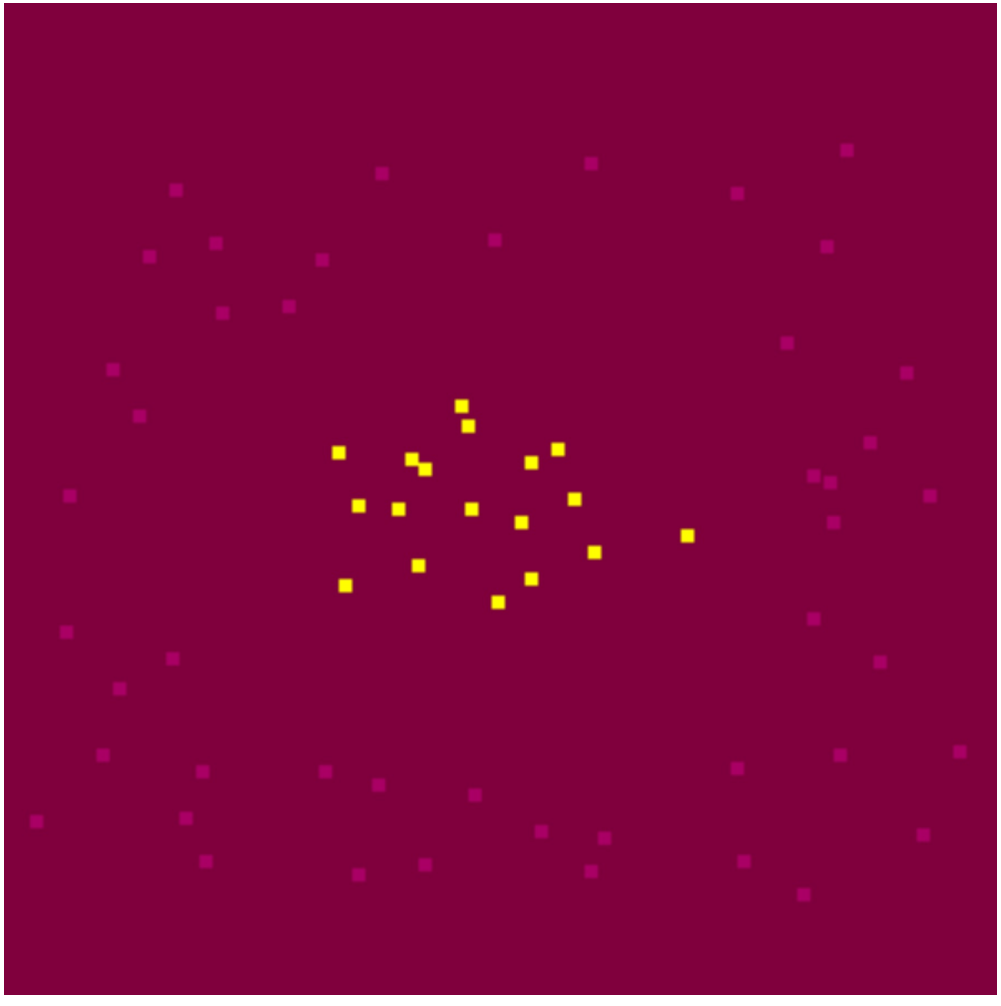
Polynomial kernel

Effect of kernel functions



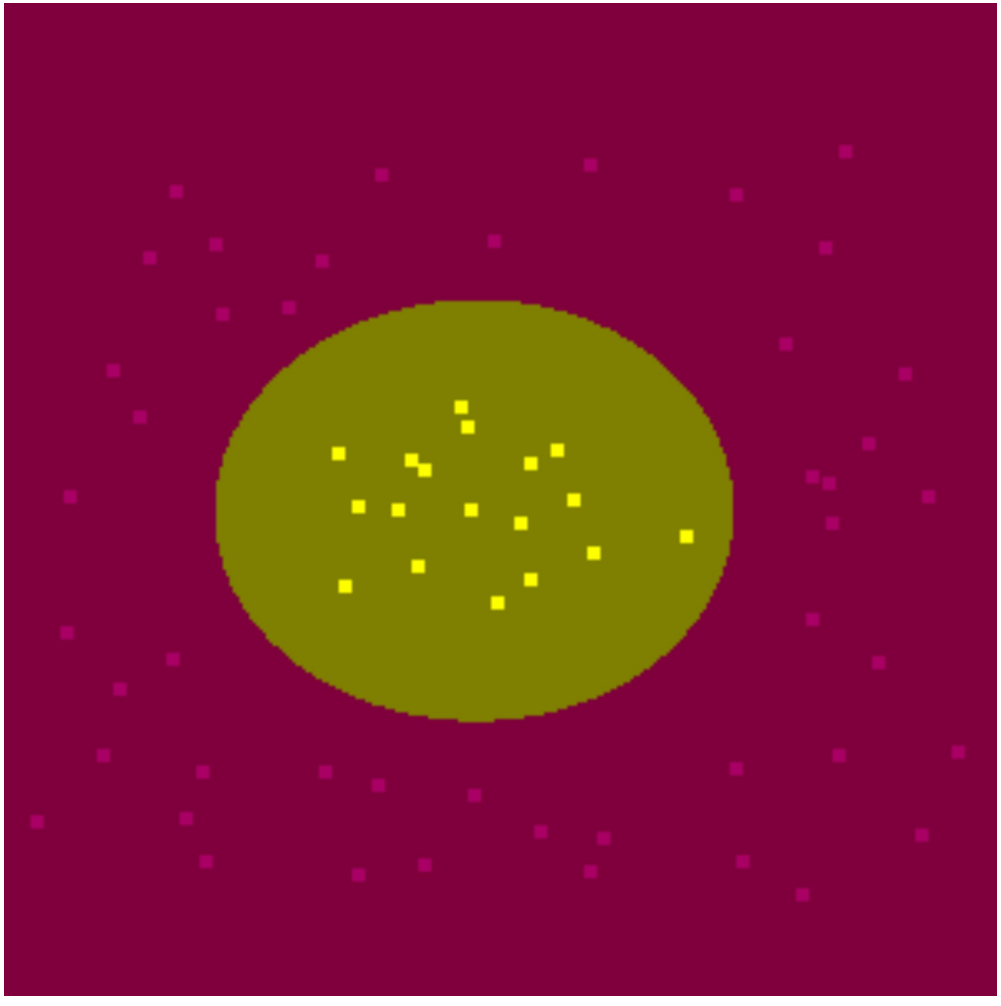
Training data

Effect of kernel functions



Linear kernel

Effect of kernel functions

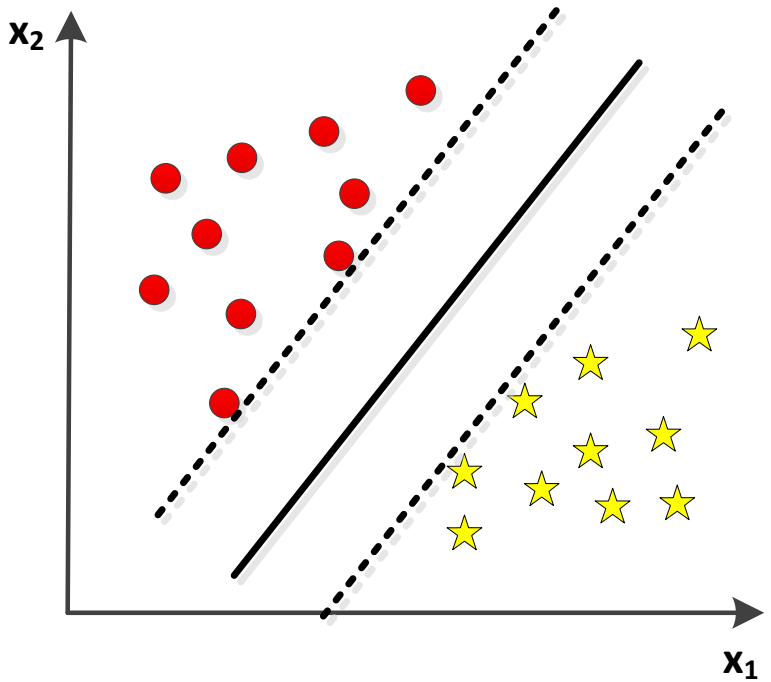


Gaussian kernel

Non-separable Case

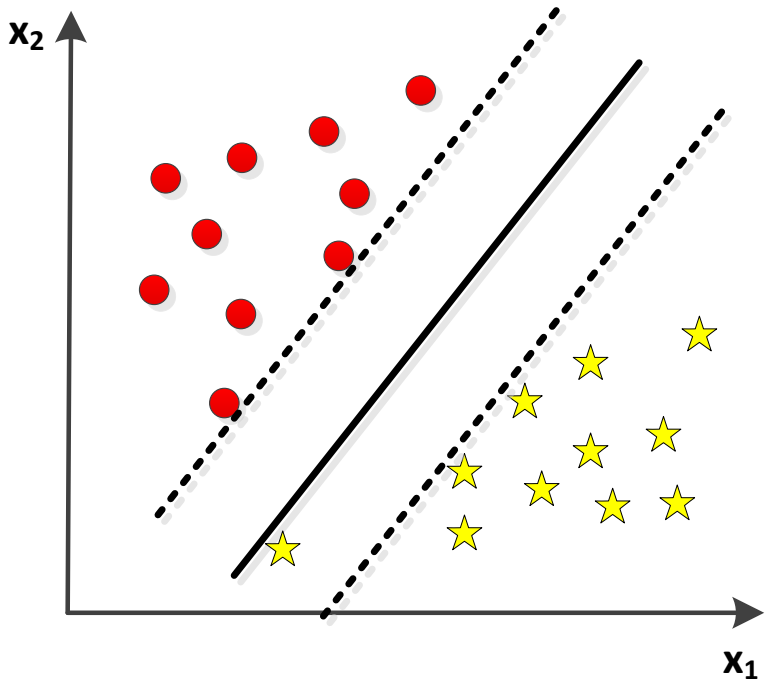
Non-separable case

- The derivation of SVMs as presented so far assumed that the data is linearly separable.
- What if the data is not separable?



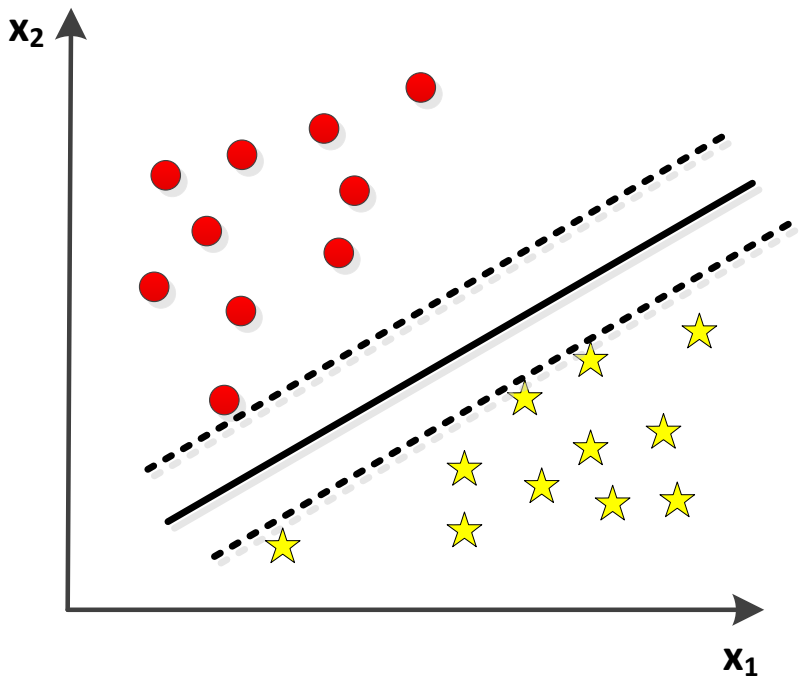
Non-separable case

- The derivation of SVMs as presented so far assumed that the data is linearly separable.
- What if the data is not separable?



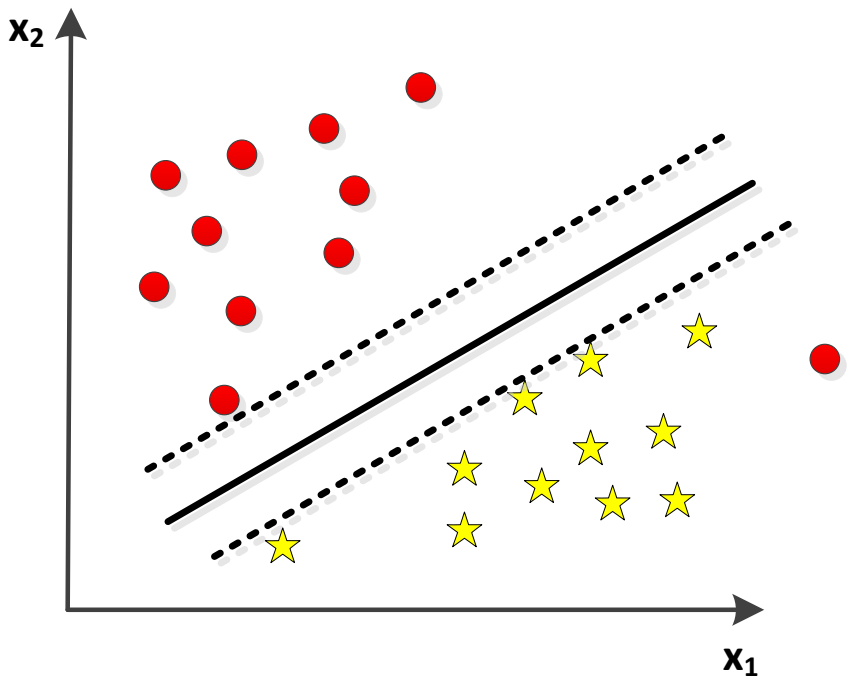
Non-separable case

- The derivation of SVMs as presented so far assumed that the data is linearly separable.
- What if the data is not separable?



Non-separable case

- The derivation of SVMs as presented so far assumed that the data is linearly separable.
- What if the data is not separable?



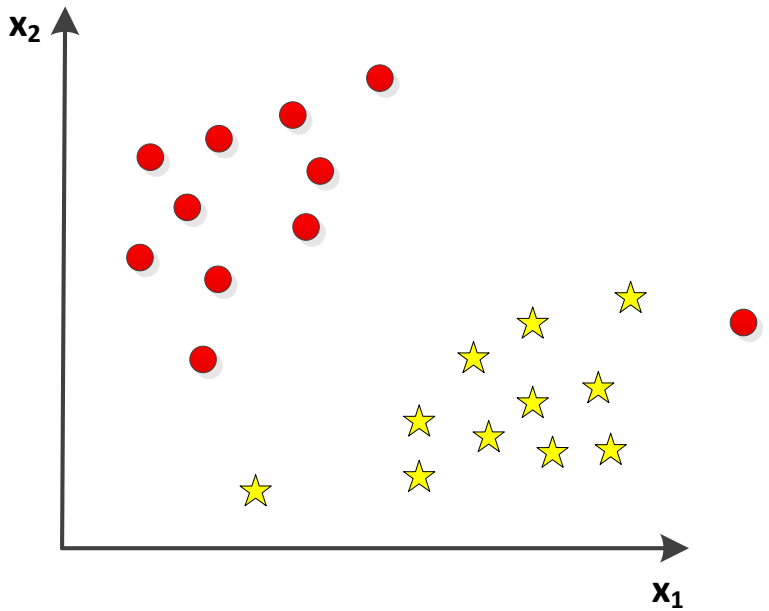
Introducing slack variables

- Non-separable $\Rightarrow$ for any separator $w^\top x + b = 0$, there exists x_i in training set such that

$$y_i(w^\top x_i + b) \ngeq 1$$

- Solution: introducing slack variables $\xi_i \geq 0$ – permit SVMs to make mistakes.

$$y_i(w^\top x_i + b) \geq 1 - \xi_i$$



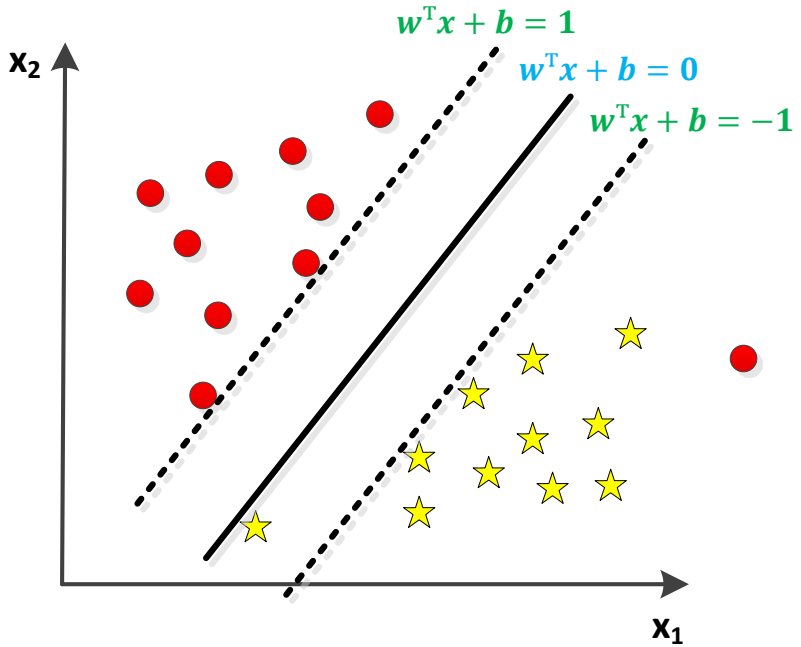
Introducing slack variables

- Non-separable $\Rightarrow$ for any separator $w^\top x + b = 0$, there exists x_i in training set such that

$$y_i(w^\top x_i + b) \ngeq 1$$

- Solution: introducing slack variables $\xi_i \geq 0$ – permit SVMs to make mistakes.

$$y_i(w^\top x_i + b) \geq 1 - \xi_i$$



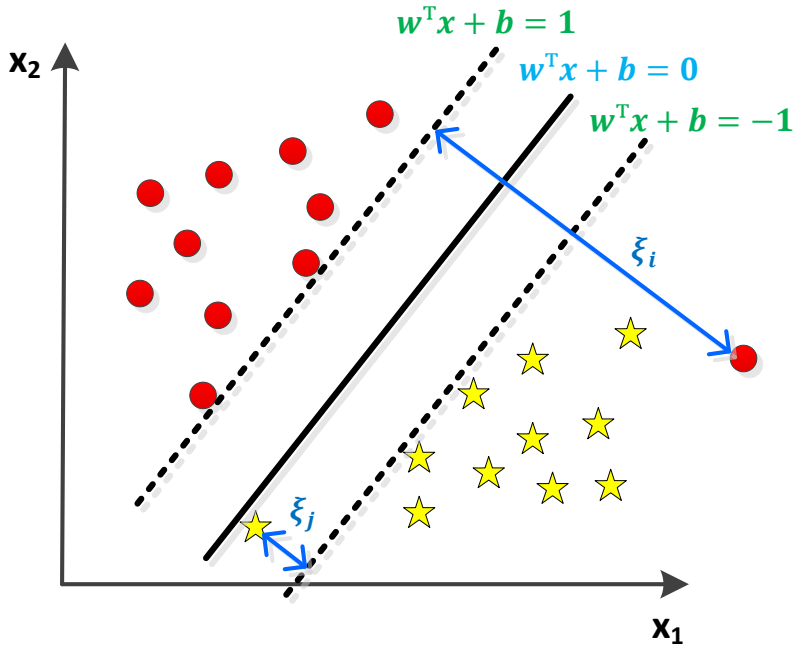
Introducing slack variables

- Non-separable $\Rightarrow$ for any separator $w^\top x + b = 0$, there exists x_i in training set such that

$$y_i(w^\top x_i + b) \ngeq 1$$

- Solution: introducing slack variables $\xi_i \geq 0$ – permit SVMs to make mistakes.

$$y_i(w^\top x_i + b) \geq 1 - \xi_i$$



Introducing slack variables (primal)

Primal optimization problem for separable cases

$$\min_{w,b} \ \frac{1}{2}||w||_2^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, n$$

- New constraints with slack variables:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \text{s.t. } \xi_i \geq 0, \ i = 1, \ldots, n$$

Introducing slack variables (primal)

Primal optimization problem for separable cases

$$\min_{w,b} \ \frac{1}{2}||w||_2^2$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1, \ \forall i = 1, \ldots, n$$

- New constraints with slack variables:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \text{s.t. } \xi_i \geq 0, \ i = 1, \ldots, n$$

Primal optimization problem for non-separable cases

$$\min_{w,b,\xi} \ \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{m} \xi_i$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i, \ \forall i = 1, \ldots, n$$

$$\xi_i \geq 0, \ \forall i = 1, \ldots, n$$

Introducing slack variables (dual)

Dual optimization problem for separable cases

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t.} \ \sum_{i=1}^m \alpha_i y_i = 0$$

$$\alpha_i \geq 0, \ i = 1, \ldots, n$$

- New constraints with slack variables:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \text{s.t. } \xi_i \geq 0, \ i = 1, \ldots, n$$

Introducing slack variables (dual)

Dual optimization problem for separable cases

$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t.} \ \sum_{i=1}^m \alpha_i y_i = 0$$

$$\alpha_i \geq 0, \ i = 1, \ldots, n$$

- New constraints with slack variables:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \text{s.t. } \xi_i \geq 0, \ i = 1, \ldots, n$$

Dual optimization problem for non-separable cases

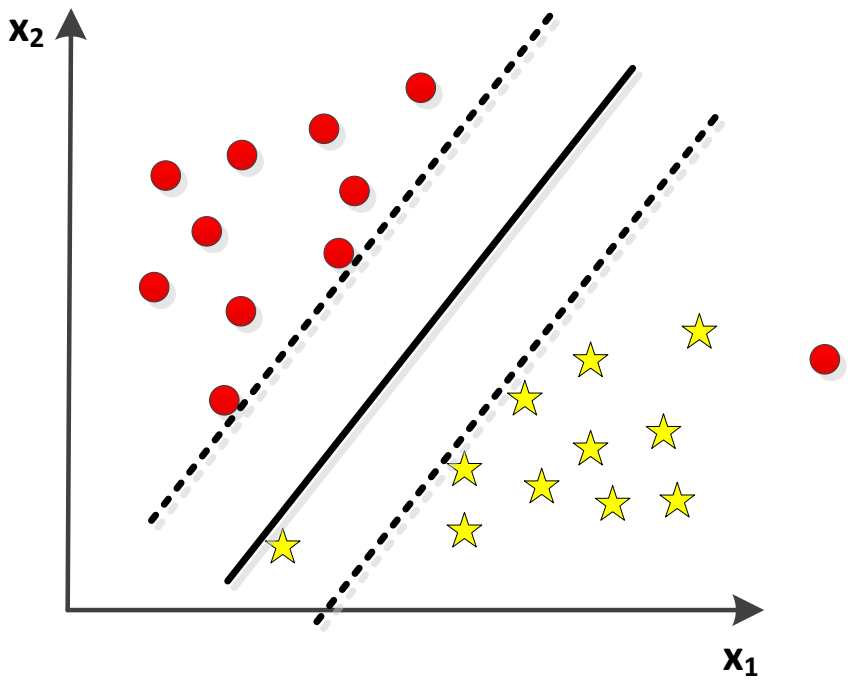
$$\max_\alpha \ \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j y_i y_j (x_i^\top x_j)$$

$$\text{s.t.} \ \sum_{i=1}^m \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \ i = 1, \ldots, n$$

Introducing slack variables (dual)

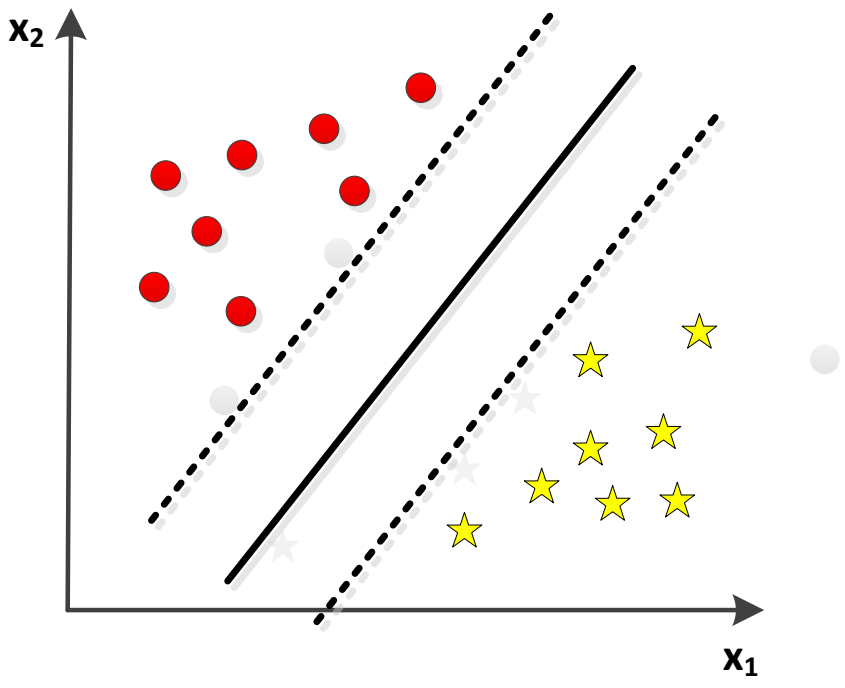
By KKT dual-complementarity conditions



Introducing slack variables (dual)

By KKT dual-complementarity conditions

$$\alpha_i = 0 \;\Rightarrow\; y_i(w^\top x_i + b) > 1 \;\Rightarrow\; \text{outside the margin}$$

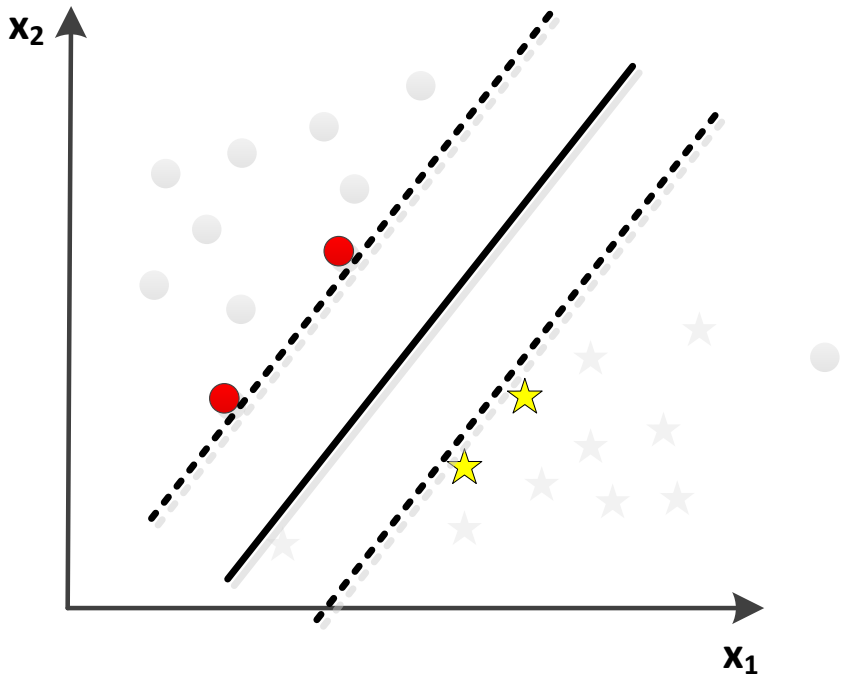


Introducing slack variables (dual)

By KKT dual-complementarity conditions

$$\alpha_i = 0 \Rightarrow y_i(w^\top x_i + b) > 1 \Rightarrow \text{outside the margin}$$

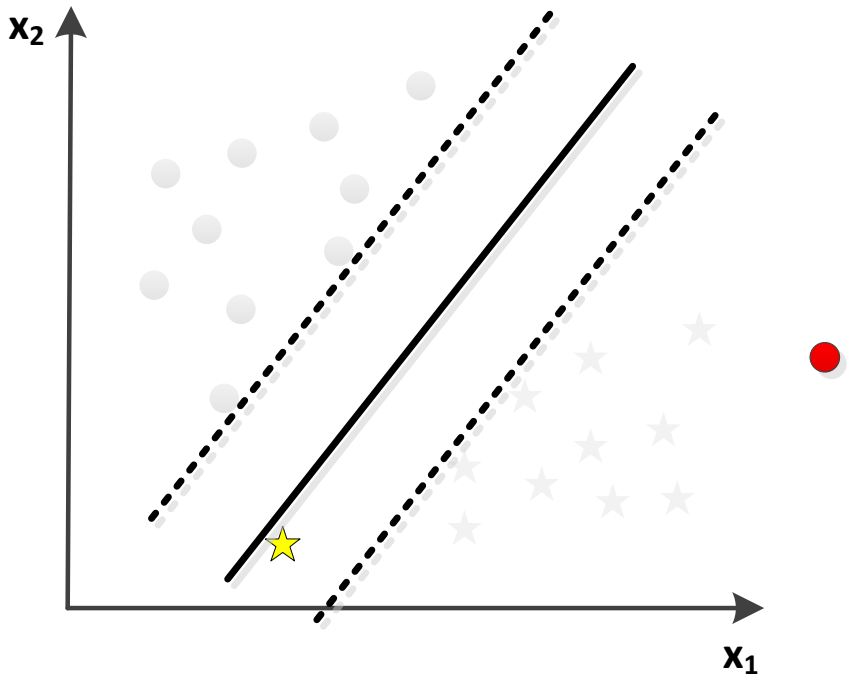
$$0 < \alpha_i < C \Rightarrow y_i(w^\top x_i + b) = 1 \Rightarrow \text{on the margin}$$



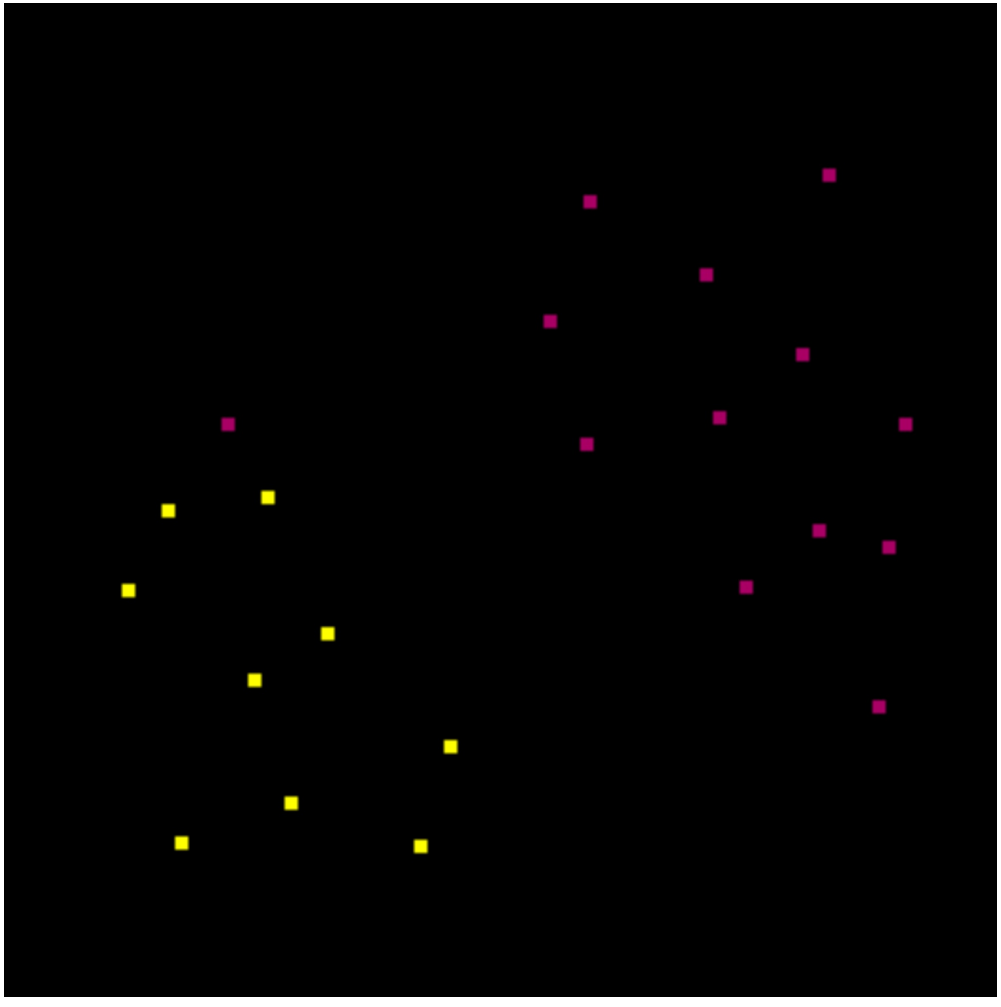
Introducing slack variables (dual)

By KKT dual-complementarity conditions

$$\alpha_i = 0 \Rightarrow y_i(w^\top x_i + b) > 1 \Rightarrow \text{outside the margin}$$
$$0 < \alpha_i < C \Rightarrow y_i(w^\top x_i + b) = 1 \Rightarrow \text{on the margin}$$
$$\alpha_i = C \Rightarrow y_i(w^\top x_i + b) < 1 \Rightarrow \text{inside the margin/misclassified}$$

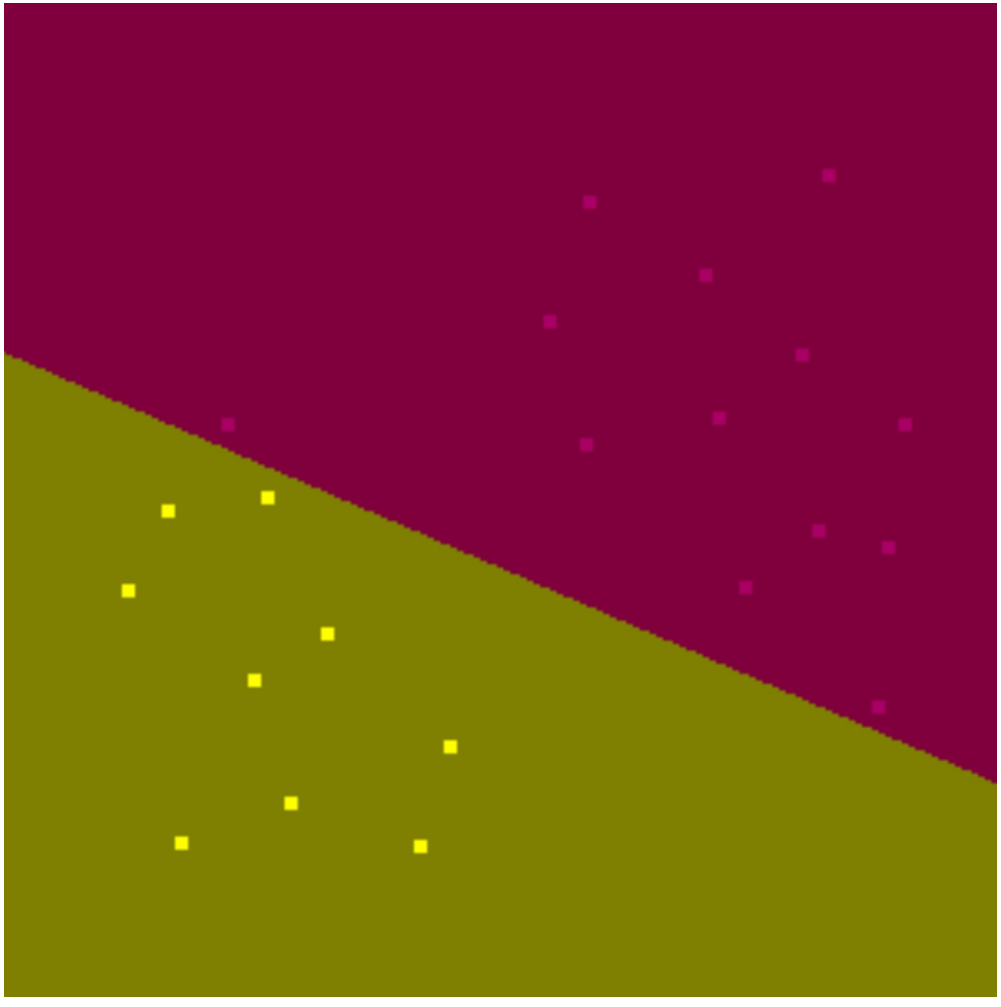


Effect of C



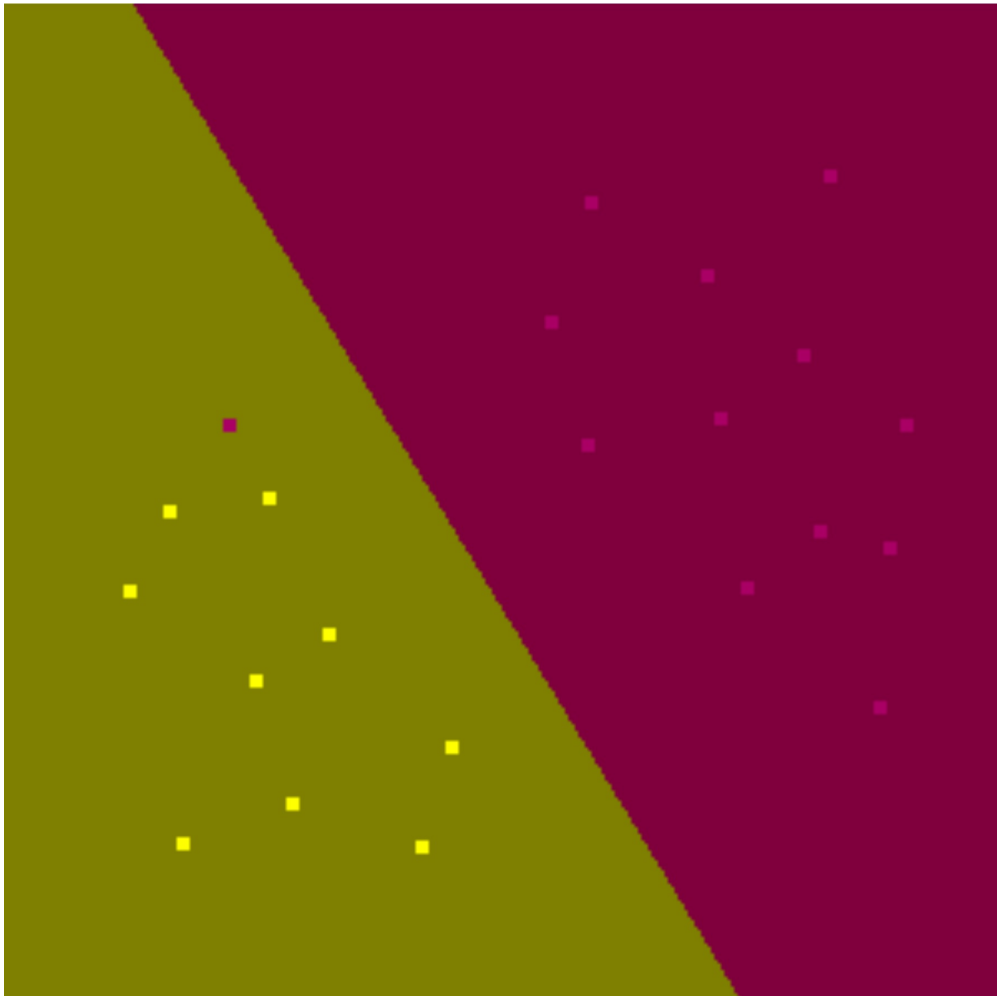
Training data

Effect of C



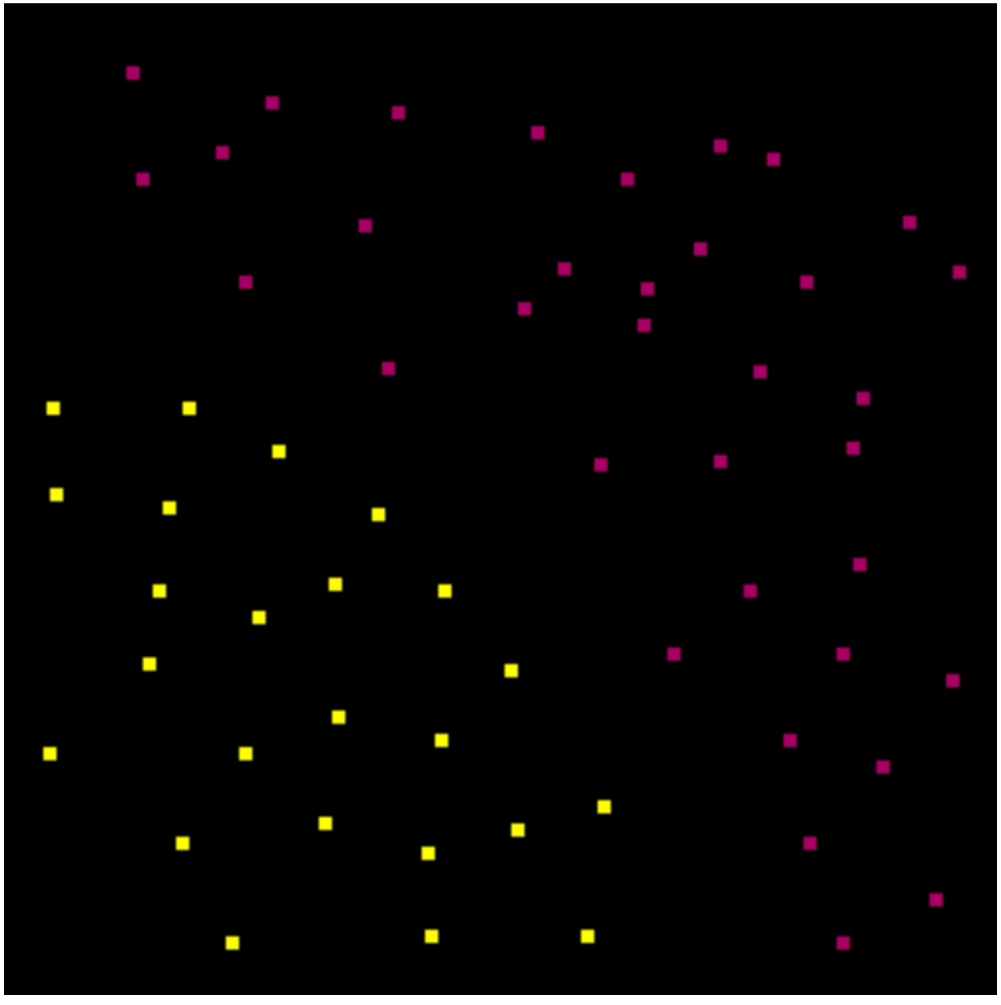
$C = 1000$

Effect of C



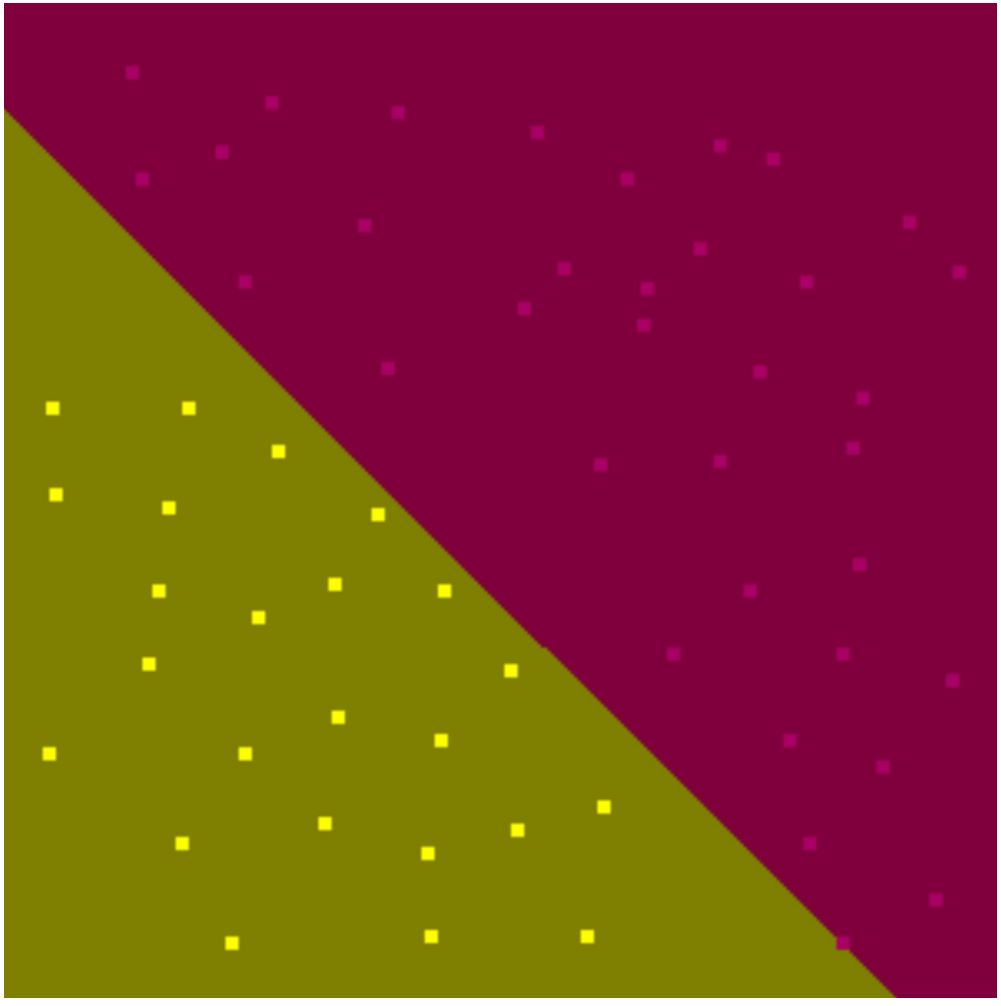
$C = 1$

Effect of kernel functions



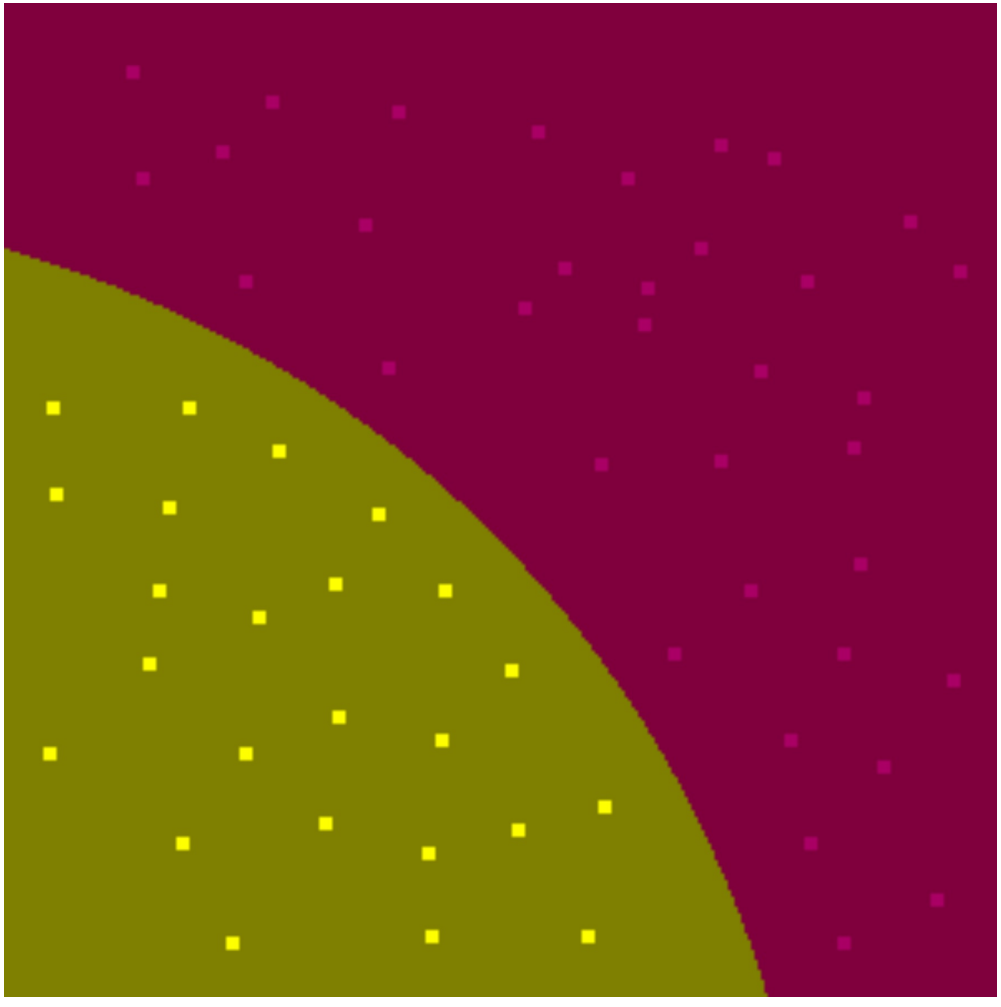
Training data

Effect of kernel functions



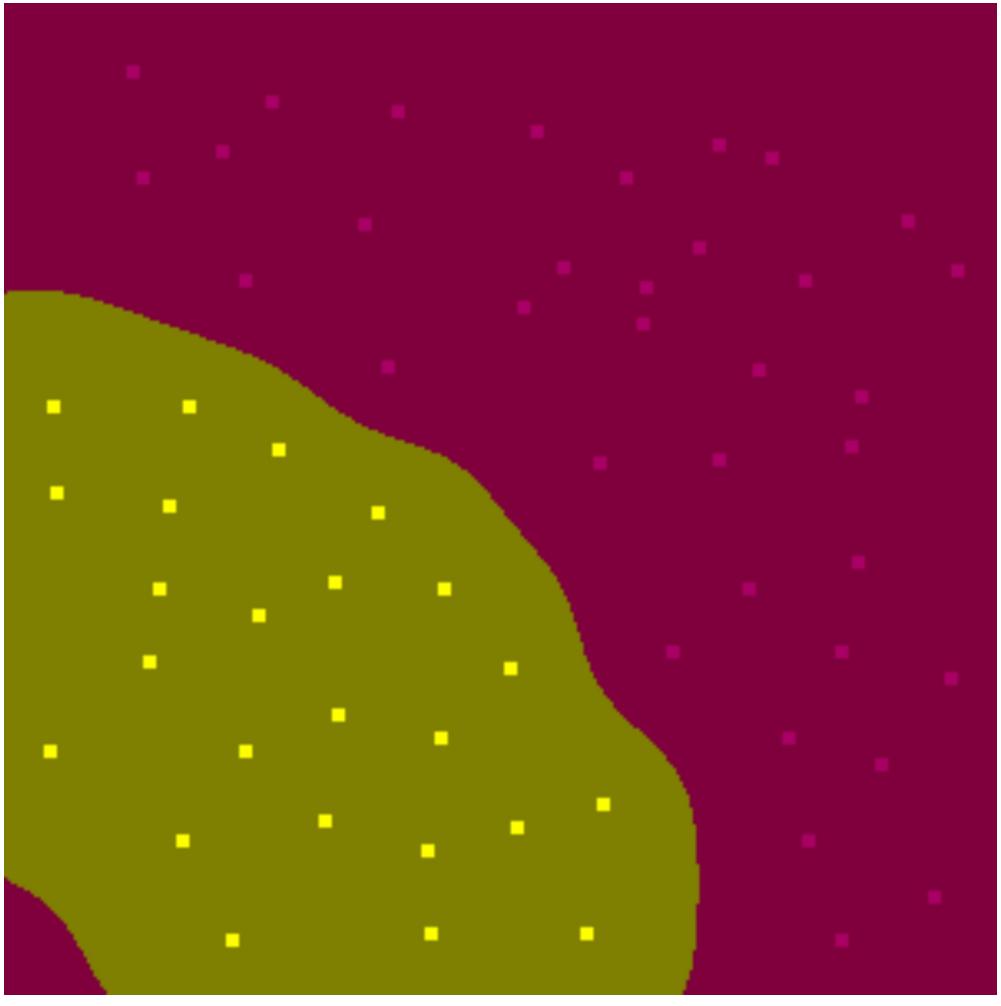
Linear kernel

Effect of kernel functions



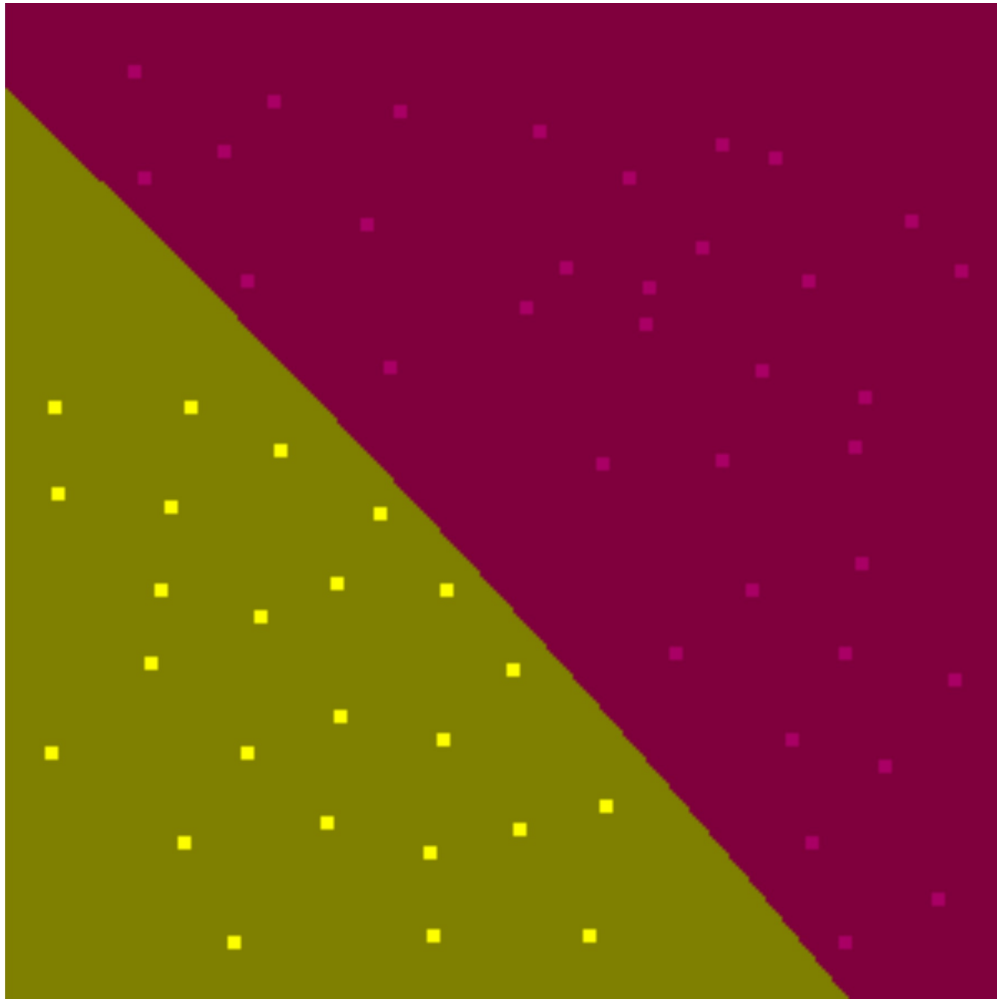
Gaussian kernel, $\sigma = 1$

Effect of kernel functions



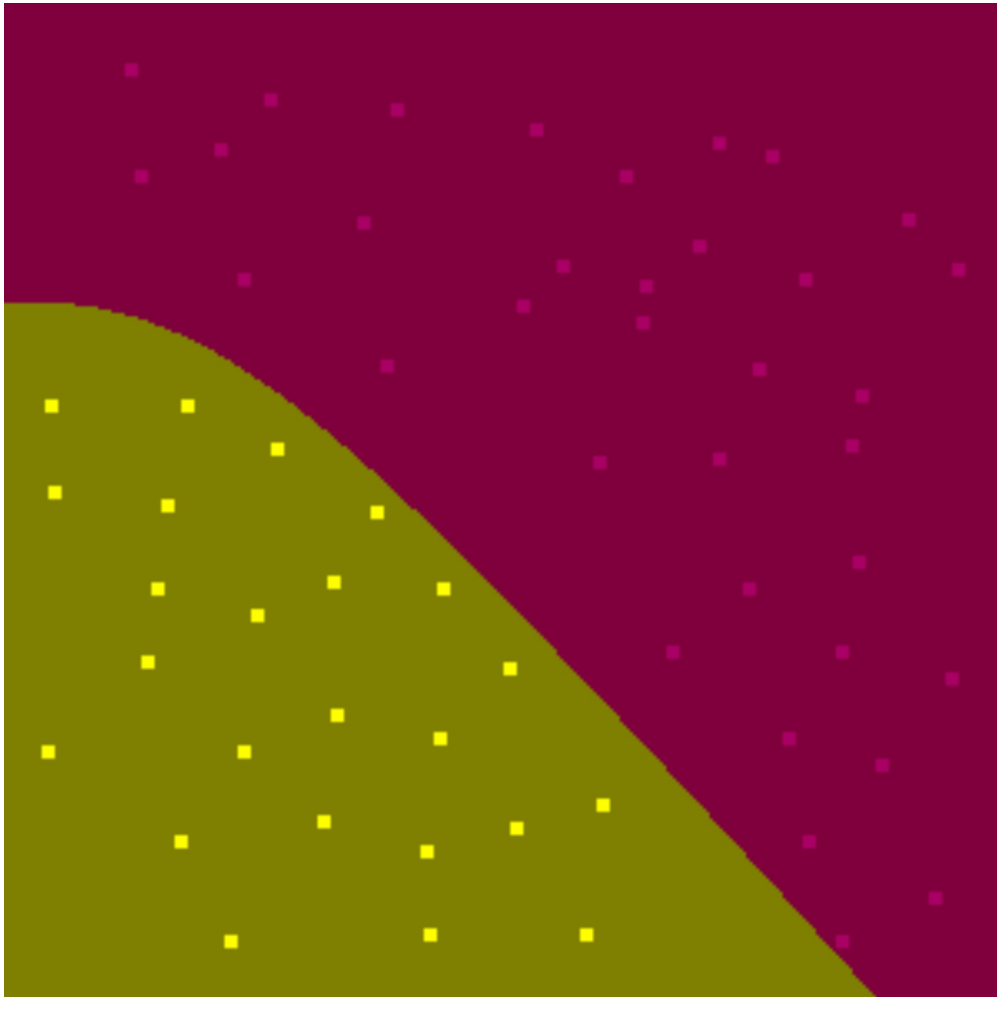
Gaussian kernel, $\sigma = 0.1$

Effect of kernel functions



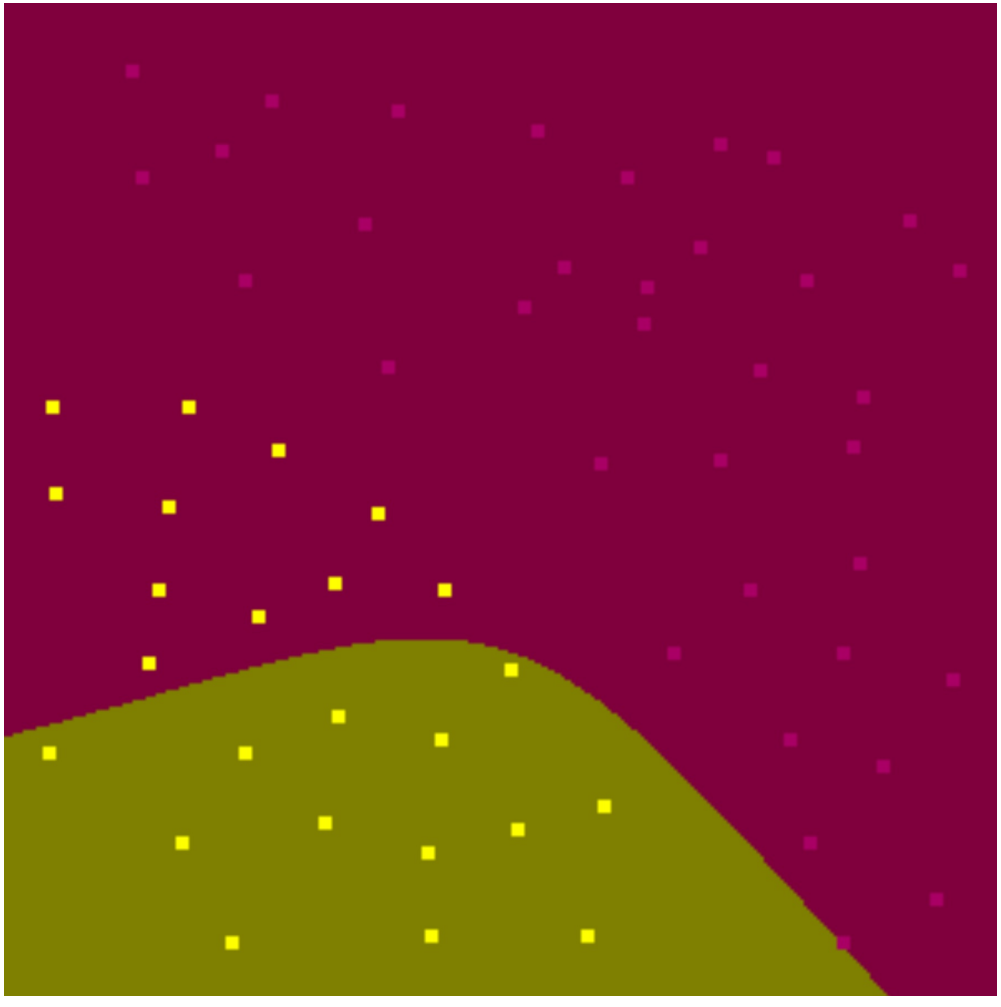
Gaussian kernel, $\sigma = 10$

Effect of kernel functions



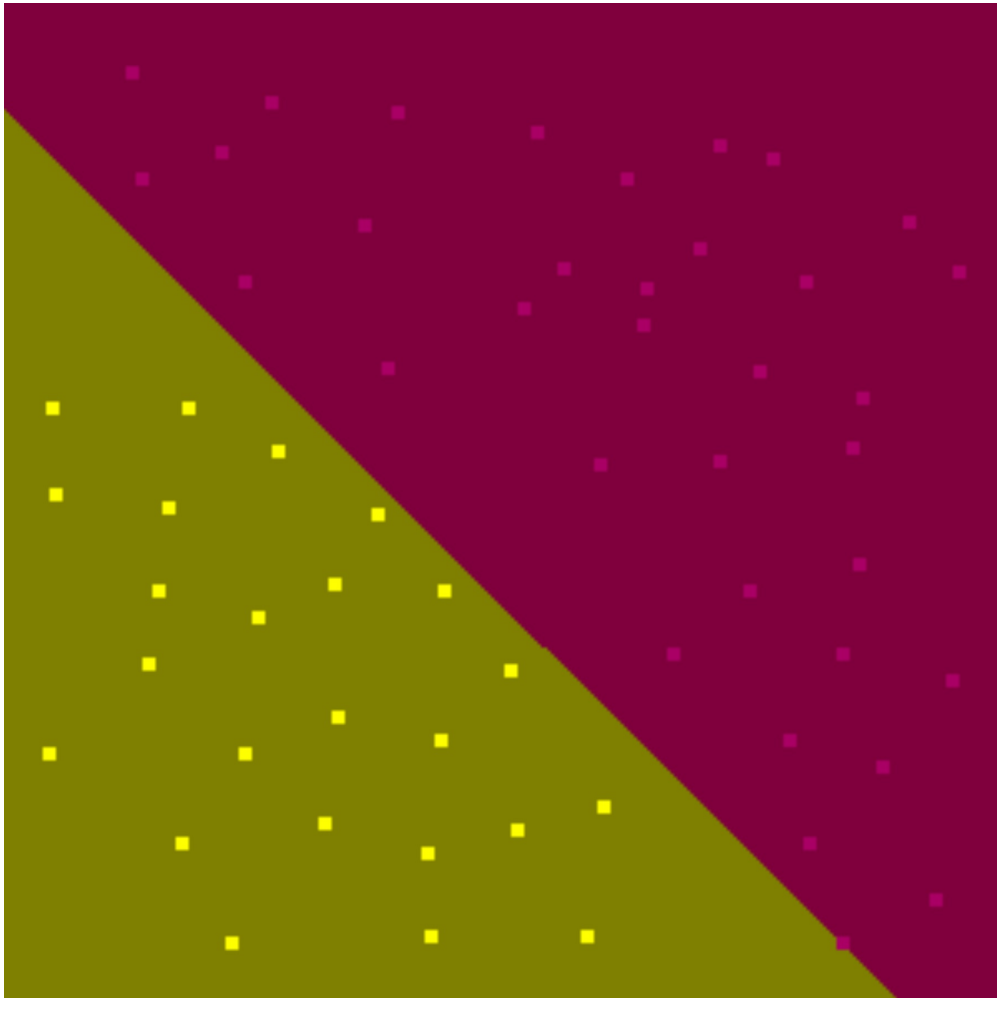
Polynomial kernel, $d = 3$

Effect of kernel functions



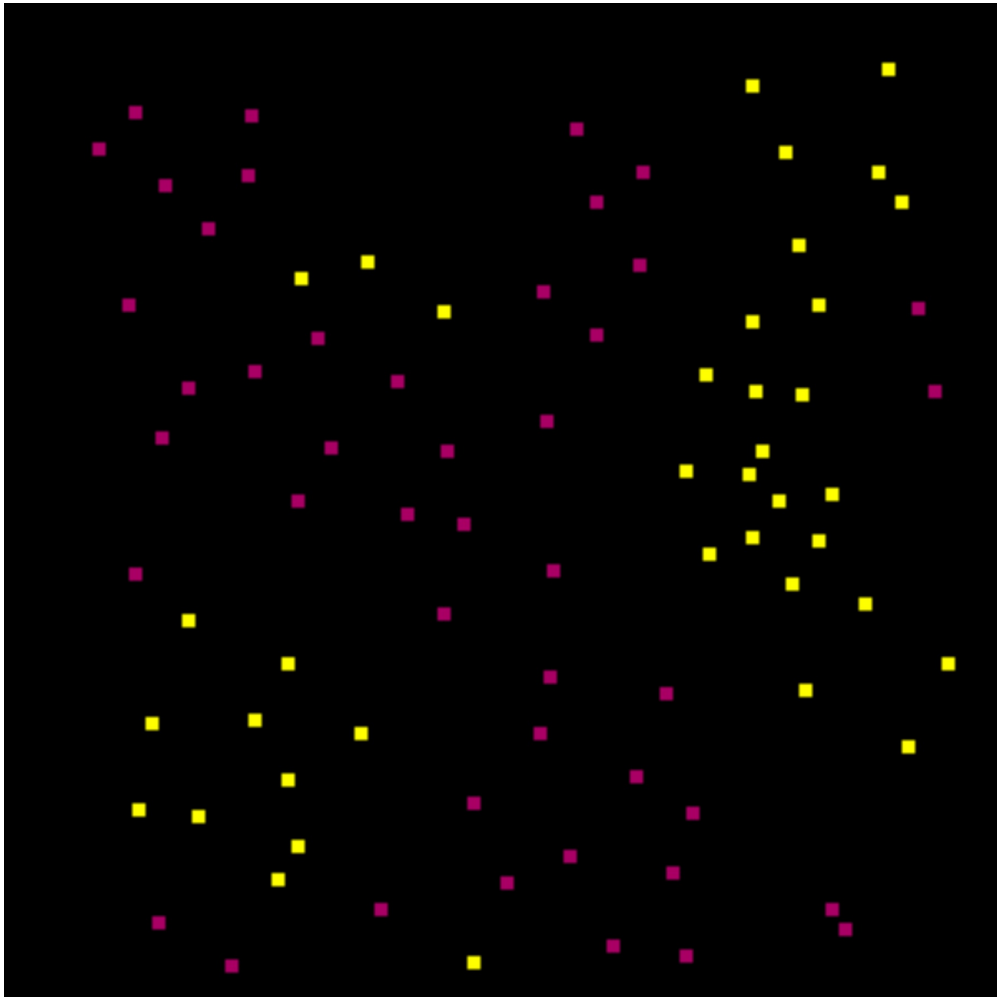
Polynomial kernel, $d = 10$

Effect of kernel functions

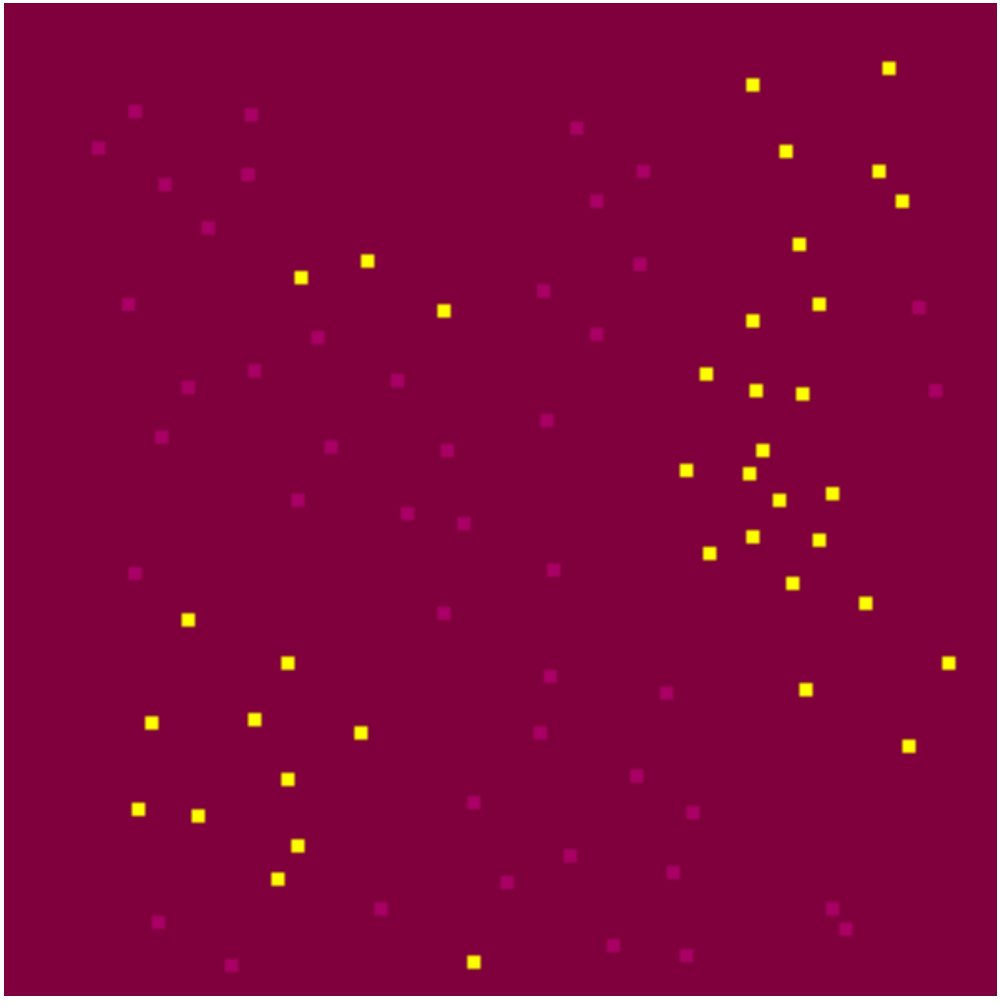


Polynomial kernel, $d = 1$

Effect of model parameters for Gaussian kernels

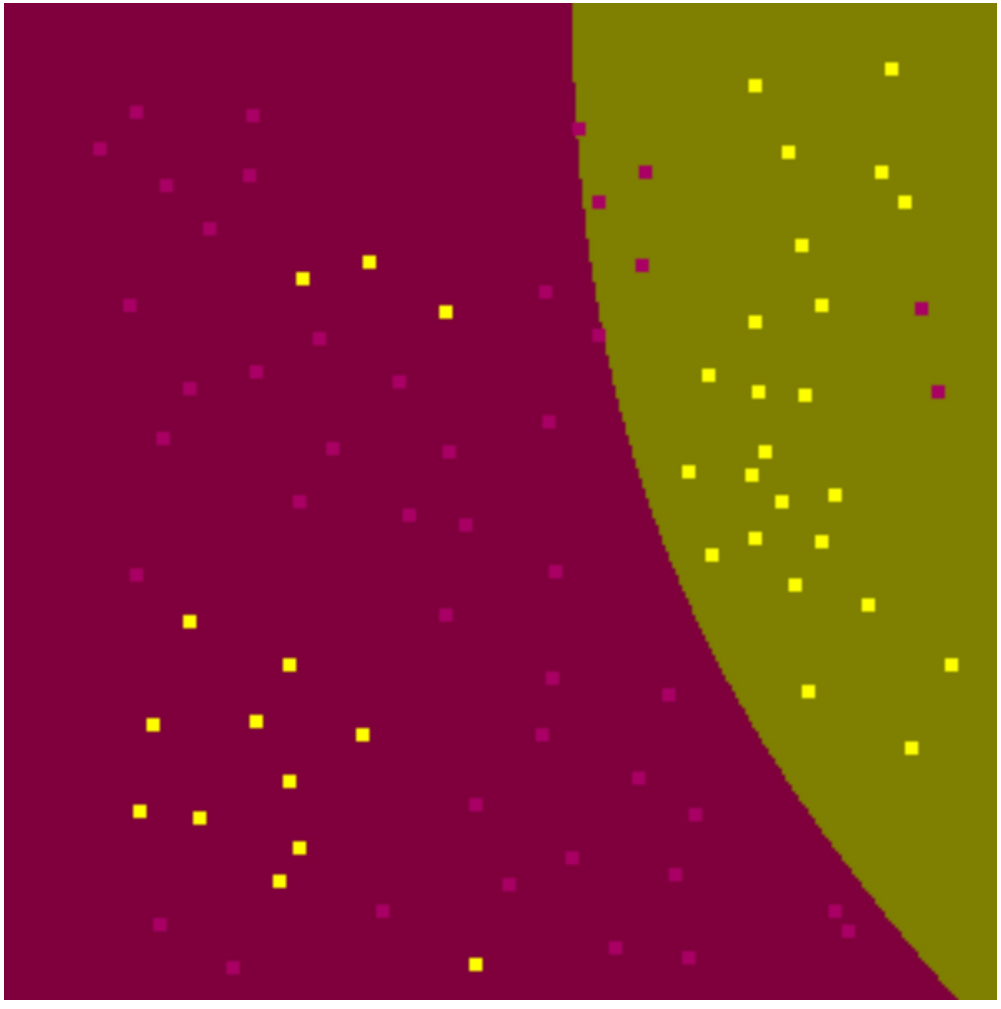


Effect of model parameters for Gaussian kernels



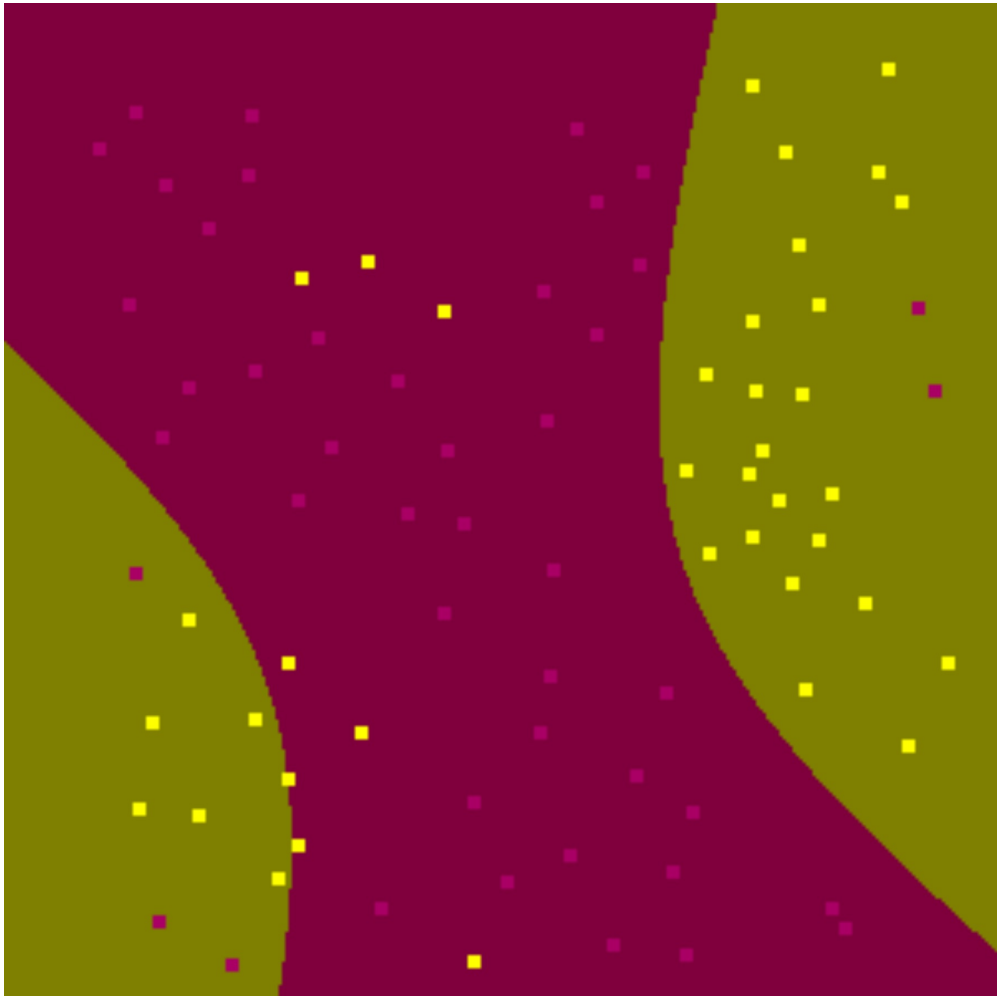
$\sigma^2 = 10, C = 1$

Effect of model parameters for Gaussian kernels



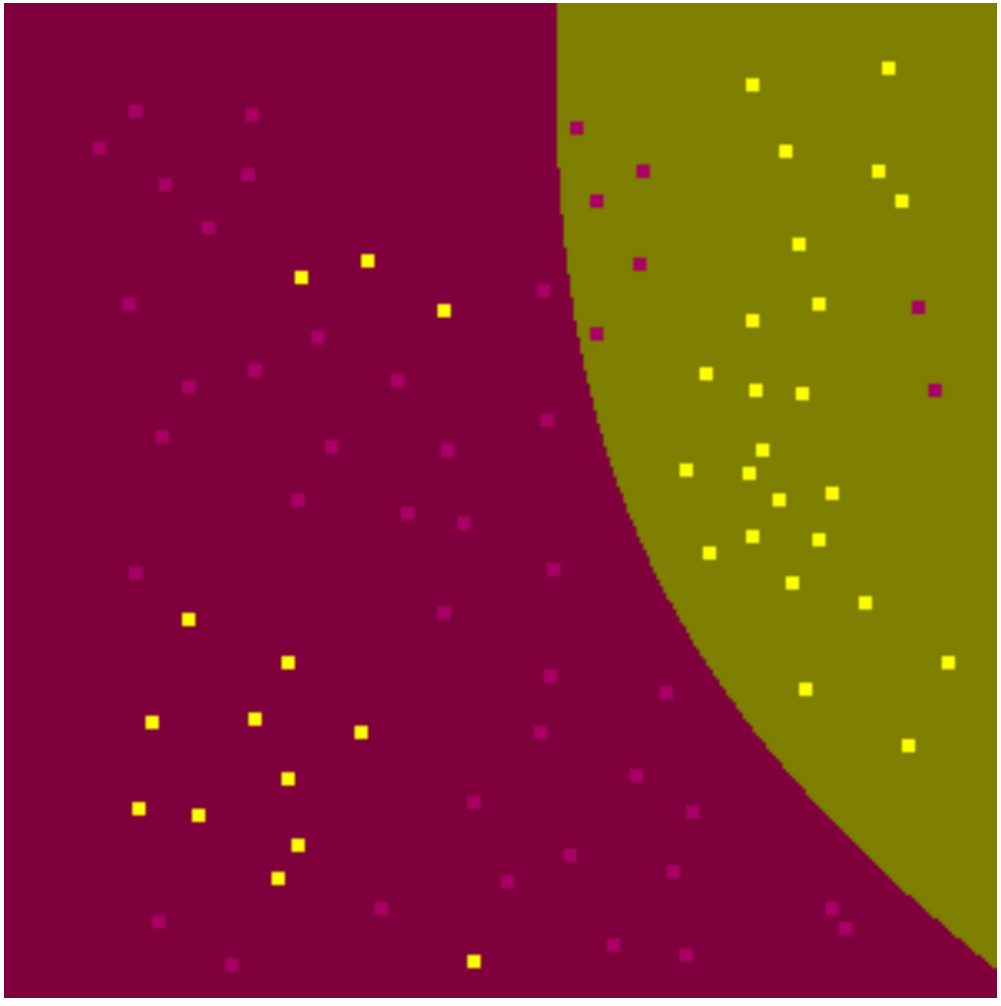
$\sigma^2 = 10, C = 100$

Effect of model parameters for Gaussian kernels



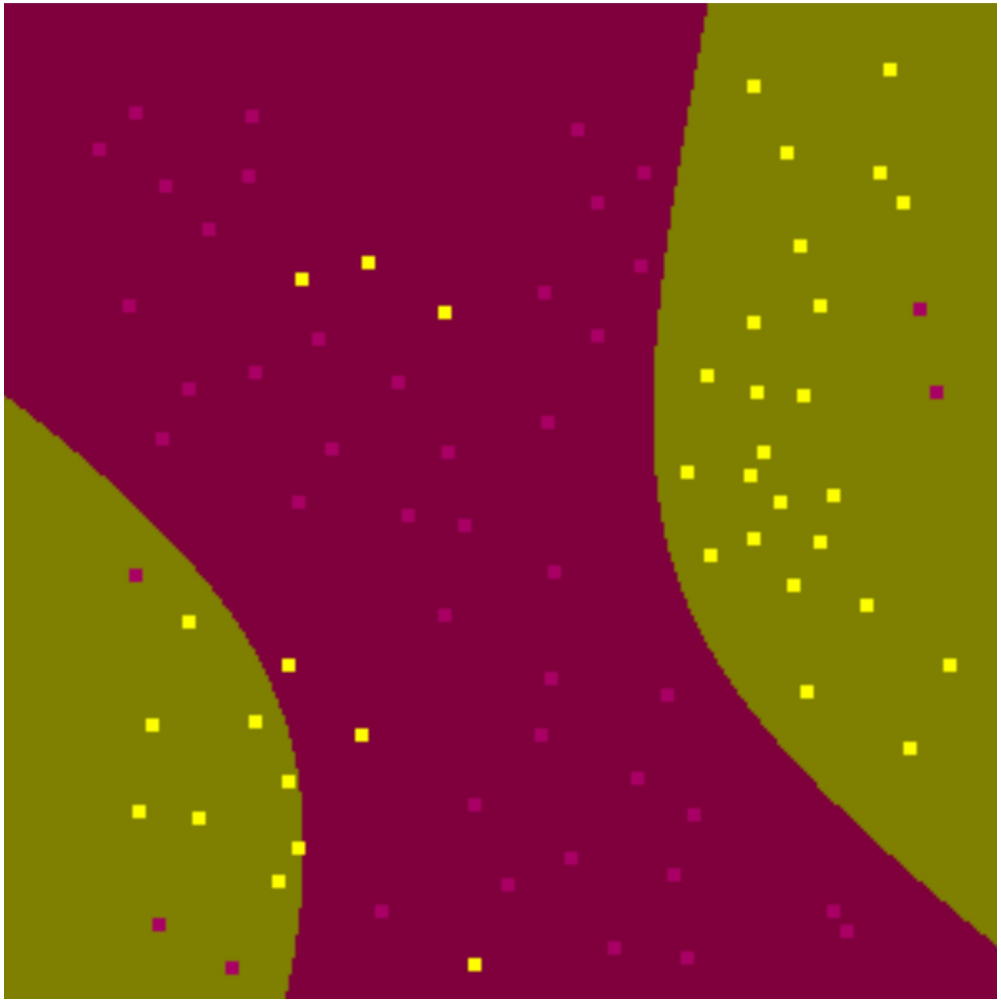
$\sigma^2 = 10, C = 10000$

Effect of model parameters for Gaussian kernels



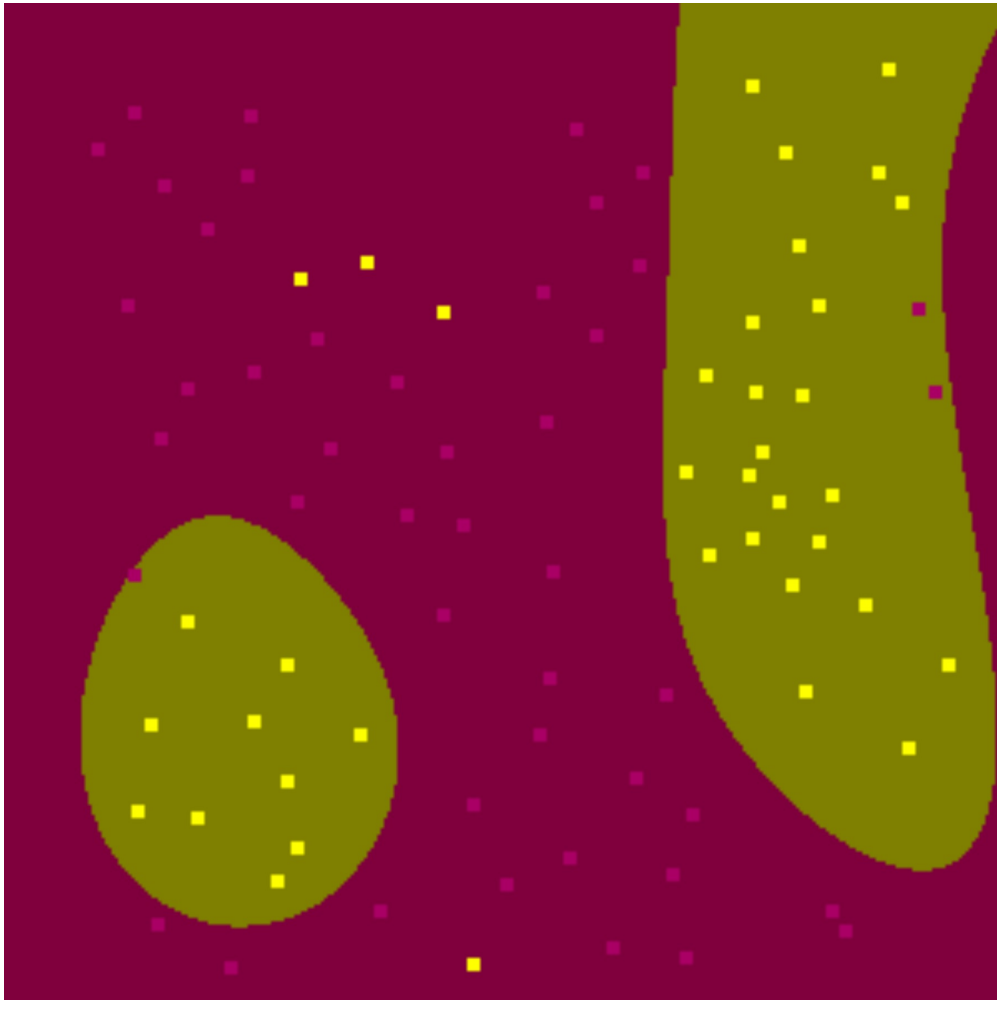
$\sigma^2 = 1, C = 1$

Effect of model parameters for Gaussian kernels



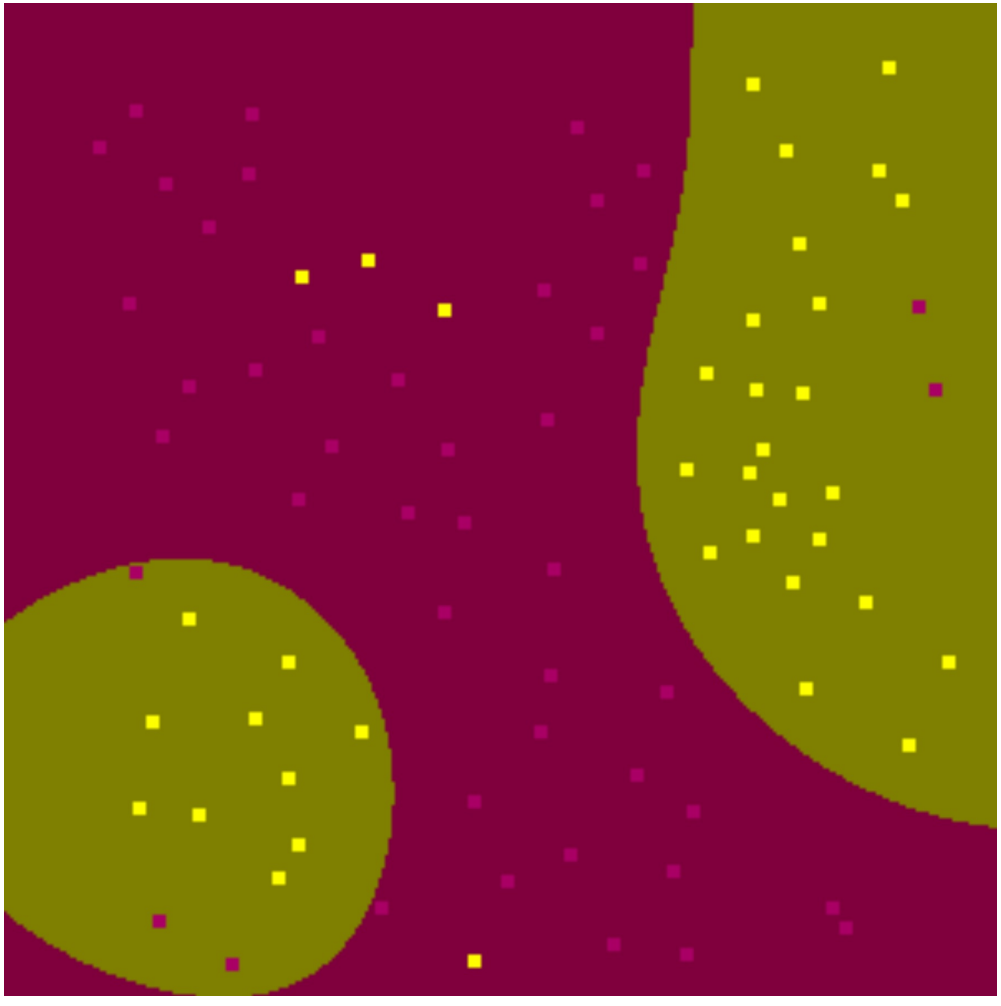
$\sigma^2 = 1, C = 100$

Effect of model parameters for Gaussian kernels



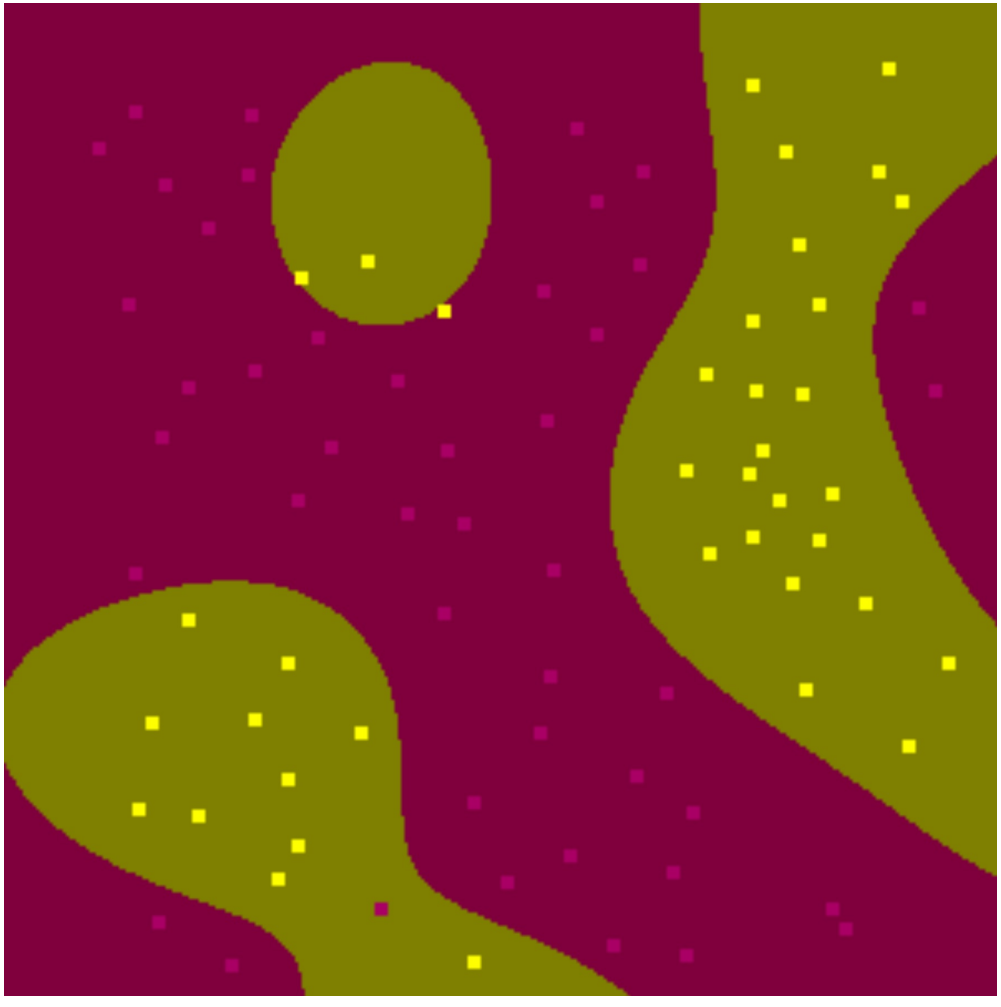
$\sigma^2 = 1, C = 10000$

Effect of model parameters for Gaussian kernels



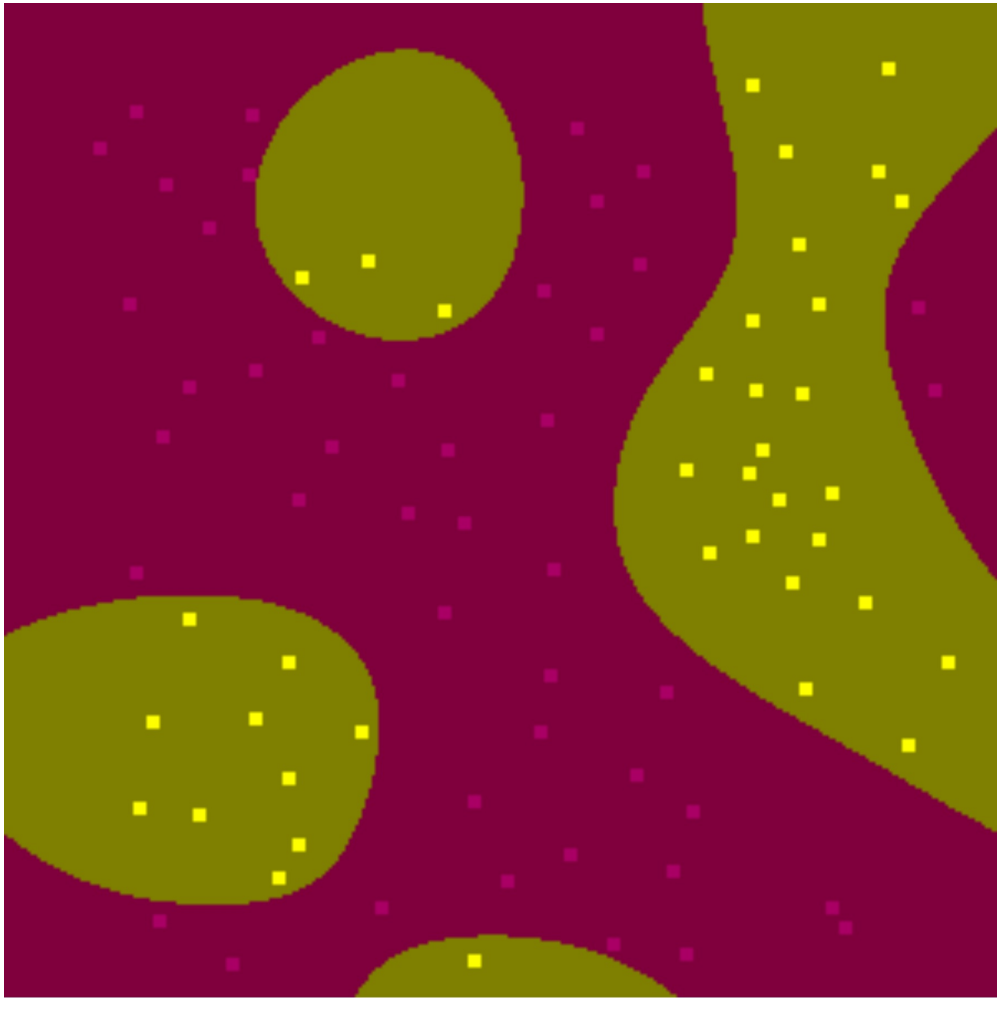
$\sigma^2 = 0.1, C = 1$

Effect of model parameters for Gaussian kernels



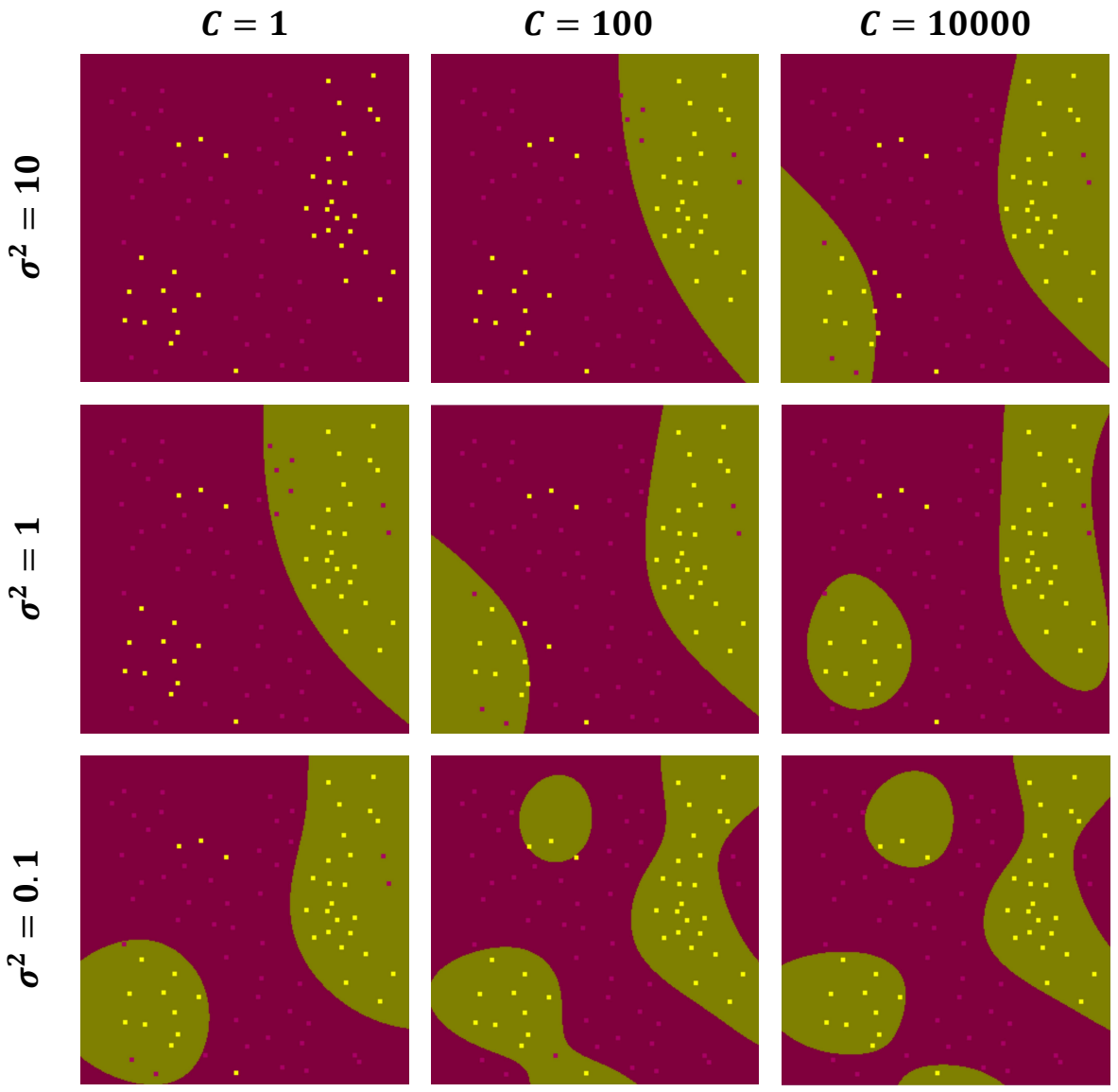
$\sigma^2 = 0.1, C = 100$

Effect of model parameters for Gaussian kernels



$\sigma^2 = 0.1, C = 10000$

Effect of model parameters



Gaussian kernels with different parameters.

Support Vector Regression

Support vector regression (SVR)

- Linear regression aims to minimize the mean-squared error

$$\min_{w,b} \sum_{i=1}^{m} (w^\top x_i + b - y_i)^2$$

- Another option would be to minimize the absolute error

$$\min_{w,b} \sum_{i=1}^{m} |w^\top x_i + b - y_i|$$

- This is more robust to outliers than the squared loss

- But we cannot require that all points be approximated correctly (overfitting!)

Loss function for SVR

- Intuition: we should allow some errors (as long as they are not large) so that the algorithm would be more robust to noise – we only care about the fitting error when it is larger than ϵ

Loss function for SVR

- Intuition: we should allow some errors (as long as they are not large) so that the algorithm would be more robust to noise – we only care about the fitting error when it is larger than ϵ
- Analogous to the large margin intuition in SVM (i.e., only support vectors matter)

Loss function for SVR

- Intuition: we should allow some errors (as long as they are not large) so that the algorithm would be more robust to noise – we only care about the fitting error when it is larger than ϵ
- Analogous to the large margin intuition in SVM (i.e., only support vectors matter)
- We introduce the ϵ-insensitive loss:

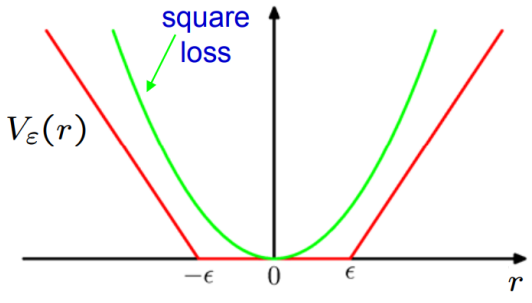
$$J_\epsilon = \sum_{i=1}^{m} V_\epsilon(x_i, y_i),$$

where

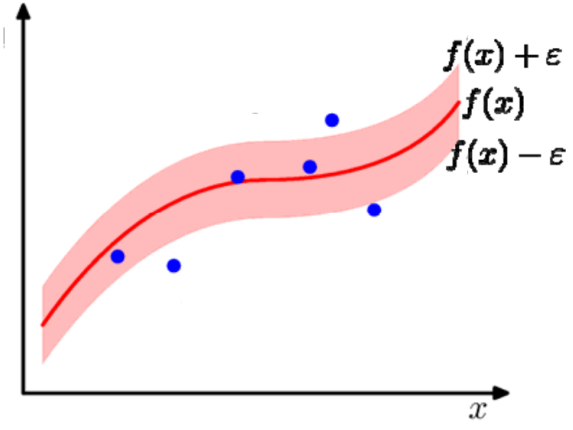
$$V_\epsilon(x_i, y_i) = \begin{cases} 0, & \text{if } |w^\top x_i + b - y_i| \leq \epsilon \\ |w^\top x_i + b - y_i| - \epsilon, & \text{otherwise} \end{cases}$$

ϵ-insensitive loss

$$V_\epsilon(x_i, y_i) = \begin{cases} 0, & \text{if } |w^\top x_i + b - y_i| \leq \epsilon \\ |w^\top x_i + b - y_i| - \epsilon, & \text{otherwise} \end{cases}$$



cost is zero inside epsilon "tube"



Effect of ϵ

As ϵ increases, the function is allowed to move away from the data points, the number of support vectors decreases, the learning curve becomes smoother, and the fit on the training set gets worse.

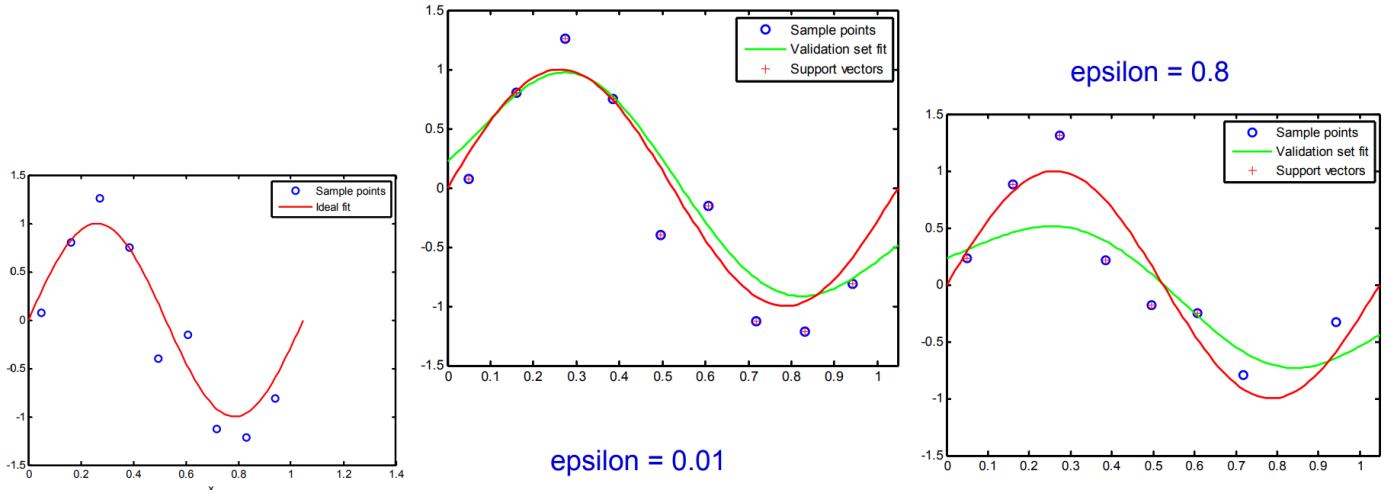


Figure credit: Andrew Zisserman

Summary

What you should know

From this lecture

- Large margin intuition and formulation for SVMs.
- Use of Lagrange multipliers to transform optimization problems.
- Primal and dual optimization problems for SVMs.
- Kernel tricks for nonlinear SVMs.
- SVMs for non-separable cases.
- Support vector regression (SVR).

SVMs Summary

- Advantages
  - Maximize margin – regularize the model complexity and less sensitive to outliers.
  - Work well on high-dimensional small data sets.
  - Use kernel tricks – produce nonlinear classifier and impose prior knowledge.
  - Have nice theoretical properties.

SVMs Summary

- Advantages
  - Maximize margin – regularize the model complexity and less sensitive to outliers.
  - Work well on high-dimensional small data sets.
  - Use kernel tricks – produce nonlinear classifier and impose prior knowledge.
  - Have nice theoretical properties.
- Disadvantages
  - Computationally expensive for large-size data sets.
  - Choosing a good kernel function and model parameter is not easy.