Elements of Machine Learning & Data Science

Winter semester 2025/26

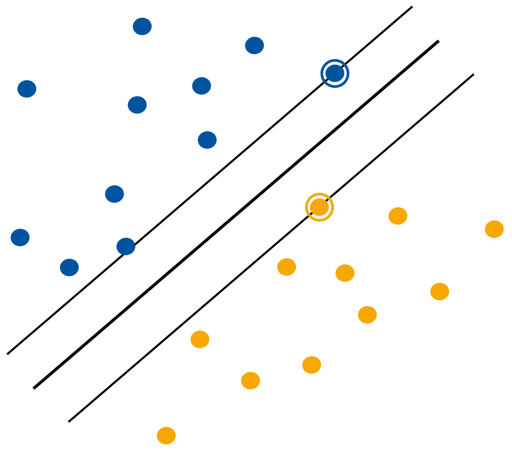
Lecture 14 – Support Vector Machines I

09.12.2025

Prof. Bastian Leibe

Machine Learning Topics

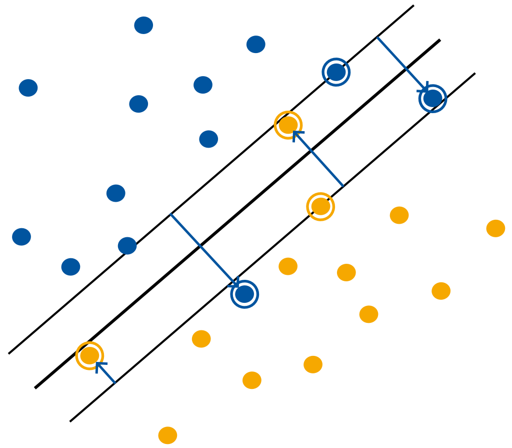
8. Introduction to ML
9. Probability Density Estimation
10. Linear Discriminants
11. Linear Regression
12. Logistic Regression
13. **Support Vector Machines**
14. Neural Network Basics



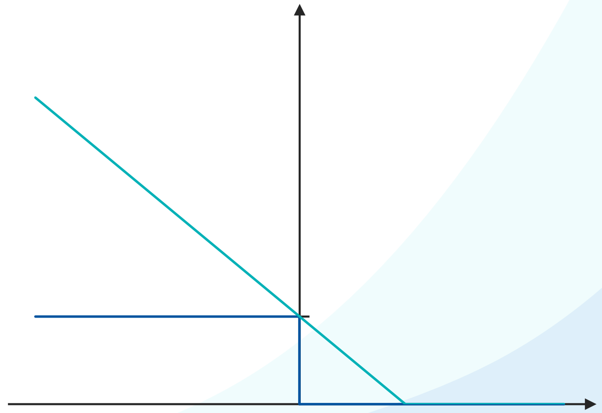
Maximum Margin Classification

$$L_p(\mathbf{w}, b, \mathbf{a})$$
$$L_d(\mathbf{a})$$

Primal & Dual Form



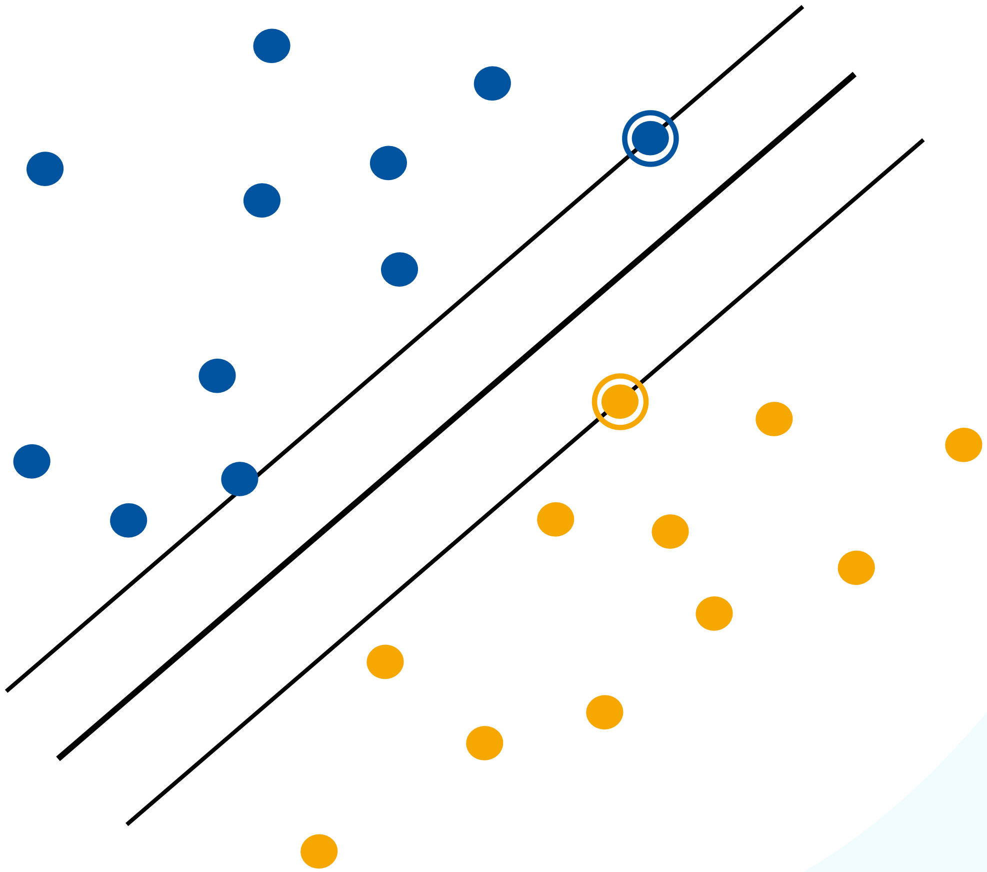
Soft-Margin SVM



Hinge Loss

Support Vector Machines

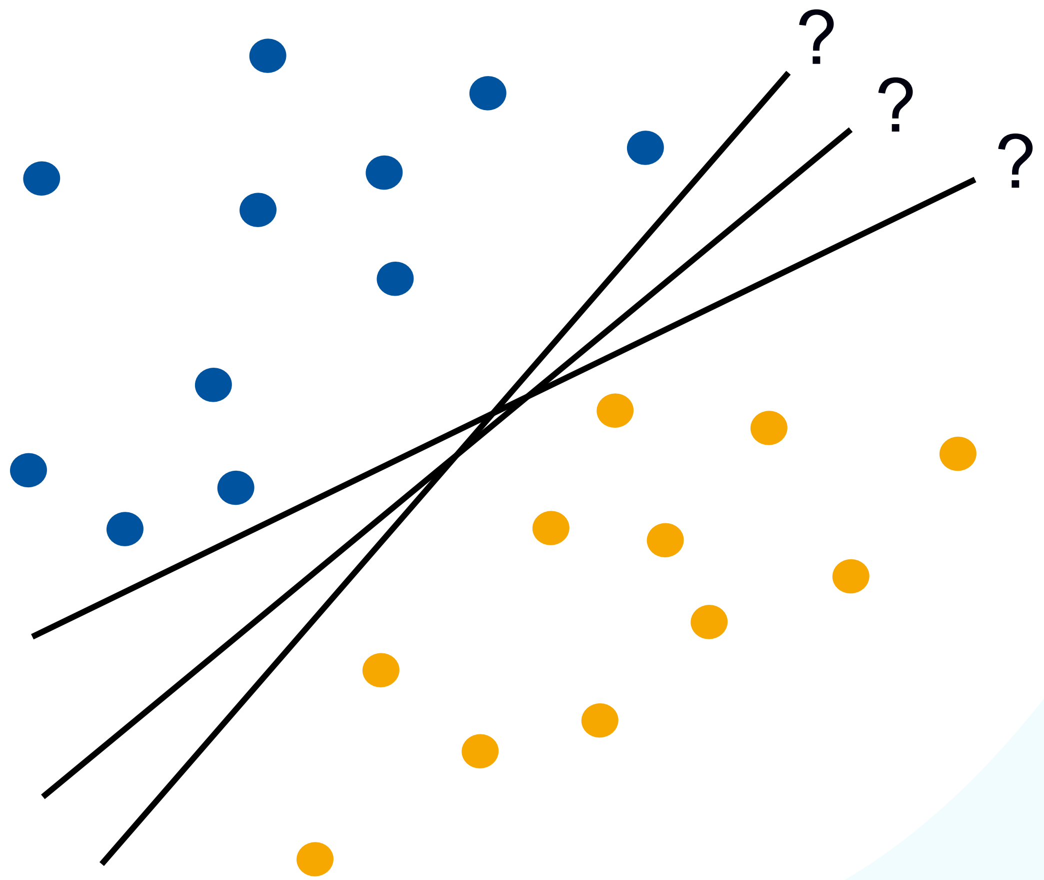
1. **Maximum Margin Classification**
2. Primal Formulation
3. Dual Formulation
4. Soft-Margin SVMs
5. Non-linear SVMs
6. Error Function Analysis



Maximum Margin Classification

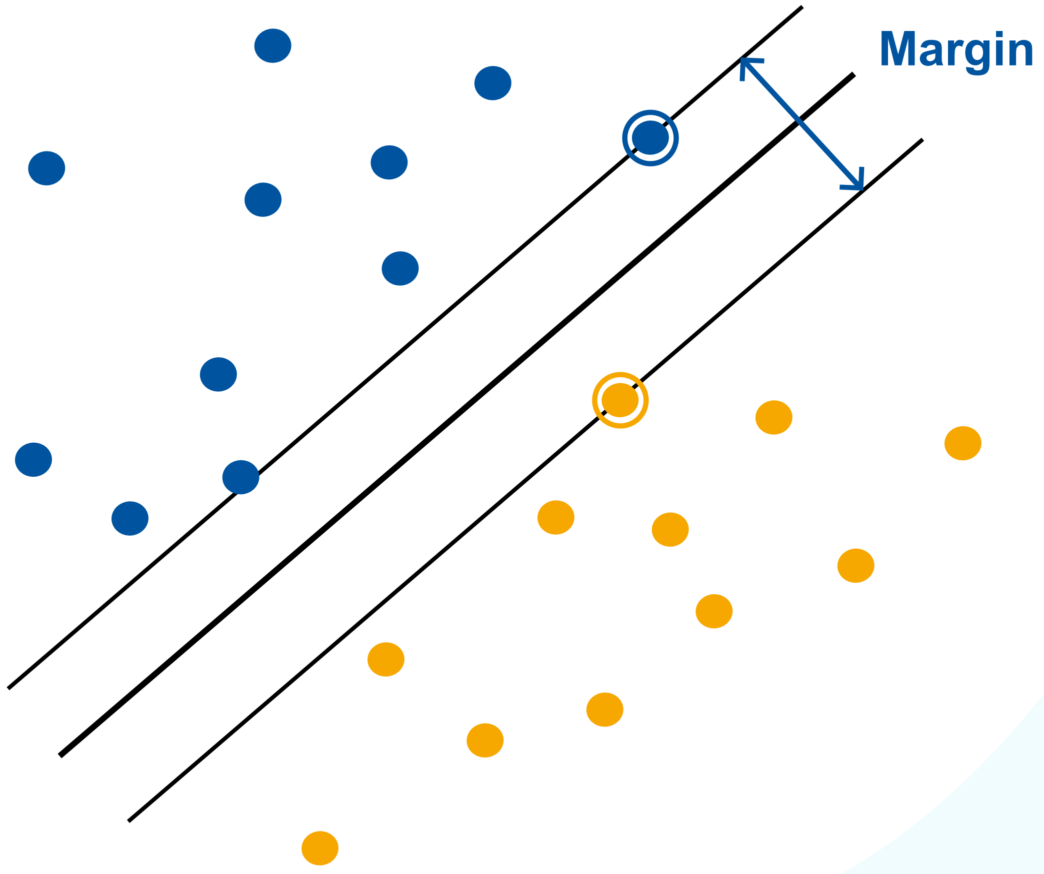
- Overfitting is often a problem with linearly separable data
  - Which of the many possible decision boundaries is correct?
  - All of them have zero error on the training set…
  - However, they will perform differently on novel test data.

- *How can we select the classifier with the best generalization performance?*



Maximum Margin Classification

- Intuitively, we want to choose the classifier which leaves maximal "safety room" for future data points.
- This classifier has the largest margin between positive and negative points.
- It can be shown: The larger the margin, the lower the capacity for overfitting.



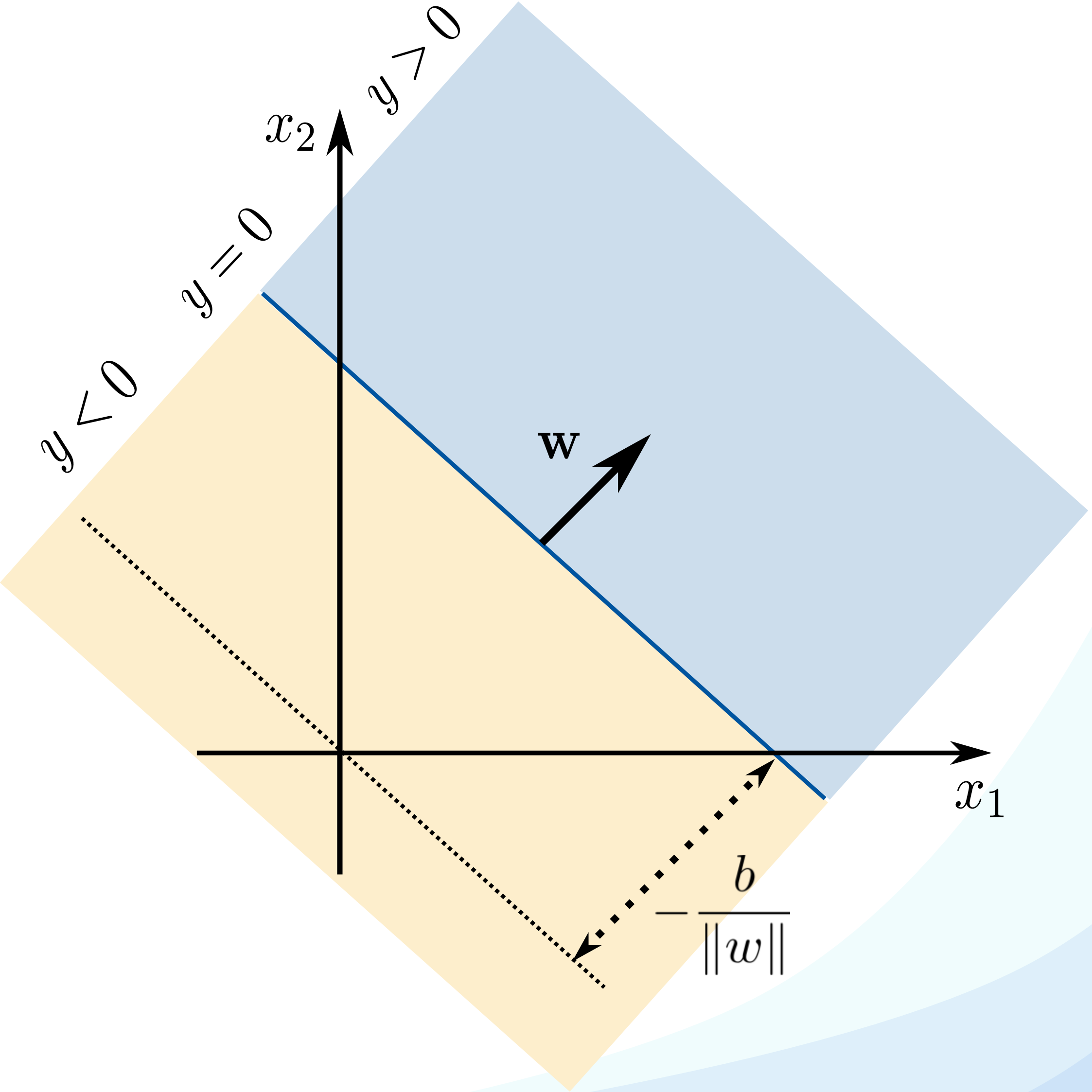
Intuition

- Let's first consider linearly separable data:
  - N training data points $\{(\mathbf{x}_i, t_i)\}_{i=1}^{N}, \;\; \mathbf{x}_i \in \mathbb{R}^D$
  - Binary labels $\quad t_i \in \{-1, 1\}$
- A linear discriminant function models a hyperplane separating the data:

$$y(\mathbf{x}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} + b$$

- *Note that we denote the bias explicitly with b.*

- Decision rule
  - Decide for $\mathcal{C}_1$ if $y(\mathbf{x}) > 0$, else for $\mathcal{C}_2$.



Support Vector Machines

- Assuming linearly separable data, we can always find a hyperplane with

$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \geq +1 \text{ for } t_n = +1$$
$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \leq -1 \text{ for } t_n = -1$$

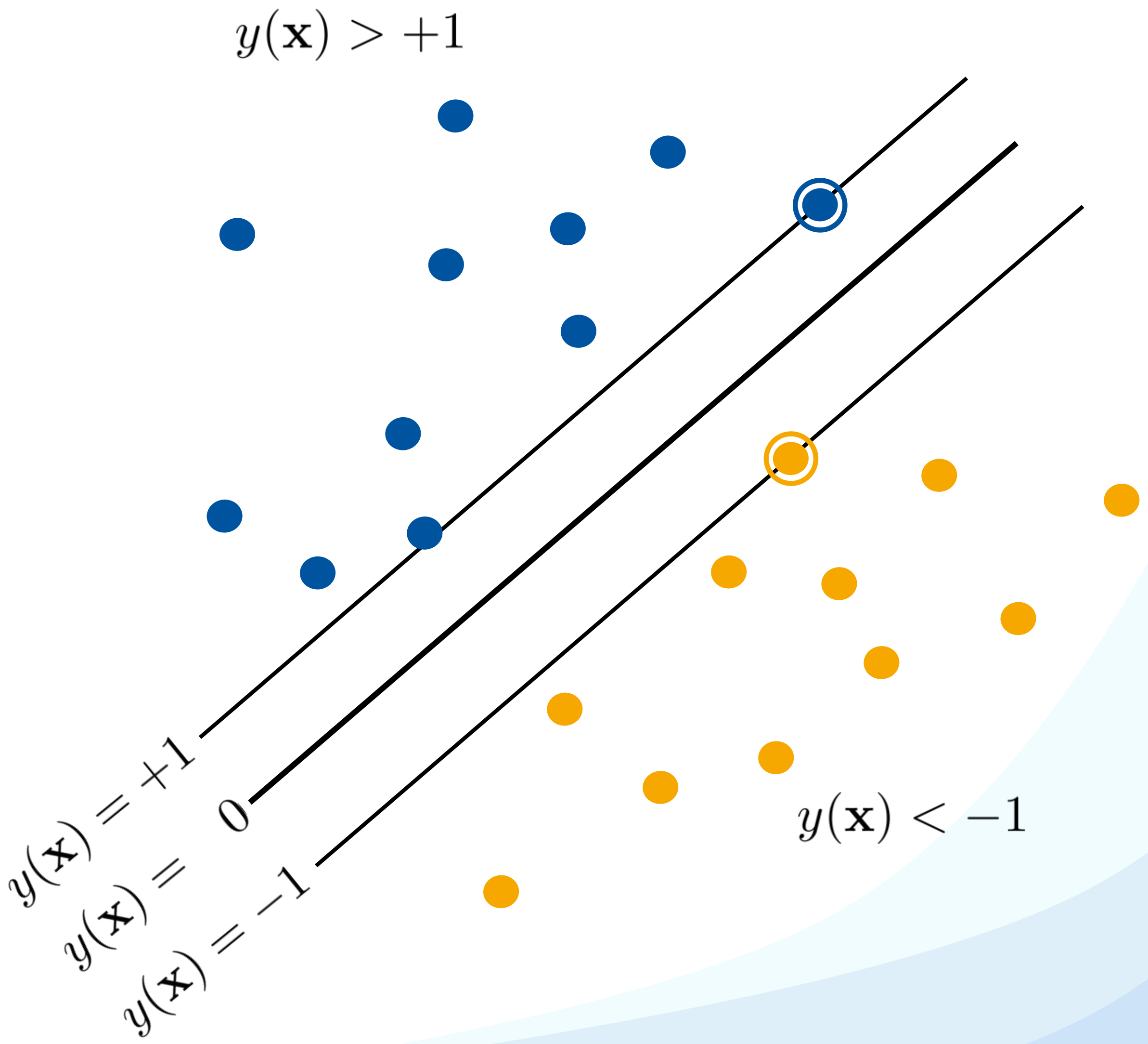
- In short:

$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

- We can rescale $\mathbf{w}$ such that the equation holds exactly for the points on the margin:

$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) = 1$$

- There will be at least one such point.



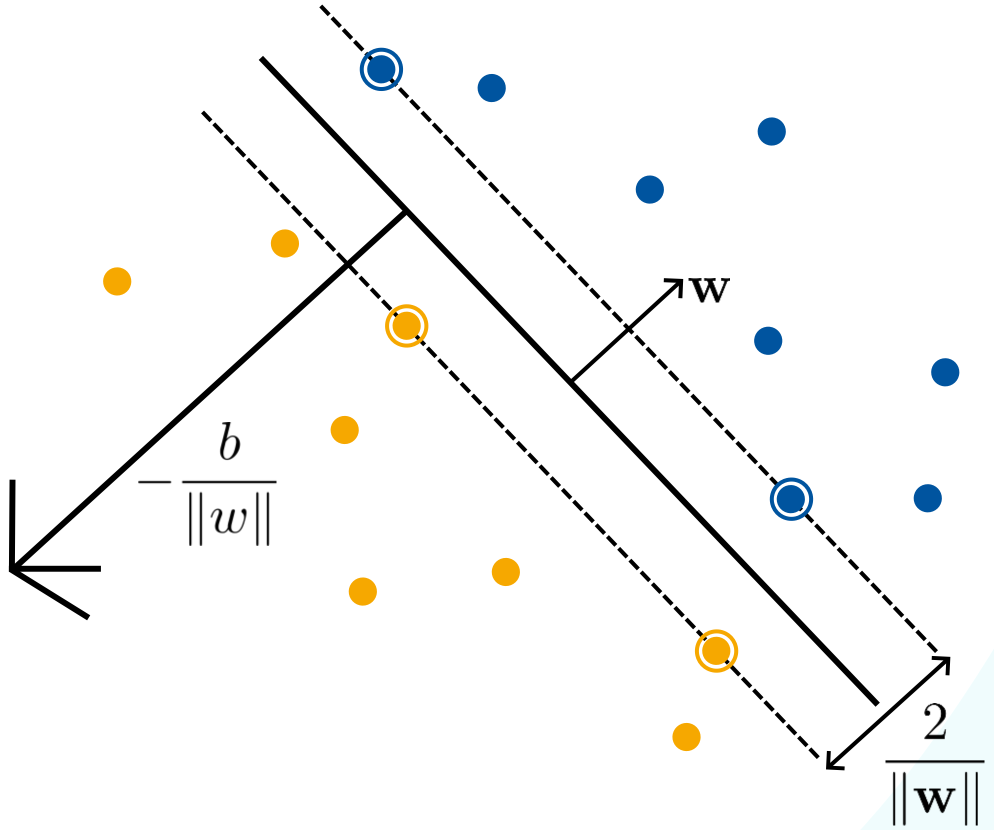
- We can choose $\mathbf{w}$ such that

$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b = +1 \text{ for one } t_n = +1$$
$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b = -1 \text{ for one } t_n = -1$$

- The distance between those hyperplanes is then the margin:

$$d_- = d_+ = \frac{1}{\|\mathbf{w}\|}$$
$$d_- + d_+ = \frac{2}{\|\mathbf{w}\|}$$

$\Rightarrow$ Maximize the margin by minimizing $\|\mathbf{w}\|^2$



- Optimization problem
  - Find the hyperplane with maximum margin by optimizing:

$$\underset{\mathbf{w},b}{\arg\min}\ \frac{1}{2}\|\mathbf{w}\|^2$$

"Maximize the margin"

such that

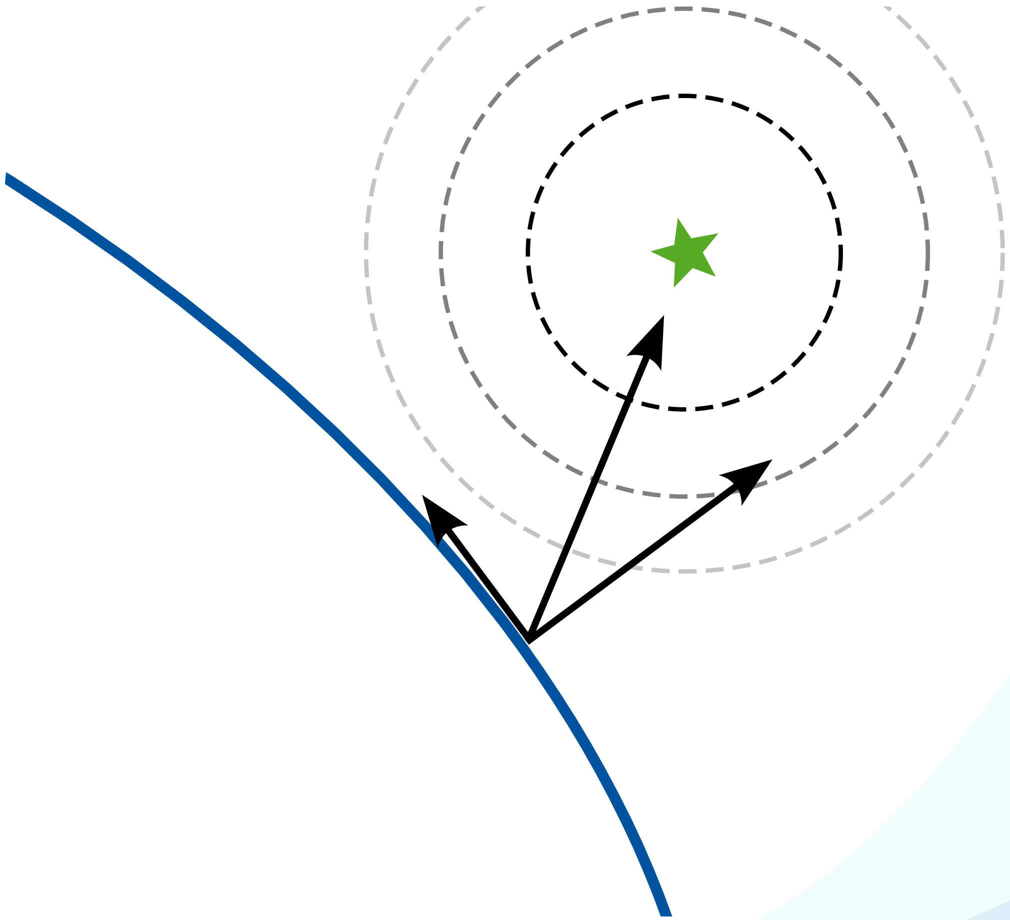
$$t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

"such that each point is on the correct side of the margin"

- This is a quadratic programming problem with linear constraints.

Support Vector Machines

1. Maximum Margin Classification
    a) **Constrained Optimization**
2. Primal Formulation
3. Dual Formulation
4. Soft-Margin SVMs
5. Non-linear SVMs
6. Error Function Analysis



Constrained Optimization

- Recall the SVM objective:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{such that} \quad t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

- This is a constrained optimization problem.
  - We want to optimize an objective $K(\mathbf{x})$ subject to constraints $f(\mathbf{x})$:

$$\operatorname*{opt}_{\mathbf{x}} K(\mathbf{x}) \qquad \text{min } \textit{or} \text{ max}$$

$$\text{such that} \quad f(\mathbf{x}) = 0 \qquad \textit{equality constraints}$$

$$f(\mathbf{x}) \geq 0 \qquad \textit{inequality constraints}$$

SVM

$$K(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$$

$$f_n(\mathbf{w}) = t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1 \geq 0 \quad \forall n$$

- We can solve such constrained optimization problems using the technique of Lagrange multipliers.

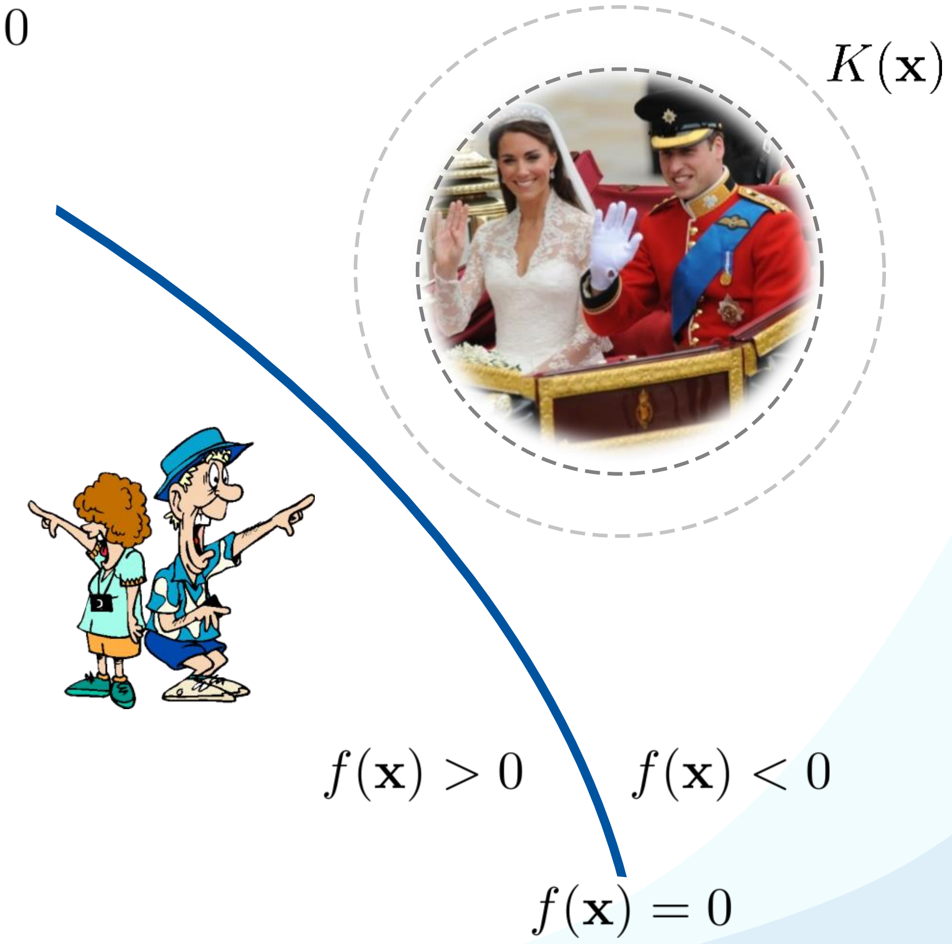
Lagrange Multipliers

- We want to maximize $K(\mathbf{x})$

$K(\mathbf{x})$

Lagrange Multipliers

- We want to maximize $K(\mathbf{x})$ subject to constraints $f(\mathbf{x}) = 0$



Lagrange Multipliers

- We want to maximize $K(\mathbf{x})$ subject to constraints $f(\mathbf{x}) = 0$
- We can only move along $\nabla_{||}K = \nabla K + \lambda \nabla f$, with $\lambda \neq 0$.
- Add the constraints to the objective by introducing auxiliary variables λ:

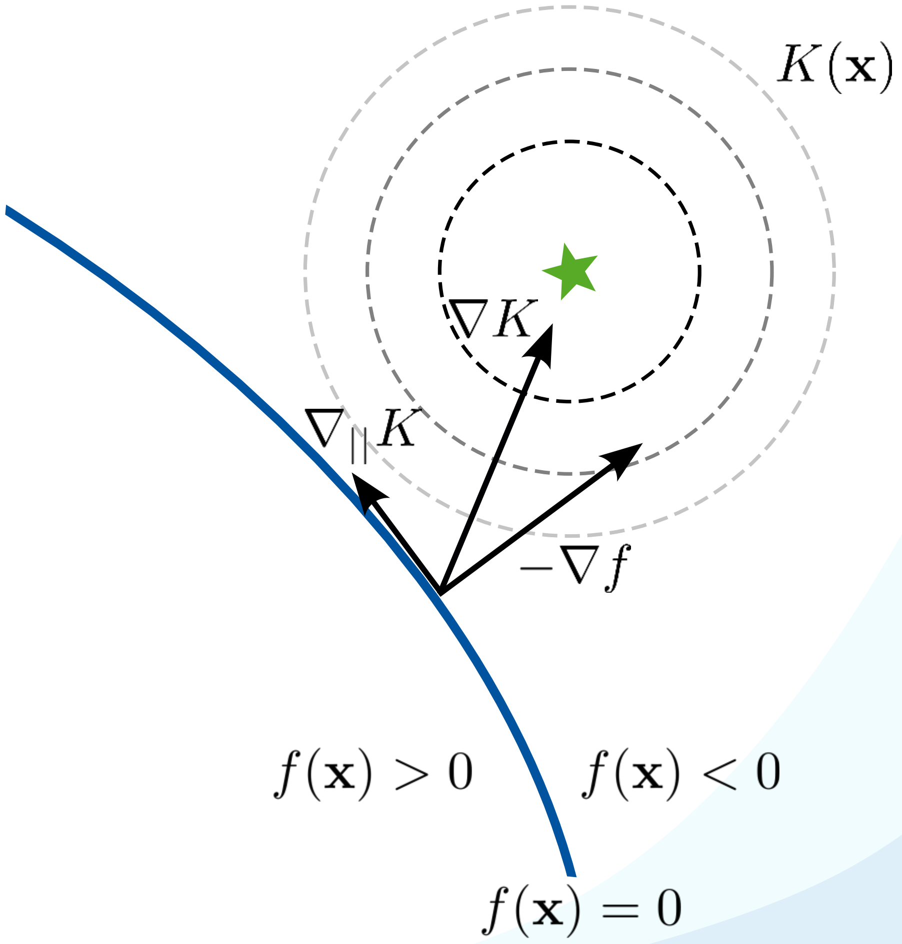
$$\mathcal{L}(\mathbf{x}, \lambda) = K(\mathbf{x}) + \lambda f(\mathbf{x})$$

- $\mathcal{L}$ is called the Lagrangian form of the optimization problem, and λ is referred to as a Lagrange multiplier.
- Optimize $\mathcal{L}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \nabla_{||}K \stackrel{!}{=} 0$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = f(\mathbf{x}) \stackrel{!}{=} 0$$

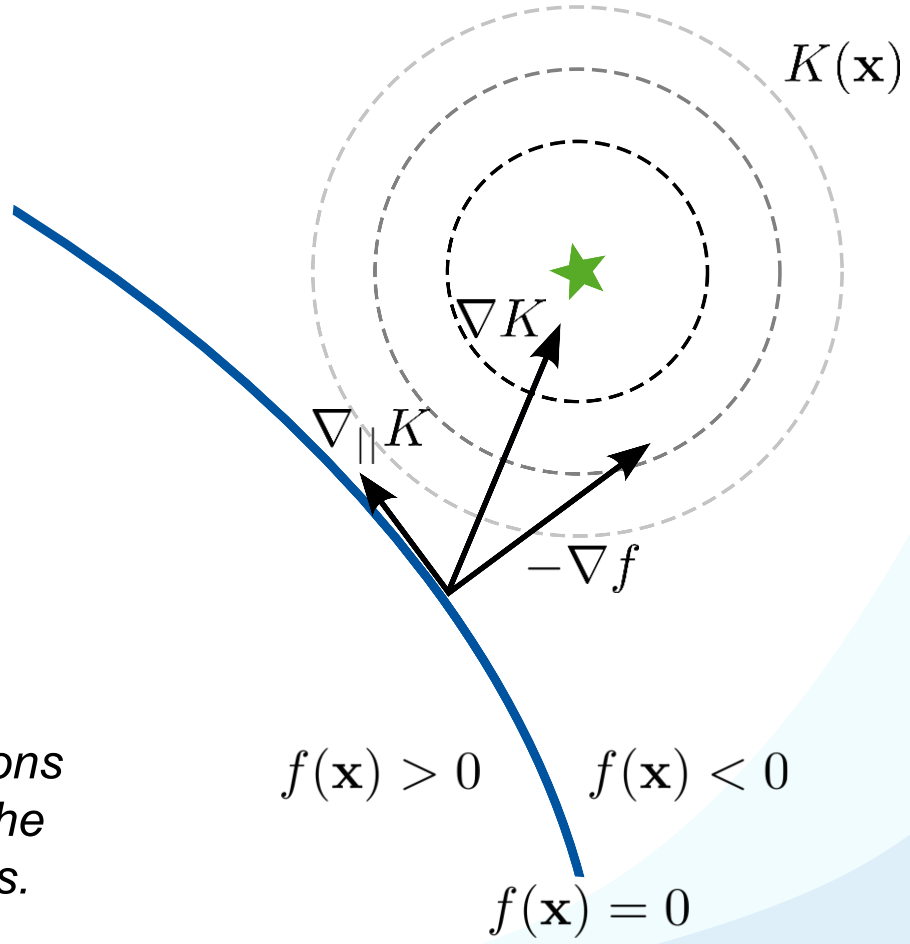
The objective is maximized while satisfying the constraints.



Inequality Constraints

- Now let's use inequality constraints $f(\mathbf{x}) \geq 0$.
- Optimize $\mathcal{L}(\mathbf{x}, \lambda) = K(\mathbf{x}) + \lambda f(\mathbf{x})$
  - Two cases
    - Solution lies on boundary:
      $\Rightarrow f(\mathbf{x}) = 0$ for some $\lambda > 0$
    - Solution lies inside $f(\mathbf{x}) > 0$:
      $\Rightarrow$ Constraint inactive: $\lambda = 0$

In both cases:
$\lambda f(\mathbf{x}) = 0$



- Karush-Kuhn-Tucker (KKT) conditions:

$$\lambda \geq 0$$
$$f(\mathbf{x}) \geq 0$$
$$\lambda f(\mathbf{x}) = 0$$

All valid solutions need to fulfill the KKT conditions.

Maximization vs. Minimization

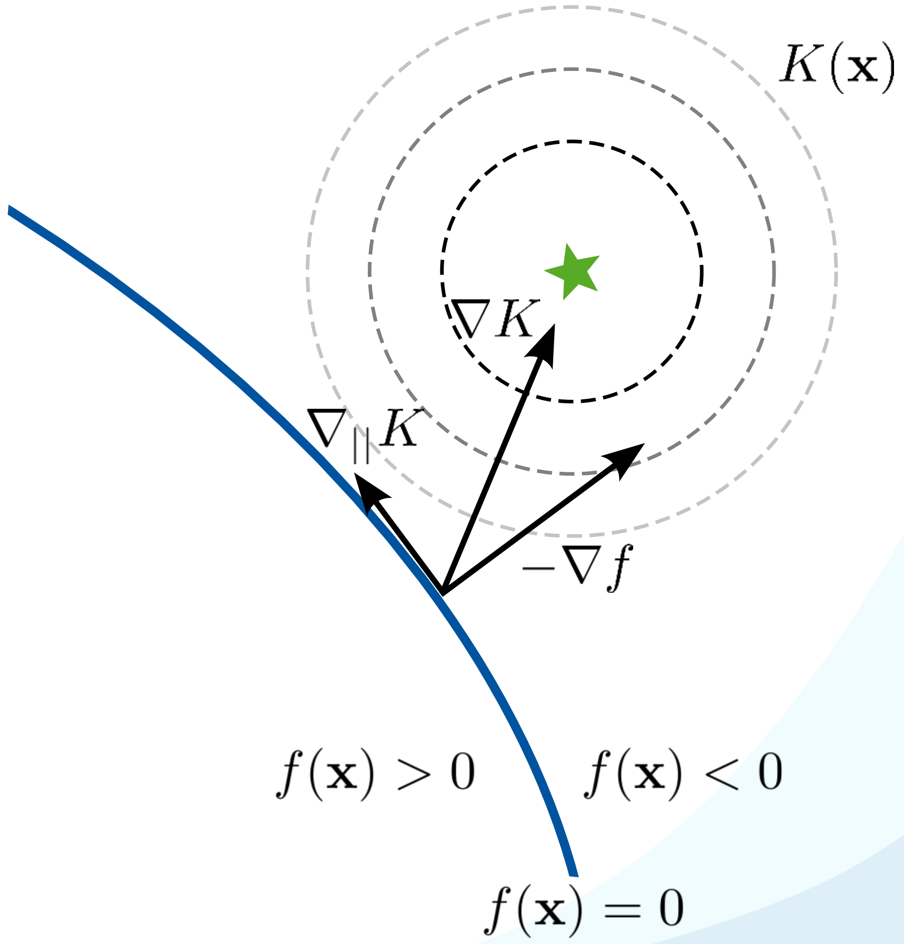
- Note: differences for maximization vs. minimization.
- If we want to maximize $K(\mathbf{x})$ subject to $f(\mathbf{x}) \geq 0$, we optimize the Lagrangian form

  $$\mathcal{L}(\mathbf{x}, \lambda) = K(\mathbf{x}) + \lambda f(\mathbf{x})$$

  - *maximize* w.r.t. $\mathbf{x}$
  - *minimize* w.r.t. λ
- If we want to minimize $K(\mathbf{x})$ subject to $f(\mathbf{x}) \geq 0$, we optimize the Lagrangian form

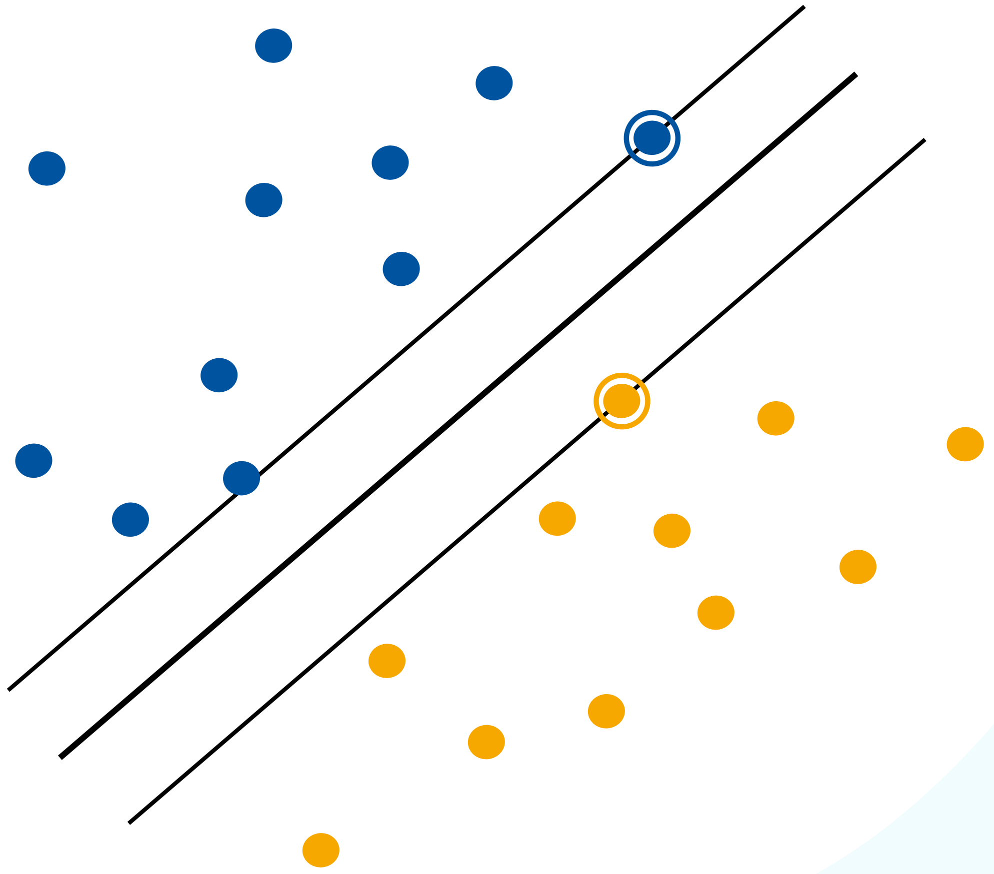
  $$\mathcal{L}(\mathbf{x}, \lambda) = K(\mathbf{x}) - \lambda f(\mathbf{x})$$

  - *minimize* w.r.t. $\mathbf{x}$
  - *maximize* w.r.t. λ



Support Vector Machines

1. Maximum Margin Classification
2. **Primal Formulation**
3. Dual Formulation
4. Soft-Margin SVMs
5. Non-linear SVMs
6. Error Function Analysis



Primal SVM Formulation

- Recall the SVM objective:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{such that} \quad t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

- We introduce positive Lagrange multipliers $a_n \geq 0$ and get the primal form of SVMs:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left[t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1\right]$$

Necessary and sufficient conditions:

$$\begin{aligned} a_n &\geq 0 \\ t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1 &\geq 0 \\ a_n[t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1] &= 0 \end{aligned}$$

KKT conditions:

$$\begin{aligned} \lambda &\geq 0 \\ f(\mathbf{x}) &\geq 0 \\ \lambda f(\mathbf{x}) &= 0 \end{aligned}$$

Lagrangian Formulation

- We want to minimize the primal form:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left[t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1\right]$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{a})}{\partial b} = \sum_{n=1}^{N} a_n t_n \qquad \frac{\partial L(\mathbf{w}, b, \mathbf{a})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

- Setting the gradients for $\mathbf{w}, b$ to zero, we get:

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \boxed{\sum_{n=1}^{N} a_n t_n = 0} \qquad \frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \boxed{\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n}$$

- The hyperplane is computed as a linear combination of training examples:

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

- Additionally, the solution needs to fulfill

$$a_n \left[t_n (\mathbf{w}^\mathsf{T} \mathbf{x}_n + b) - 1 \right] = 0$$

- This implies $a_n > 0$ only for those points for which

$$\left[t_n (\mathbf{w}^\mathsf{T} \mathbf{x}_n + b) - 1 \right] = 0$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \boxed{\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n}$$

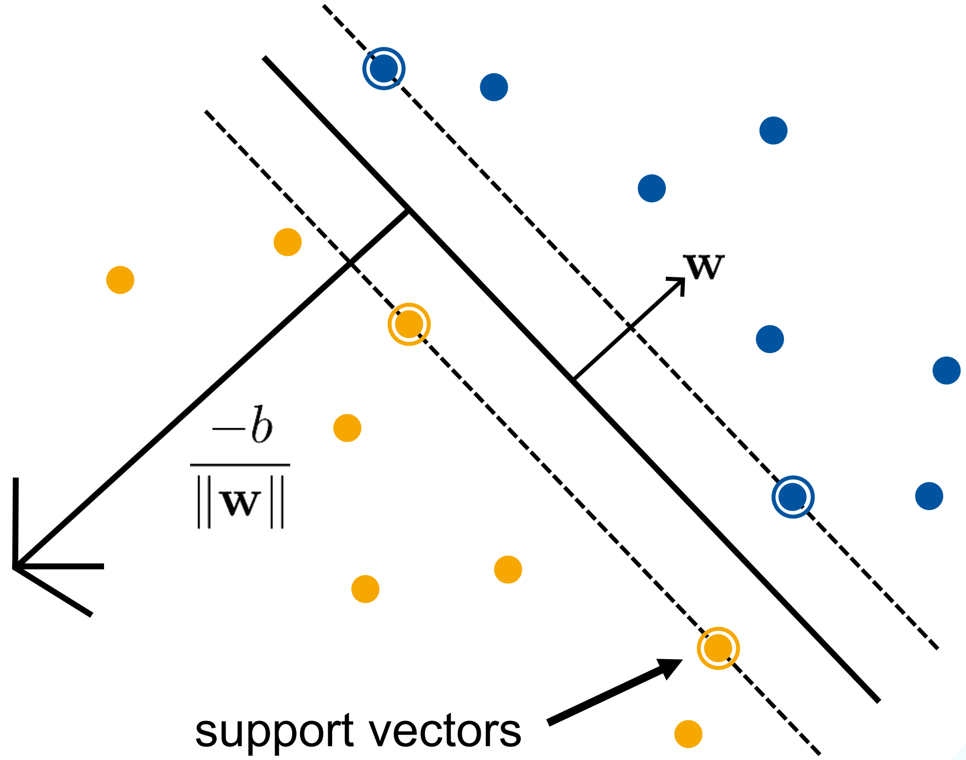
KKT conditions:

$$a_n \geq 0$$
$$t_n (\mathbf{w}^\mathsf{T} \mathbf{x}_n + b) - 1 \geq 0$$
$$a_n [t_n (\mathbf{w}^\mathsf{T} \mathbf{x}_n + b) - 1] = 0$$

Only some data points influence the decision boundary!

Intuition

- The training points with $a_n > 0$ are called support vectors.
- They are the points on the margin.
- This makes the SVM robust to "too correct" points!



- We still need to find b.
- Observation: Any support vector $\mathbf{x}_n$ satisfies

$$t_n y(\mathbf{x}_n) = t_n \left(\sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^\mathsf{T} \mathbf{x}_n + b \right) = 1$$

- Using $t_n^2 = 1$, we can derive

$$b = t_n - \sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^\mathsf{T} \mathbf{x}_n$$

- In practice, it is more robust to average over all support vectors:

$$\boxed{b = \frac{1}{N_\mathcal{S}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^\mathsf{T} \mathbf{x}_n \right)}$$

Advantages

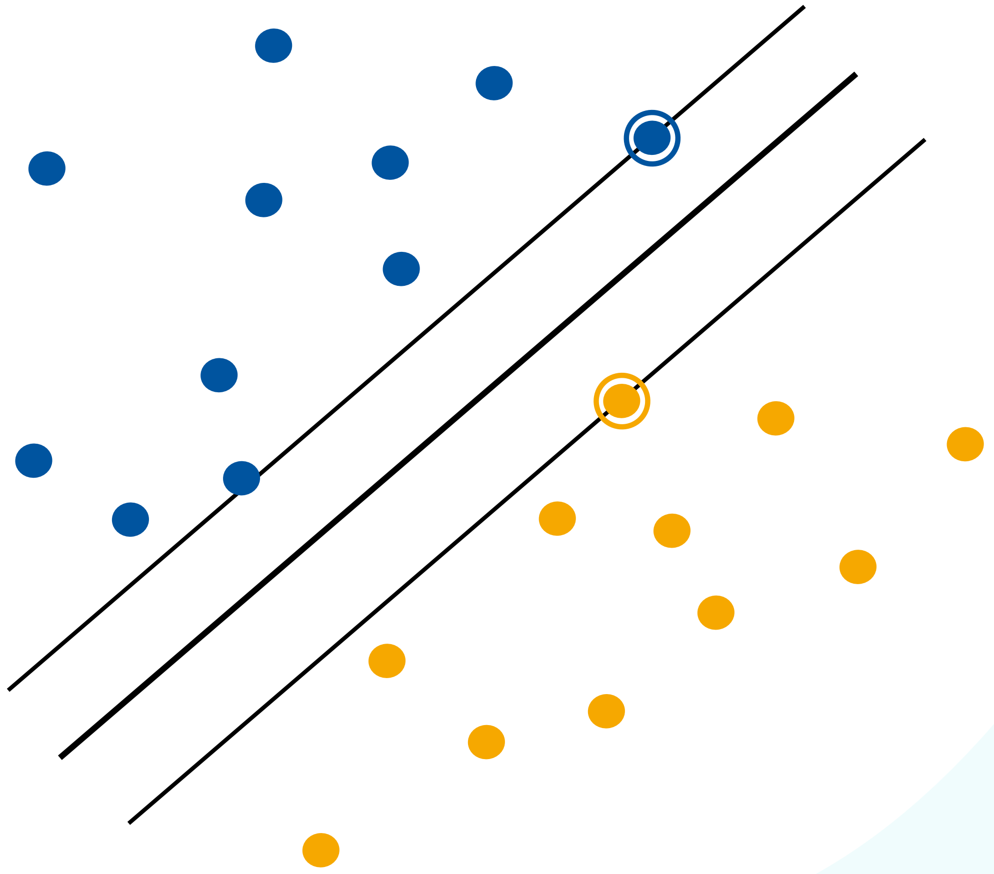
- SVMs yield a linear classifier with "guaranteed" generalization capability.
- Convex optimization, yields globally optimal solution.
- Solution depends only on a subset of the input data points, the support vectors.
- Automatic robustness against "too correct" data points.

Limitations

- Need to solve quadratic programming problem: time complexity for that is cubic in the number of variables.
- Here: Time complexity is in $\mathcal{O}(D^3)$.
- Scaling to high-dimensional data is difficult.

Support Vector Machines

1. Maximum Margin Classification
2. Primal Formulation
3. **Dual Formulation**
4. Soft-Margin SVMs
5. Non-linear SVMs
6. Error Function Analysis



Reminder: Primal SVM Formulation

- SVM objective:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{such that} \quad t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

- This is a Quadratic Programming (QP) problem with linear inequality constraints.
  - In order to solve it, we have derived the Lagrangian primal form

$$L_p(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left[t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1\right]$$

  - We are *minimizing* this objective with respect to $\mathbf{w}$ and b, and *maximizing* with respect to $\mathbf{a}$.

Solving a QP

- SVM objective:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{such that} \quad t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$$

- Solving QPs is a well-understood problem
  - Typically done with the help of a QP solver.
  - Solving a QP in K variables can be done in runtime $\mathcal{O}(K^3)$.

- In our case: $\mathbf{x}, \mathbf{w} \in \mathbb{R}^D$
  - #Variables: $D + 1$

  $\Rightarrow$ Complexity: $\mathcal{O}(D^3)$

Solving a QP

- SVM objective:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{such that} \quad t_n(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_n) + b) \geq 1 \quad \forall n$$

- Solving QPs is a well-understood problem
  - Typically done with the help of a QP solver.
  - Solving a QP in K variables can be done in runtime $\mathcal{O}(K^3)$.

- In our case: $\mathbf{x}, \mathbf{w} \in \mathbb{R}^D$
  - #Variables: $D + 1$

  $\Rightarrow$ Complexity: $\mathcal{O}(D^3)$

- With basis functions: $\boldsymbol{\phi}(\mathbf{x}), \mathbf{w} \in \mathbb{R}^M, \ M \gg D$
  - #Variables: $M + 1$

  $\Rightarrow$ Complexity: $\mathcal{O}(M^3)$

$\Rightarrow$ Curse of dimensionality, the SVM Primal Form does not scale well!

Dual SVM Formulation

- Idea
  - We will re-write the SVM primal form objective:

$$L_p(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left[t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) - 1\right]$$

  in a dual form L_d that does not dependent on the dimensionality of the data.

  - To do this, we will use the results we have derived in the previous section

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \boxed{\sum_{n=1}^{N} a_n t_n = 0} \qquad \frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \boxed{\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n}$$

Dual Form of the SVM Objective

- Maximize

$$L_d(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^{\mathsf{T}} \mathbf{x}_n)$$

under the conditions

$$a_n \geq 0 \quad \forall n$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

- We now have an optimization problem in N variables.

  $\Rightarrow$ Complexity: $\mathcal{O}(N^3)$

For the derivation, please watch the video

Discussion

- What have we gained?
  - Previous complexity was $\mathcal{O}(D^3)$, now it is $\mathcal{O}(N^3)$.
  - *Isn't this much worse for large training sets???*

- However, the dual form has several advantages
  1. SVMs have sparse solutions: $a_n \neq 0$ only for support vectors.
     - This makes very efficient algorithms possible.
     - E.g., Sequential Minimal Optimization (SMO)
     - Effective runtime between $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$.
  2. No dependency on the dimensionality anymore.
     - We can work with high-dimensional feature spaces!

Advantages

- Optimization problem only depends on the Lagrange multipliers a_n resulting in a worst-case runtime complexity of $\mathcal{O}(N^3)$.
- Since SVMs have sparse solutions and only few $a_n \neq 0$, specialized algorithms can solve the dual form very efficiently.
- The complexity of QP optimization no longer depends on the dimensionality of the feature space. This makes it possible to use very high-dimensional feature spaces.

Limitations

- Evaluating the SVM decision function

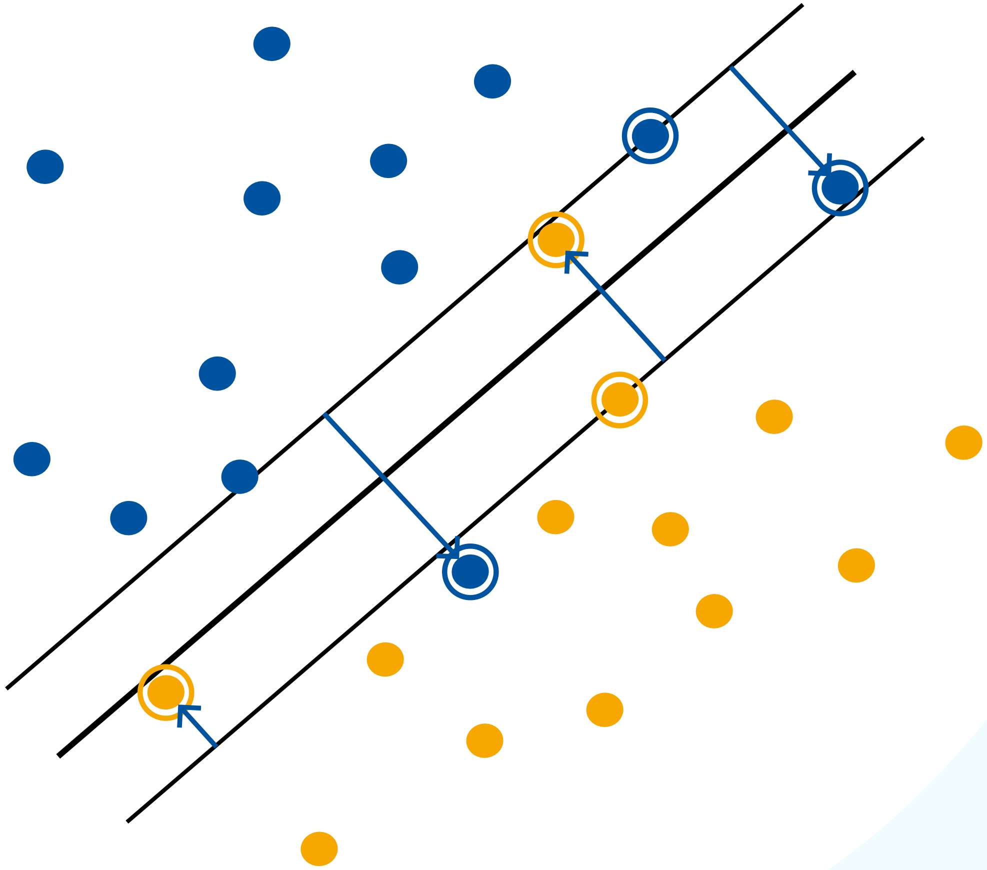
$$y(\mathbf{x}) = \mathbf{w}^{\mathsf{T}} \phi(\mathbf{x}) + b$$

with
$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)$$

is still costly for high-dimensional feature spaces $\phi(\mathbf{x})$.

Support Vector Machines

1. Maximum Margin Classification
2. Primal Formulation
3. Dual Formulation
4. **Soft-Margin SVMs**
5. Non-linear SVMs
6. Error Function Analysis

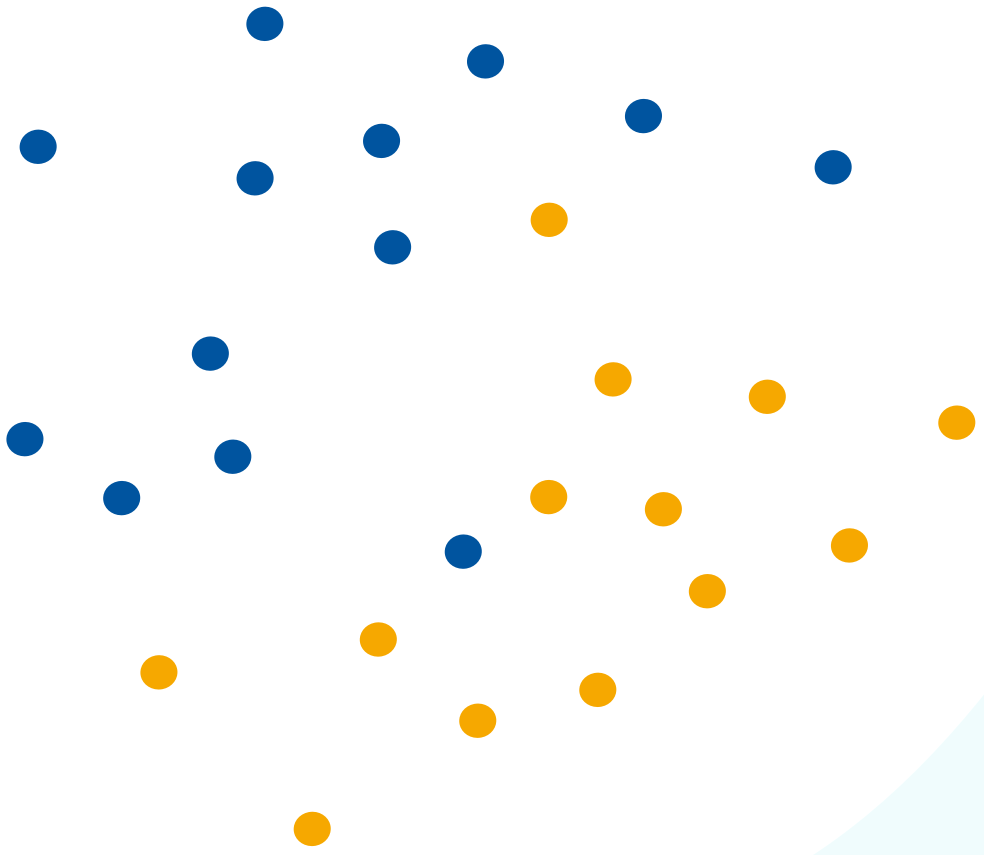


Soft-Margin SVM

- So far, we assumed linearly separable data.
  - Our current formulation has no solution if the data are not linearly separable!

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

such that $t_n(\mathbf{w}^\mathsf{T}\mathbf{x}_n + b) \geq 1 \quad \forall n$



- Need to introduce tolerance to outlier data points.
  - The resulting model is called soft-margin SVM.

Slack Variables

- For non-linearly separable data, not all constraints can be satisfied:

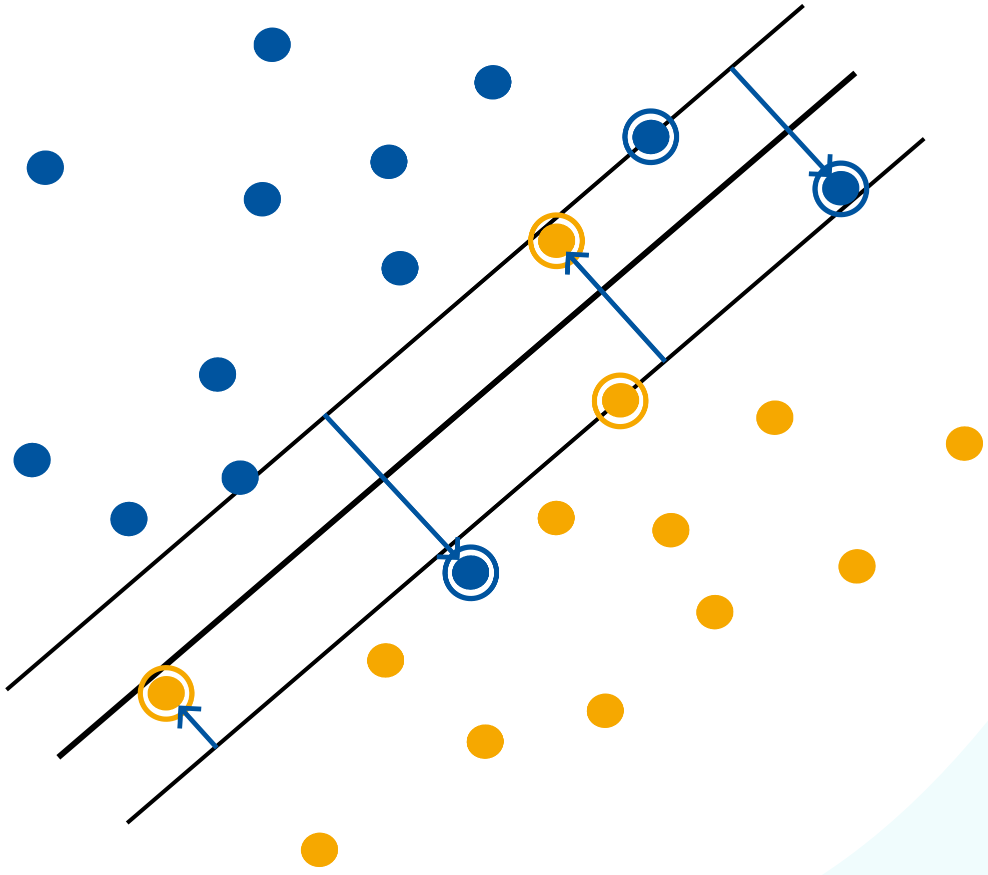
$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \geq +1 \quad \text{for } t_n = +1$$
$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \leq -1 \quad \text{for } t_n = -1$$

- Idea: Introduce slack variables $\xi_n \geq 0$:

$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \geq +1-\xi_n \quad \text{for } t_n = +1$$
$$\mathbf{w}^\mathsf{T}\mathbf{x}_n + b \leq -1+\xi_n \quad \text{for } t_n = -1$$

$\Rightarrow$ We allow some datapoints to violate the constraint.

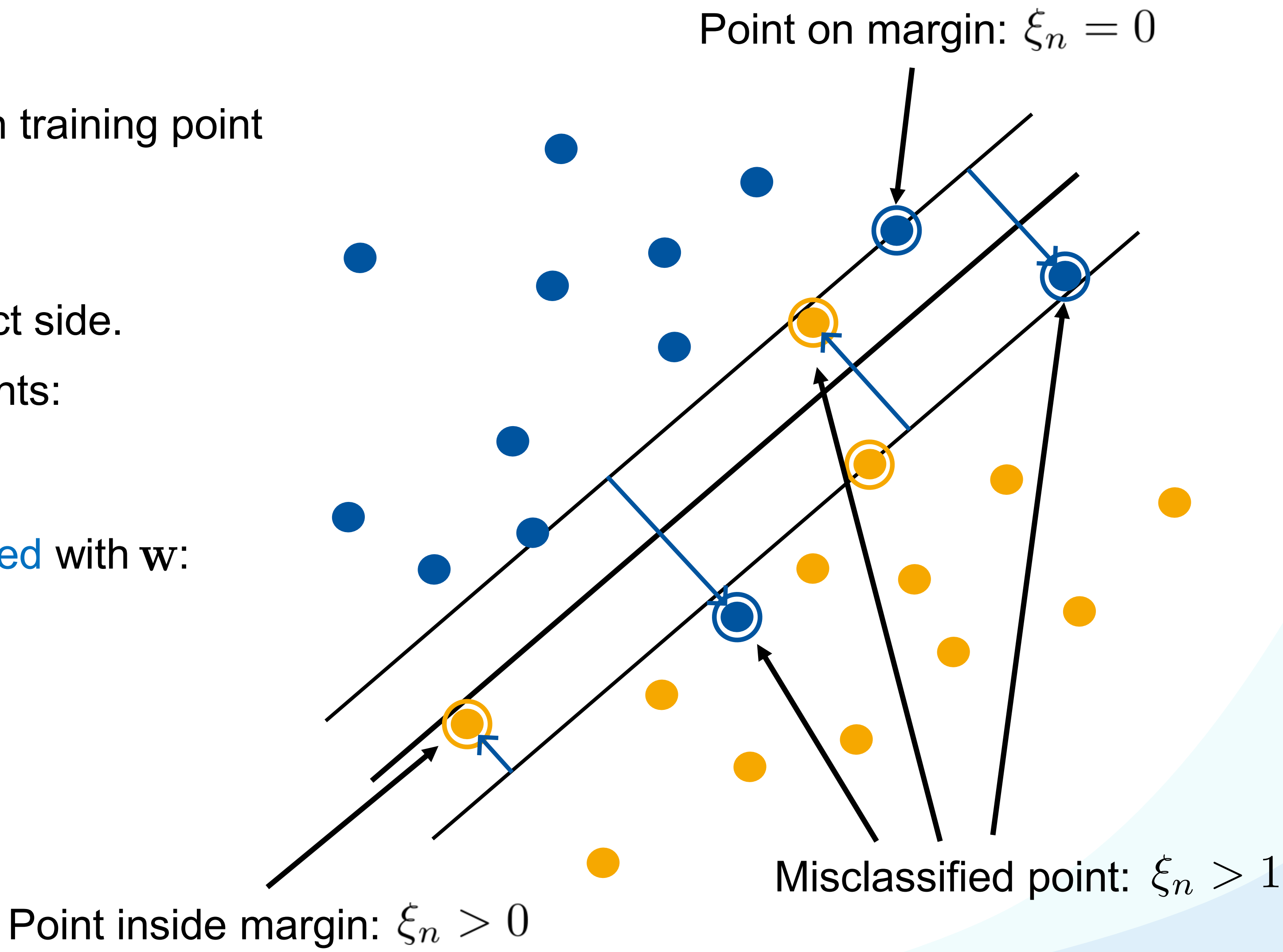
- For those points, the slack ξ_n makes up for the difference.



- Slack variables
  - One slack variable ξ_n for each training point
- Effect
  - $\xi_n = 0$ for points on the correct side.
  - Linear penalty for all other points:
    $\xi_n = |t_n - y(\mathbf{x}_n)|$
- Slack variables are jointly optimized with $\mathbf{w}$:

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n$$

where C is a tradeoff parameter.



New Primal Formulation

- Minimize

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \underbrace{\sum_{n=1}^{N} a_n\left[t_n(y(\mathbf{x}_n) - 1 + \xi_n\right]}_{\substack{\text{Constraint} \\ t_n y(\mathbf{x}_n) \geq 1 - \xi_n}} - \underbrace{\sum_{n=1}^{N}\mu_n\xi_n}_{\substack{\text{Constraint} \\ \xi_n \geq 0}}$$

- KKT conditions

$$\begin{aligned} a_n &\geq 0 & \mu_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 + \xi_n &\geq 0 & \xi_n &\geq 0 \\ a_n\left[t_n y(\mathbf{x}_n) - 1 + \xi_n\right] &= 0 & \mu_n\xi_n &= 0 \end{aligned}$$

New Dual Formulation

- Maximize

$$L_d(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m (\mathbf{x}_m^\top \mathbf{x}_n)$$

- Under the side conditions

$$\boxed{0 \leq a_n \leq C \quad \forall n}$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

This is the only difference to before.

New Solution

- The decision hyperplane is again a linear combination of training samples:

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \mathbf{x}_n$$

- This is still a sparse solution:
  - $a_n = 0$ for points on the correct side of the margin
  - Slack points with $\xi_n > 0$ are now also support vectors!

- Compute b by averaging over support vectors (points with $0 < a_n < C$):

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m \mathbf{x}_m^{\mathsf{T}} \mathbf{x}_n \right)$$

References and Further Reading

- More information about SVMs is available in Chapter 7.1 of Bishop's book.

Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006