Wissensentdeckung in Datenbanken

Support Vector Machine

Nico Piatkowski und Uwe Ligges

Informatik—Künstliche Intelligenz
Computergestützte Statistik
Technische Universität Dortmund

30.05.2017

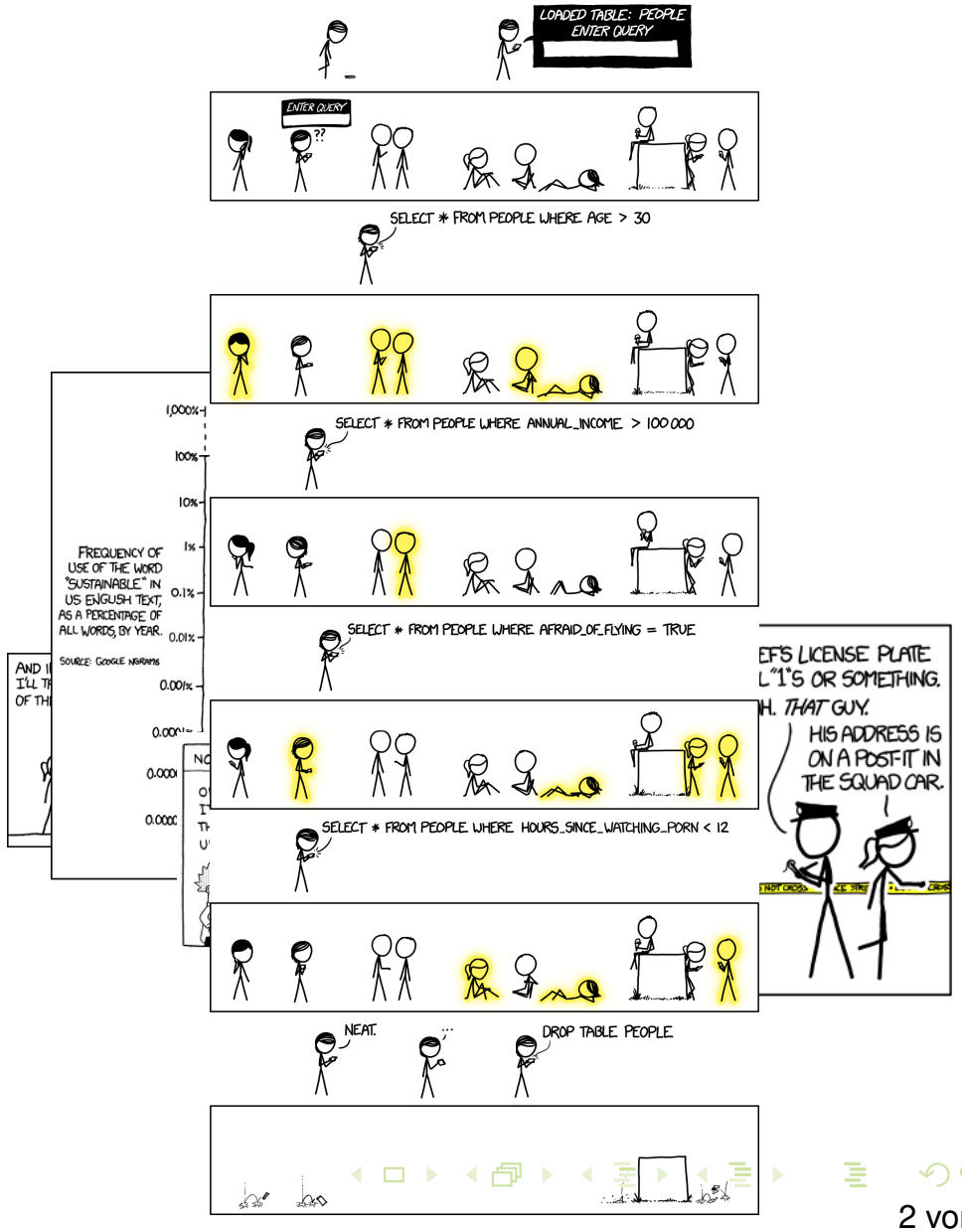
Überblick

Was bisher geschah...

- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung
- Überanpassung
- SQL, Häufige Mengen
- Klassifikation

Heute

- SVM



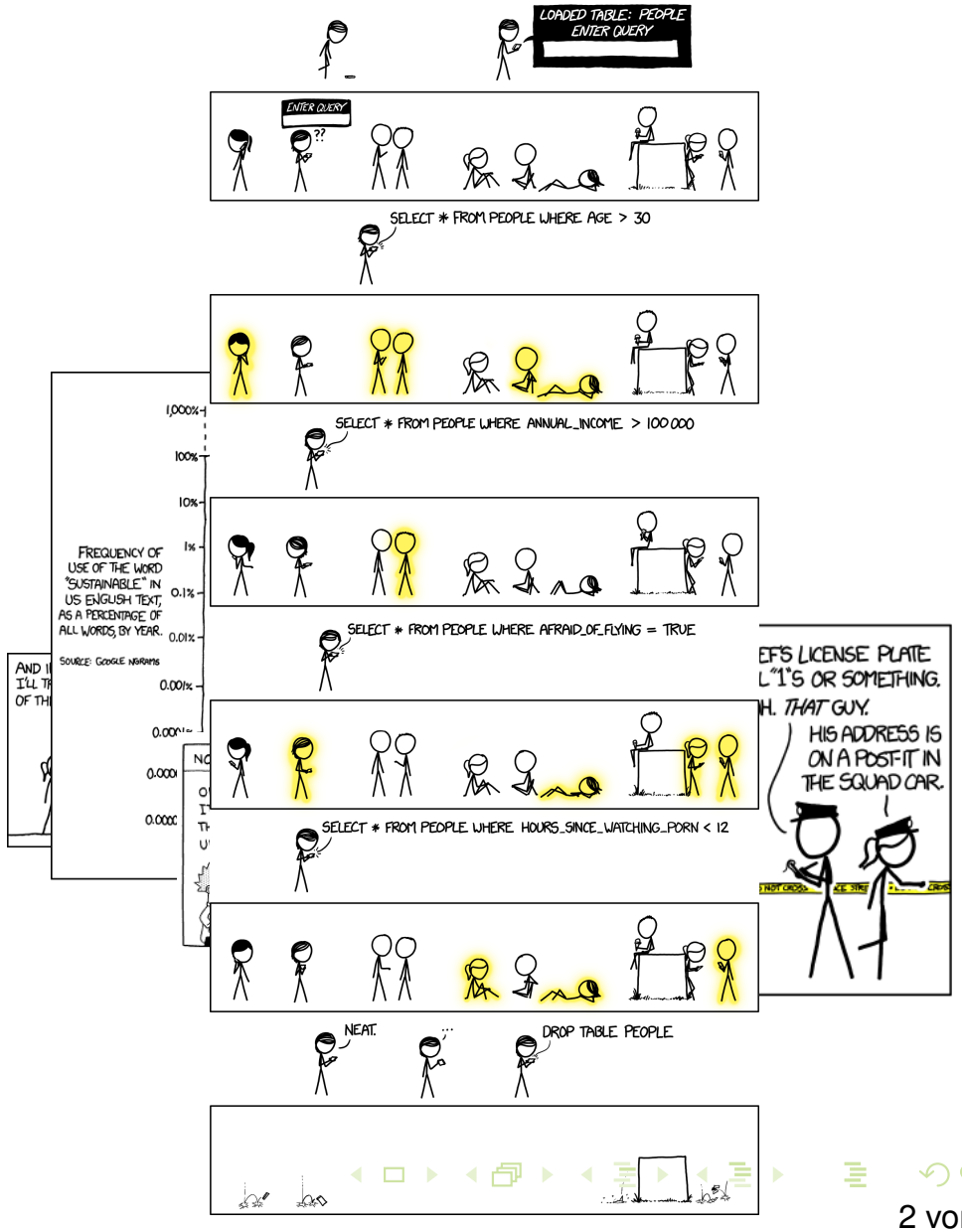
Überblick

Was bisher geschah...

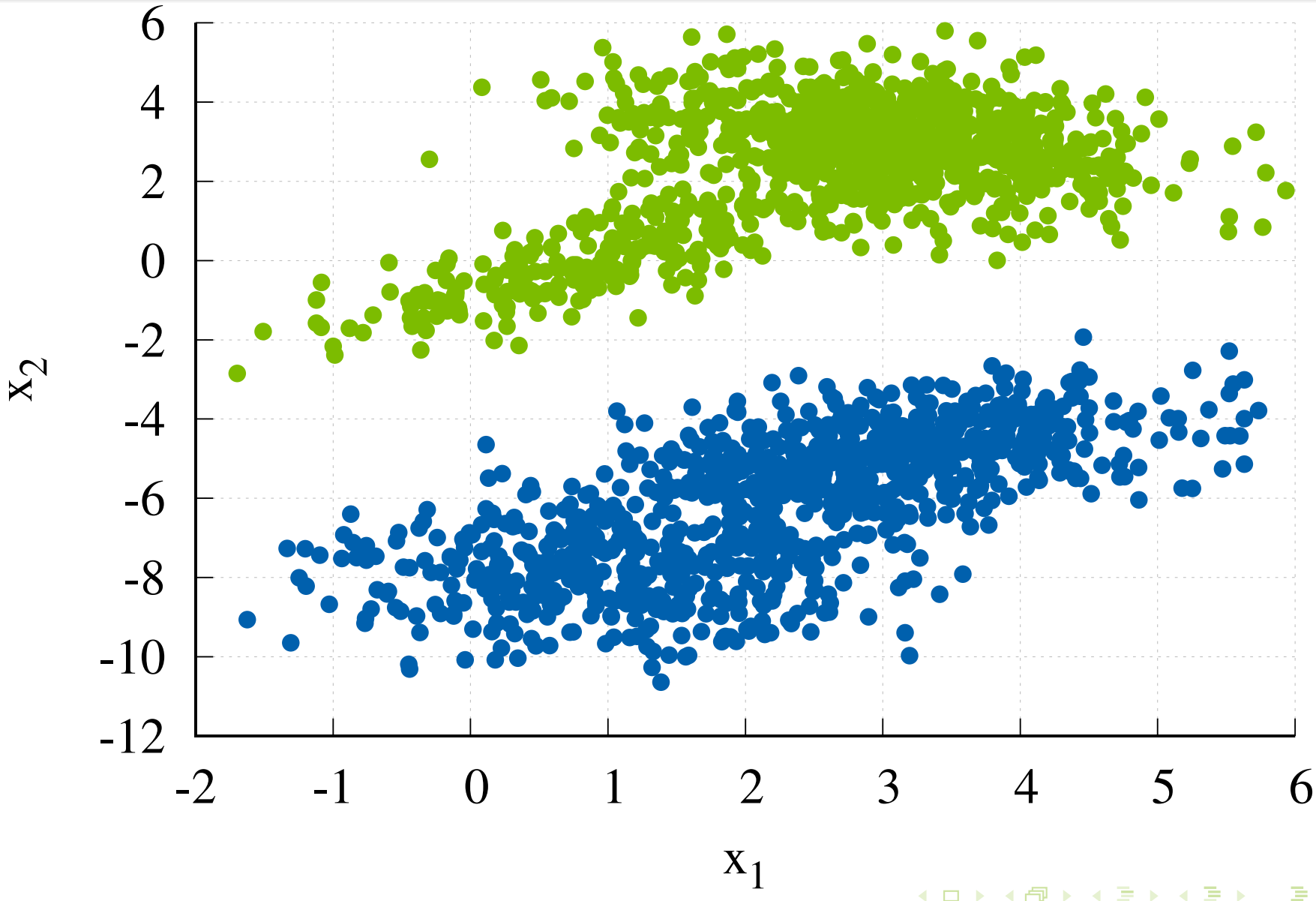
- Modellklassen
- Verlustfunktionen
- Numerische Optimierung
- Regularisierung
- Überanpassung
- SQL, Häufige Mengen
- Klassifikation

Heute

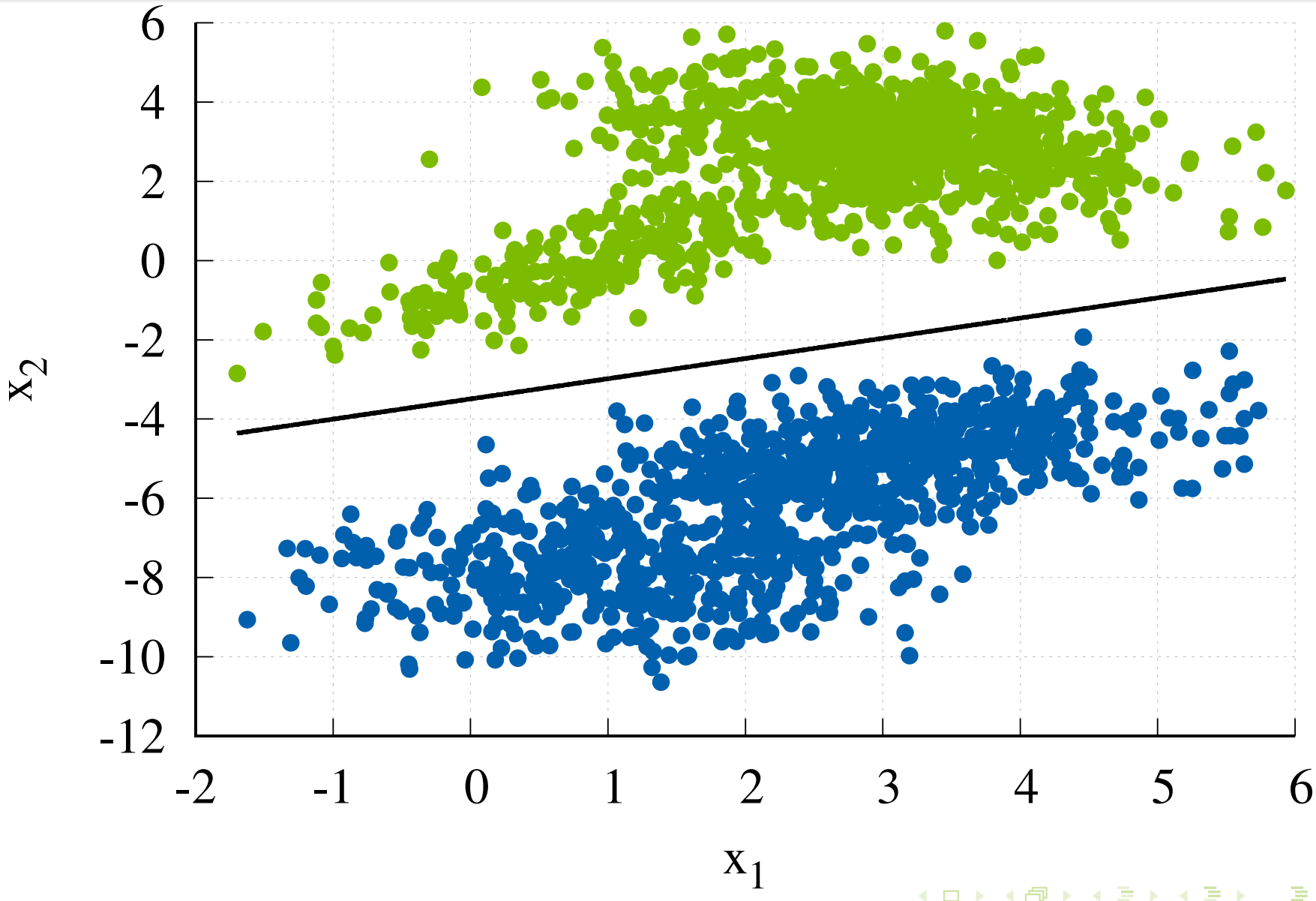
- SVM



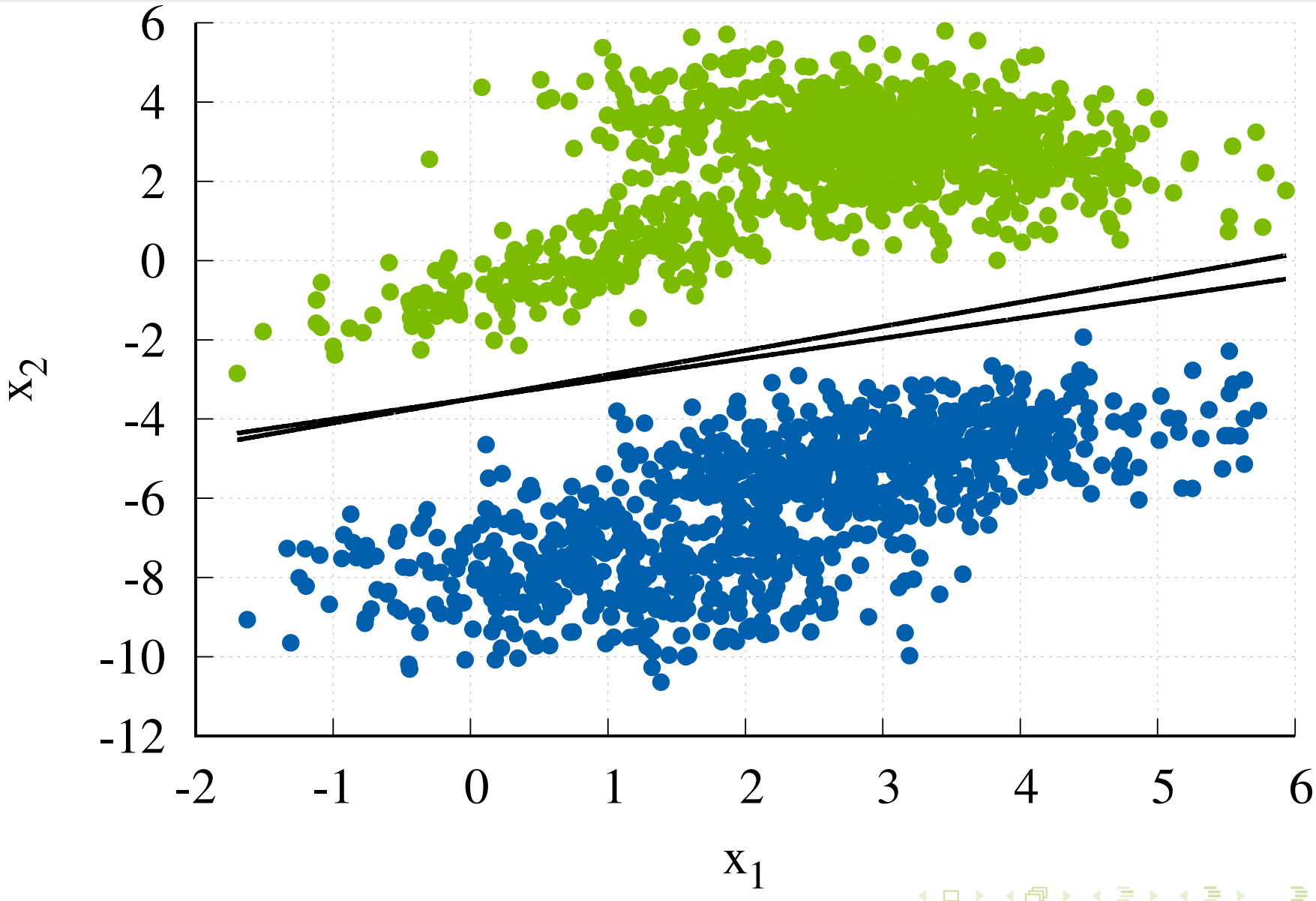
Klassifikation mit Hyperebenen



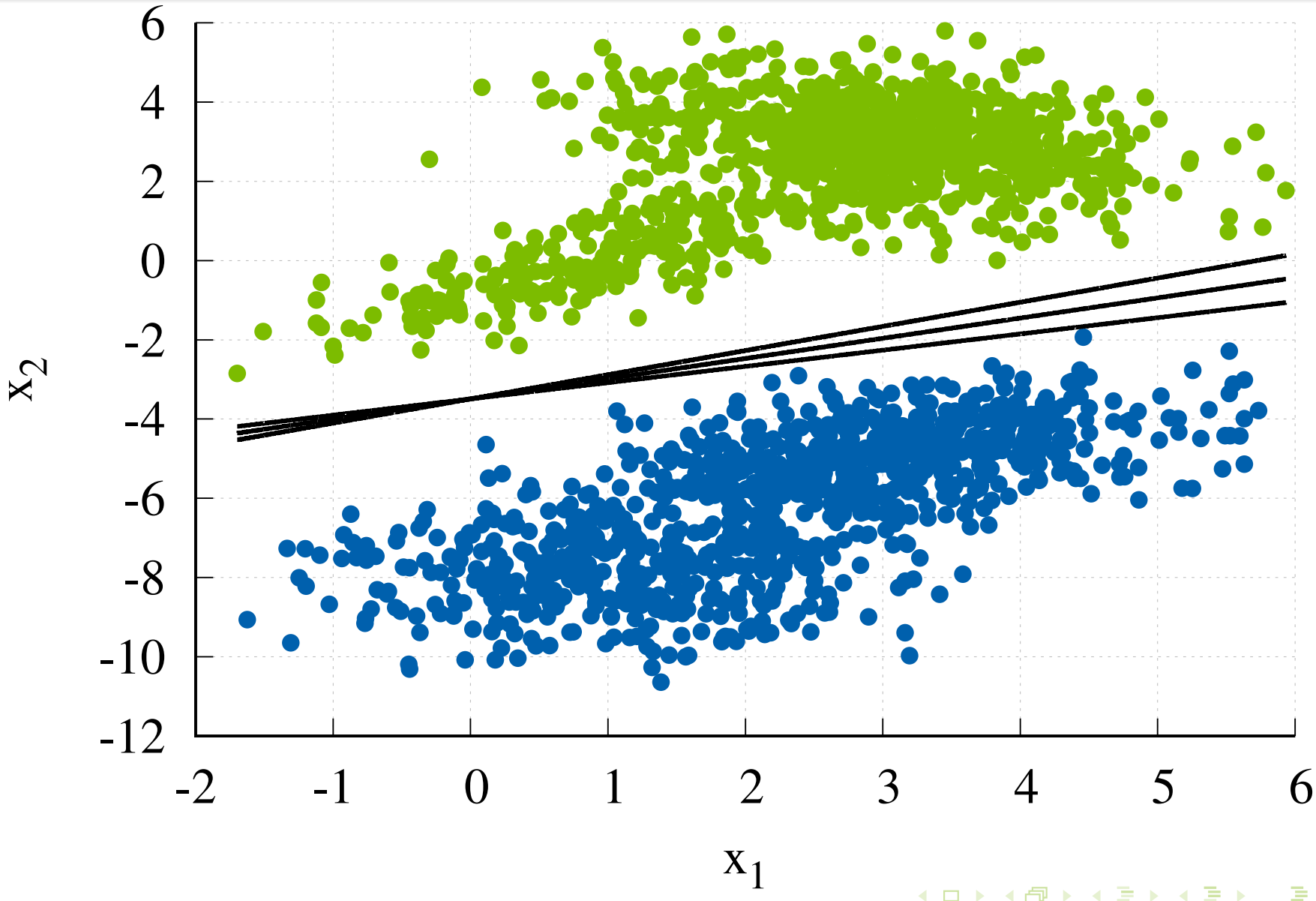
Klassifikation mit Hyperebenen



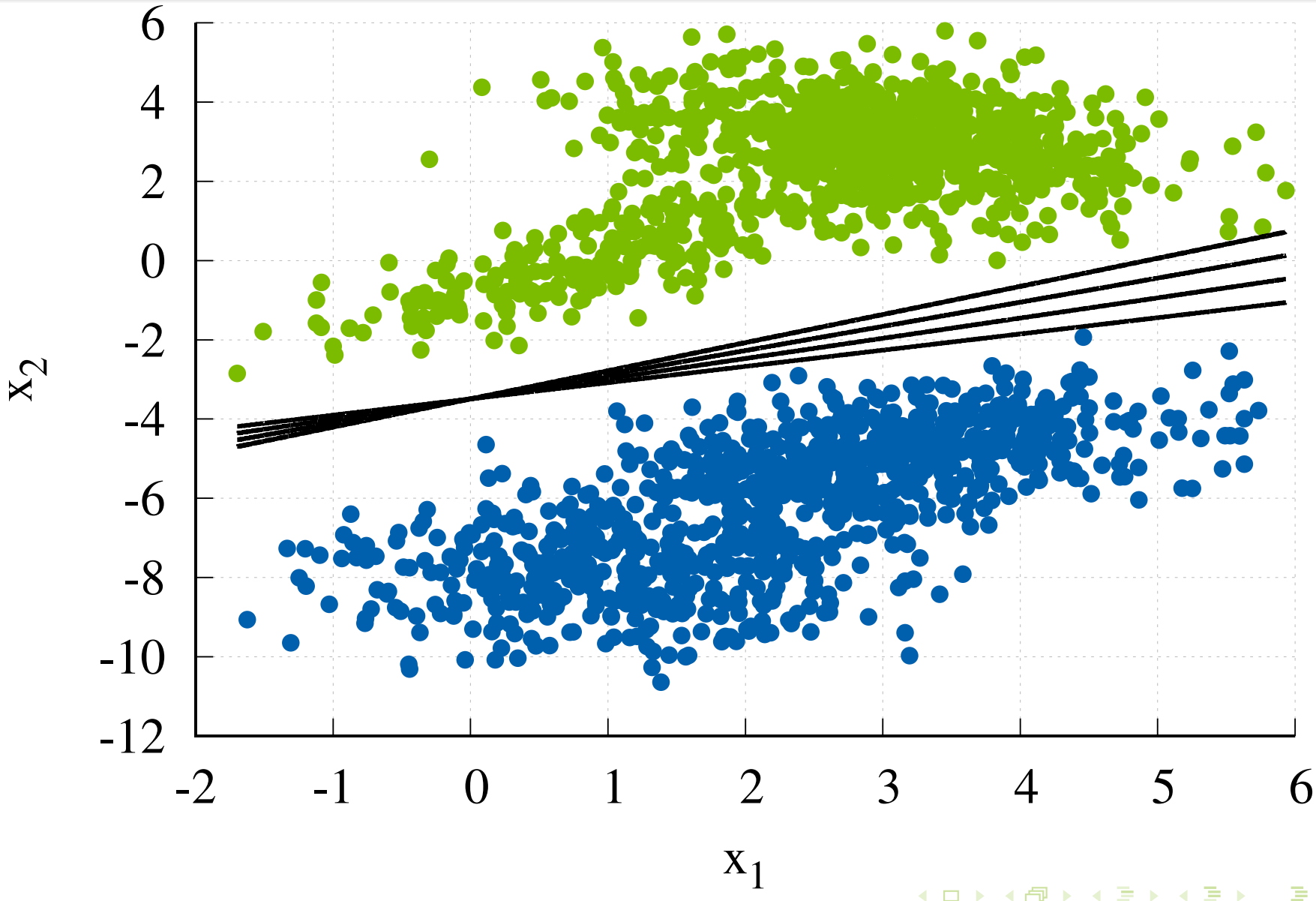
Klassifikation mit Hyperebenen



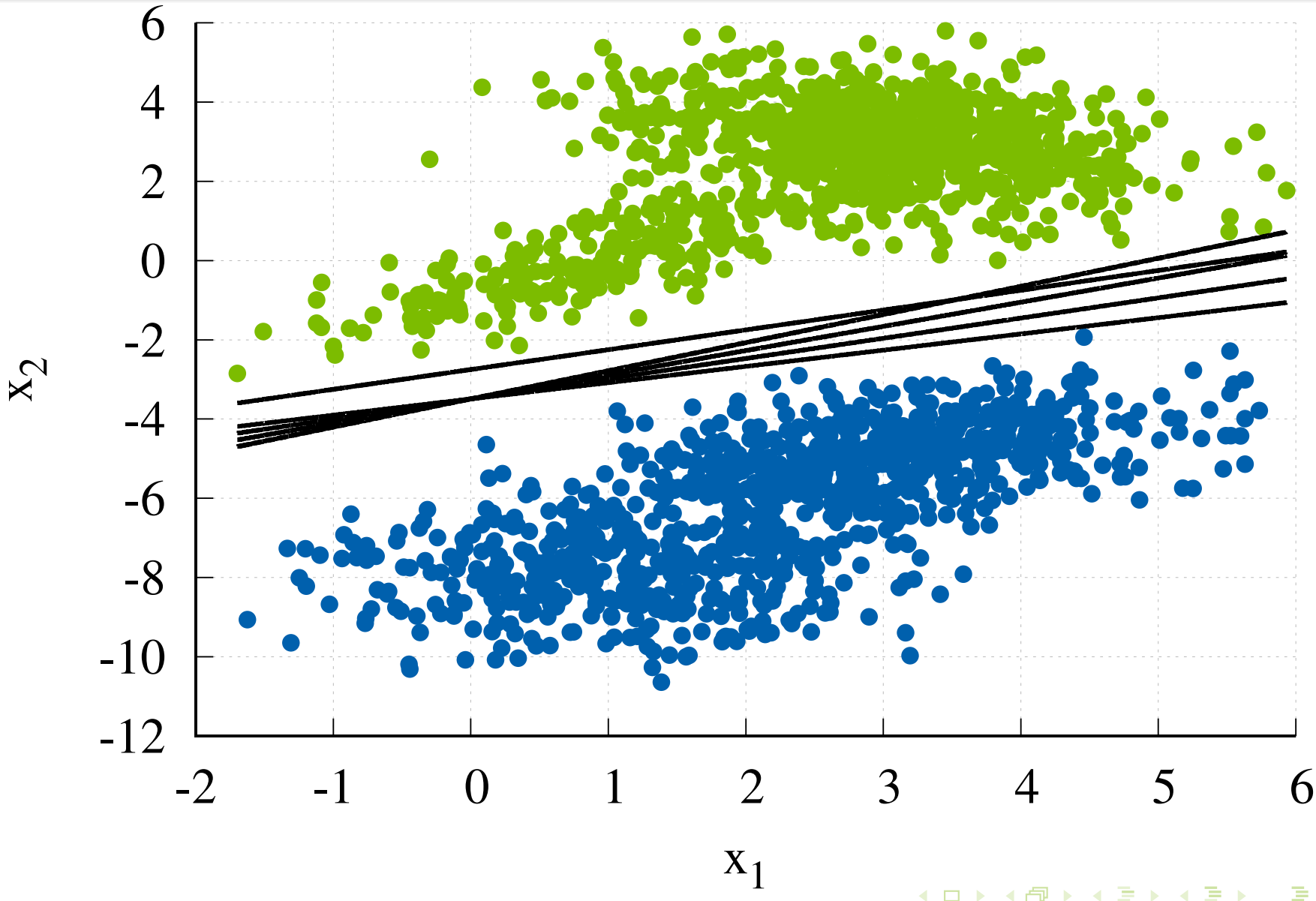
Klassifikation mit Hyperebenen



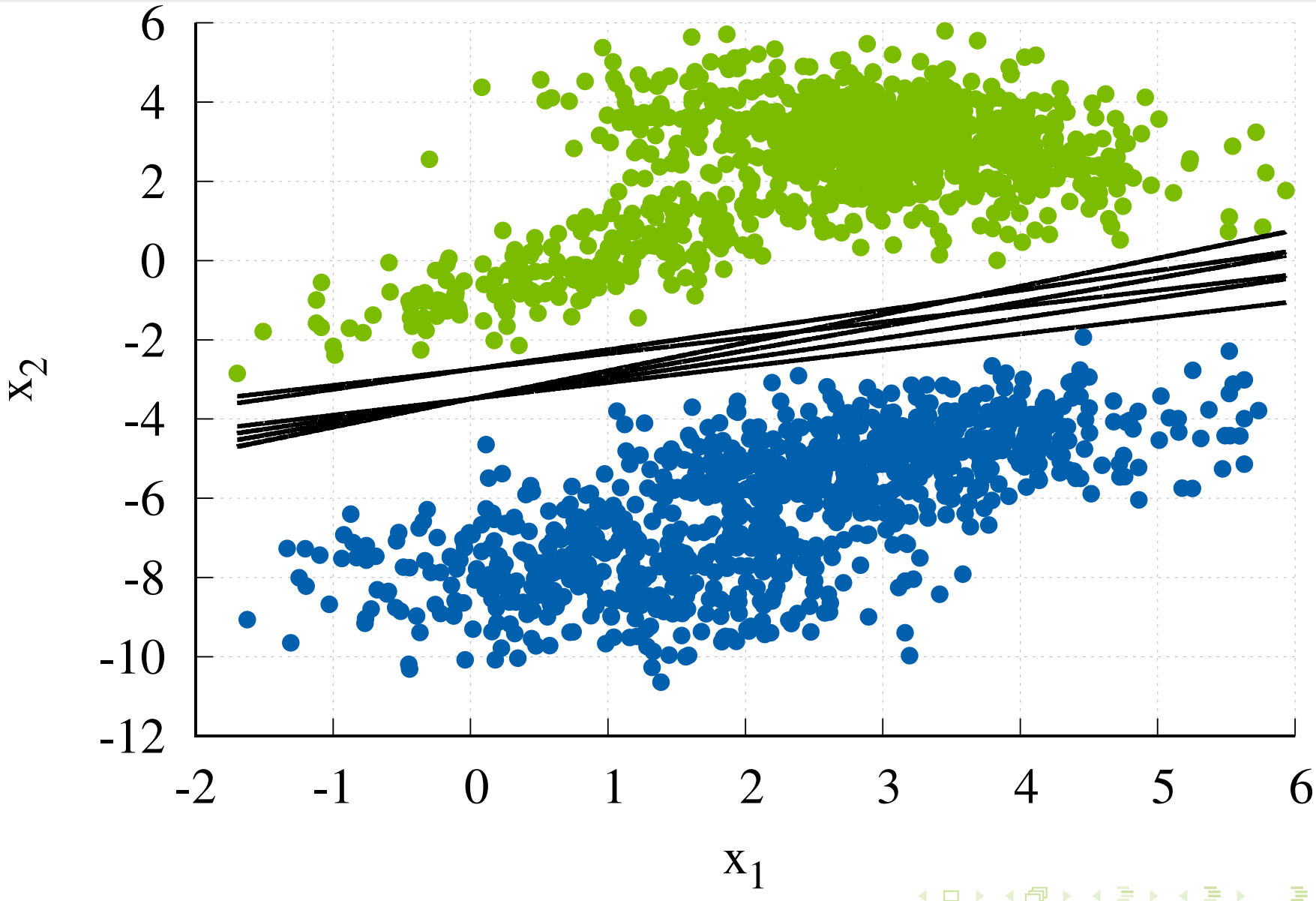
Klassifikation mit Hyperebenen



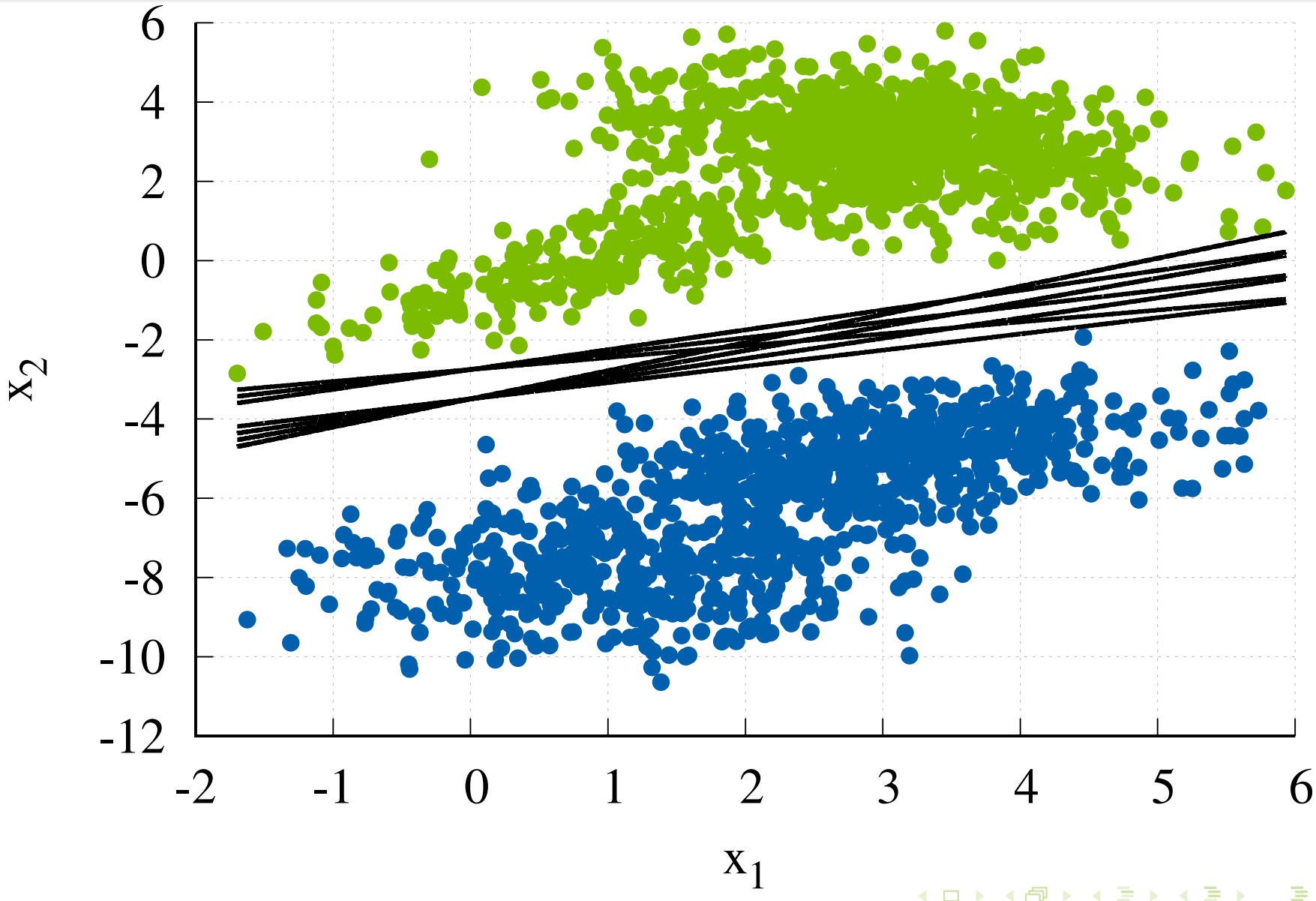
Klassifikation mit Hyperebenen



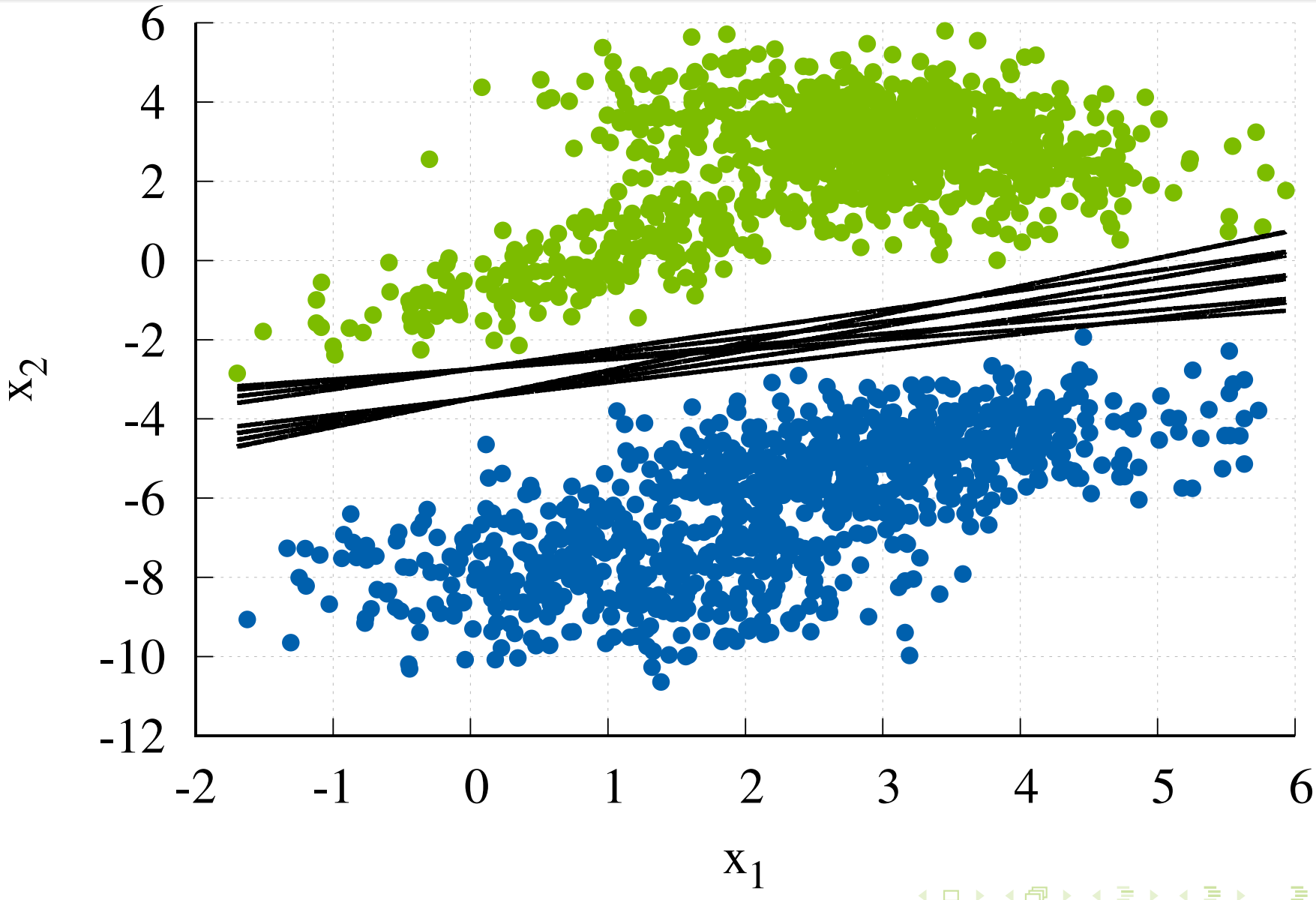
Klassifikation mit Hyperebenen



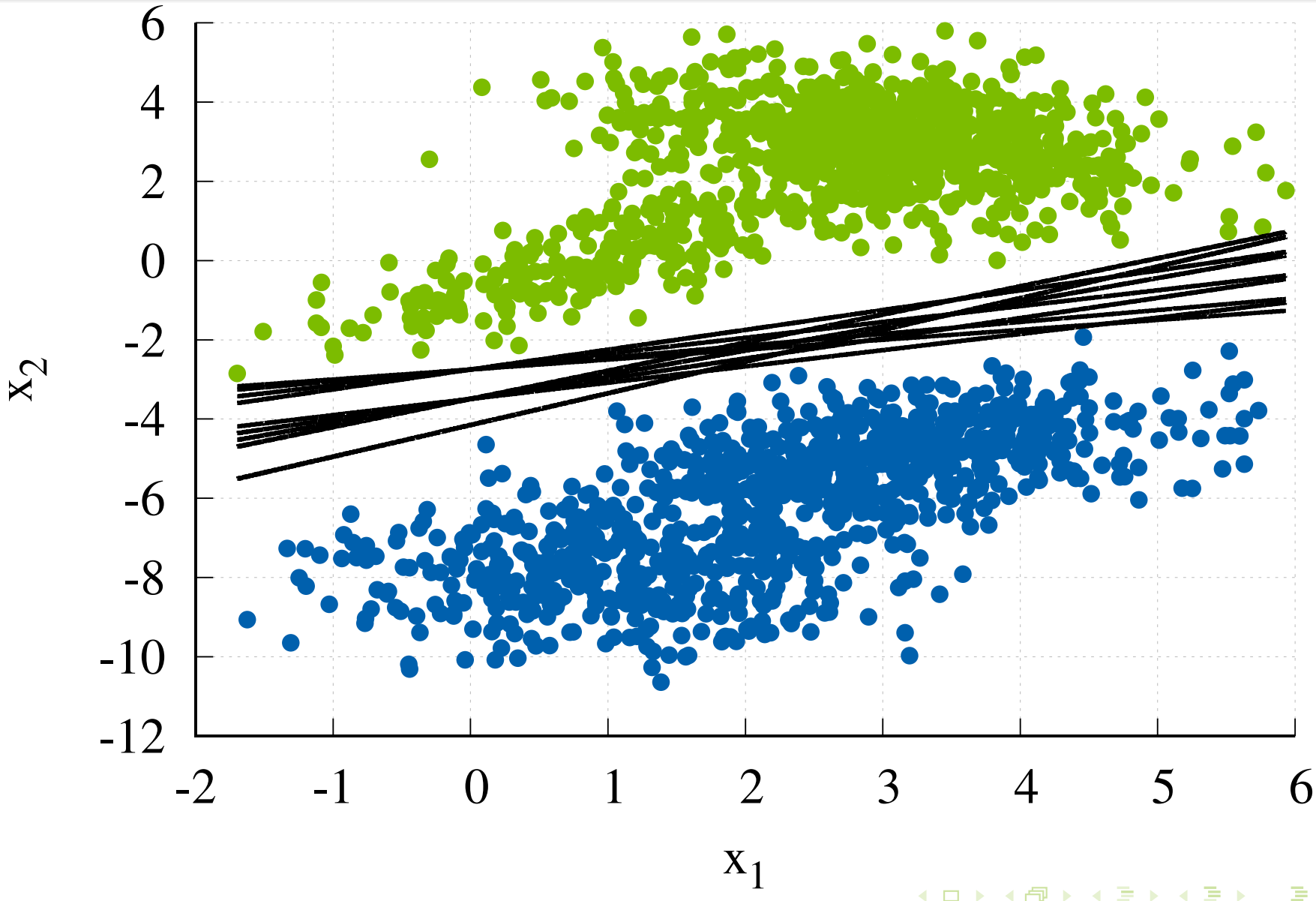
Klassifikation mit Hyperebenen



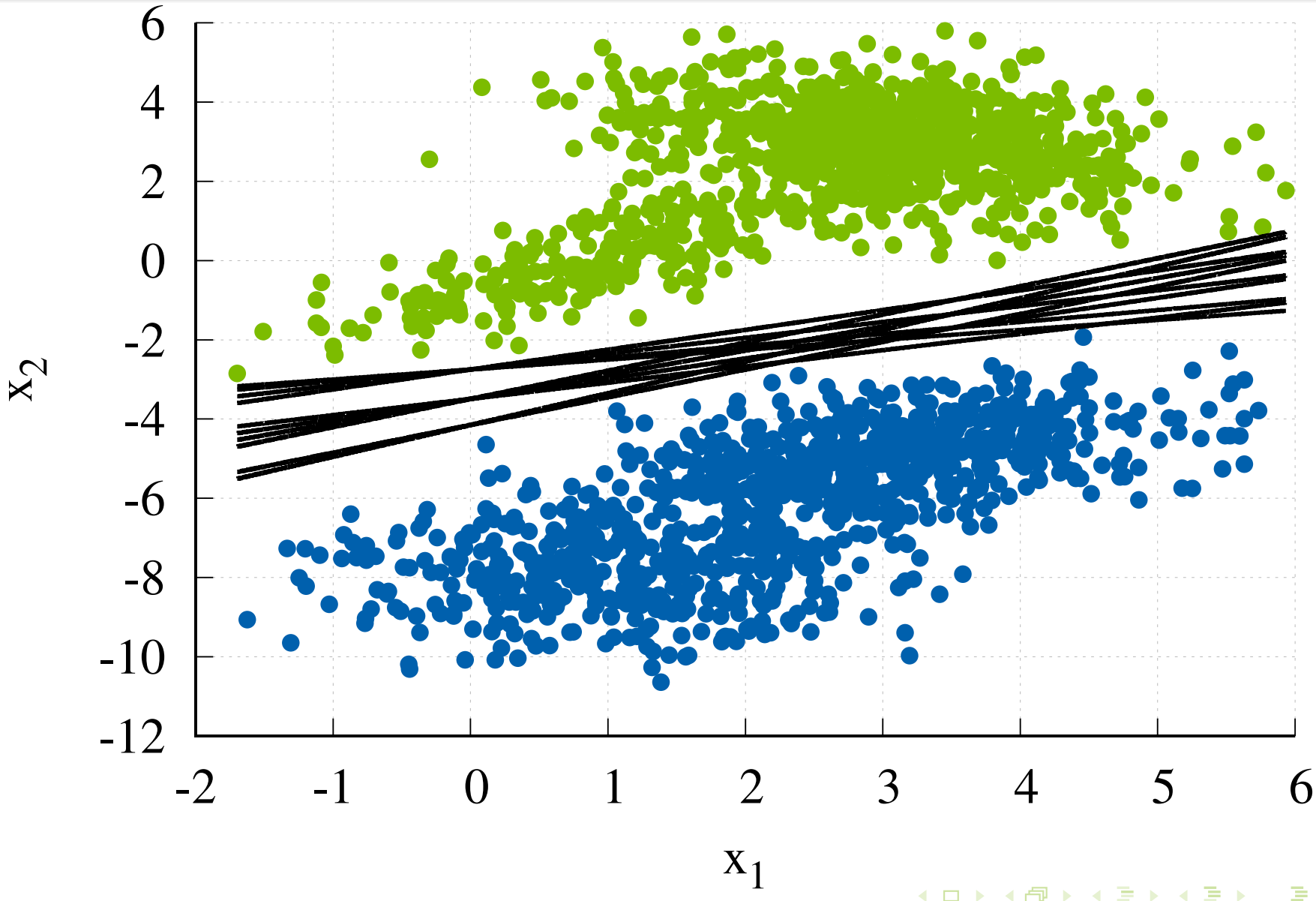
Klassifikation mit Hyperebenen



Klassifikation mit Hyperebenen



Klassifikation mit Hyperebenen



Hyperebenen

Hier: Daten $\boldsymbol{x}$ aus $\mathbb{R}^n$, Klasse y aus $\{-1, 1\}$

- Geradengleichung: $y = f_{\beta_0,\boldsymbol{\beta}}(\boldsymbol{x}) = \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- Wir wissen: $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle = \|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle(\boldsymbol{\beta}, \boldsymbol{x})$
  $\Rightarrow \cos \sphericalangle(\boldsymbol{\beta}, \boldsymbol{x})$ bestimmt das Vorzeichen von $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- $\operatorname{sign}(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \operatorname{sign}(\cos \sphericalangle(\boldsymbol{\beta}, \boldsymbol{x}))$

$$\operatorname{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \begin{cases} +1 & \beta_0 > -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle(\boldsymbol{\beta}, \boldsymbol{x}) \\ -1 & \beta_0 < -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle(\boldsymbol{\beta}, \boldsymbol{x}) \\ 0 & \text{sonst} \end{cases}$$

- $\|\boldsymbol{x}\|_2$ durch Daten bestimmt, β_0 und $\boldsymbol{\beta}$ werden gelernt

Hyperebenen

Hier: Daten $\boldsymbol{x}$ aus $\mathbb{R}^n$, Klasse y aus $\{-1, 1\}$

- Geradengleichung: $y = f_{\beta_0,\boldsymbol{\beta}}(\boldsymbol{x}) = \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- Wir wissen: $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle = \|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$
  $\Rightarrow \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$ bestimmt das Vorzeichen von $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- $\operatorname{sign}(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \operatorname{sign}(\cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}))$

$$\operatorname{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \begin{cases} +1 & \beta_0 > -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ -1 & \beta_0 < -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ 0 & \text{sonst} \end{cases}$$

- $\|\boldsymbol{x}\|_2$ durch Daten bestimmt, β_0 und $\boldsymbol{\beta}$ werden gelernt

Hyperebenen

Hier: Daten $\boldsymbol{x}$ aus $\mathbb{R}^n$, Klasse y aus $\{-1, 1\}$

- Geradengleichung: $y = f_{\beta_0,\boldsymbol{\beta}}(\boldsymbol{x}) = \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- Wir wissen: $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle = \|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$
  $\Rightarrow \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$ bestimmt das Vorzeichen von $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- $\operatorname{sign}(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \operatorname{sign}(\cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}))$

$$\operatorname{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \begin{cases} +1 & \beta_0 > -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ -1 & \beta_0 < -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ 0 & \text{sonst} \end{cases}$$

- $\|x\|_2$ durch Daten bestimmt, β_0 und $\boldsymbol{\beta}$ werden gelernt

Hyperebenen

Hier: Daten $\boldsymbol{x}$ aus $\mathbb{R}^n$, Klasse y aus $\{-1, 1\}$

- Geradengleichung: $y = f_{\beta_0,\boldsymbol{\beta}}(\boldsymbol{x}) = \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- Wir wissen: $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle = \|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$
  $\Rightarrow \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x})$ bestimmt das Vorzeichen von $\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$
- $\operatorname{sign}(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \operatorname{sign}(\cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}))$

$$\operatorname{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle) = \begin{cases} +1 & \beta_0 > -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ -1 & \beta_0 < -\|\boldsymbol{\beta}\|_2 \|\boldsymbol{x}\|_2 \cos \sphericalangle (\boldsymbol{\beta}, \boldsymbol{x}) \\ 0 & \text{sonst} \end{cases}$$

- $\|\boldsymbol{x}\|_2$ durch Daten bestimmt, β_0 und $\boldsymbol{\beta}$ werden gelernt

Hyperebenenbestimmung als Optimierungsproblem (I)

Datensatz $\mathcal{D} = \{(y^1, \boldsymbol{x}^1), (y^2, \boldsymbol{x}^2), \ldots, (y^N, \boldsymbol{x}^N)\}$

Idealisiert:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

$$\text{s.t.} \quad y^i = \text{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \quad \forall 1 \leq i \leq N$$

Pragmatisch:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

$$\text{s.t.} \quad y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \geq 1 \quad \forall 1 \leq i \leq N$$

Hyperebenenbestimmung als Optimierungsproblem (I)

Datensatz $\mathcal{D} = \{(y^1, \boldsymbol{x}^1), (y^2, \boldsymbol{x}^2), \ldots, (y^N, \boldsymbol{x}^N)\}$

Idealisiert:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

$$\text{s.t.} \quad y^i = \text{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \quad \forall 1 \leq i \leq N$$

Pragmatisch:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

$$\text{s.t.} \quad y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \geq 1 \quad \forall 1 \leq i \leq N$$

Hyperebenenbestimmung als Optimierungsproblem (I)

Datensatz $\mathcal{D} = \{(y^1, \boldsymbol{x}^1), (y^2, \boldsymbol{x}^2), \ldots, (y^N, \boldsymbol{x}^N)\}$

Idealisiert:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

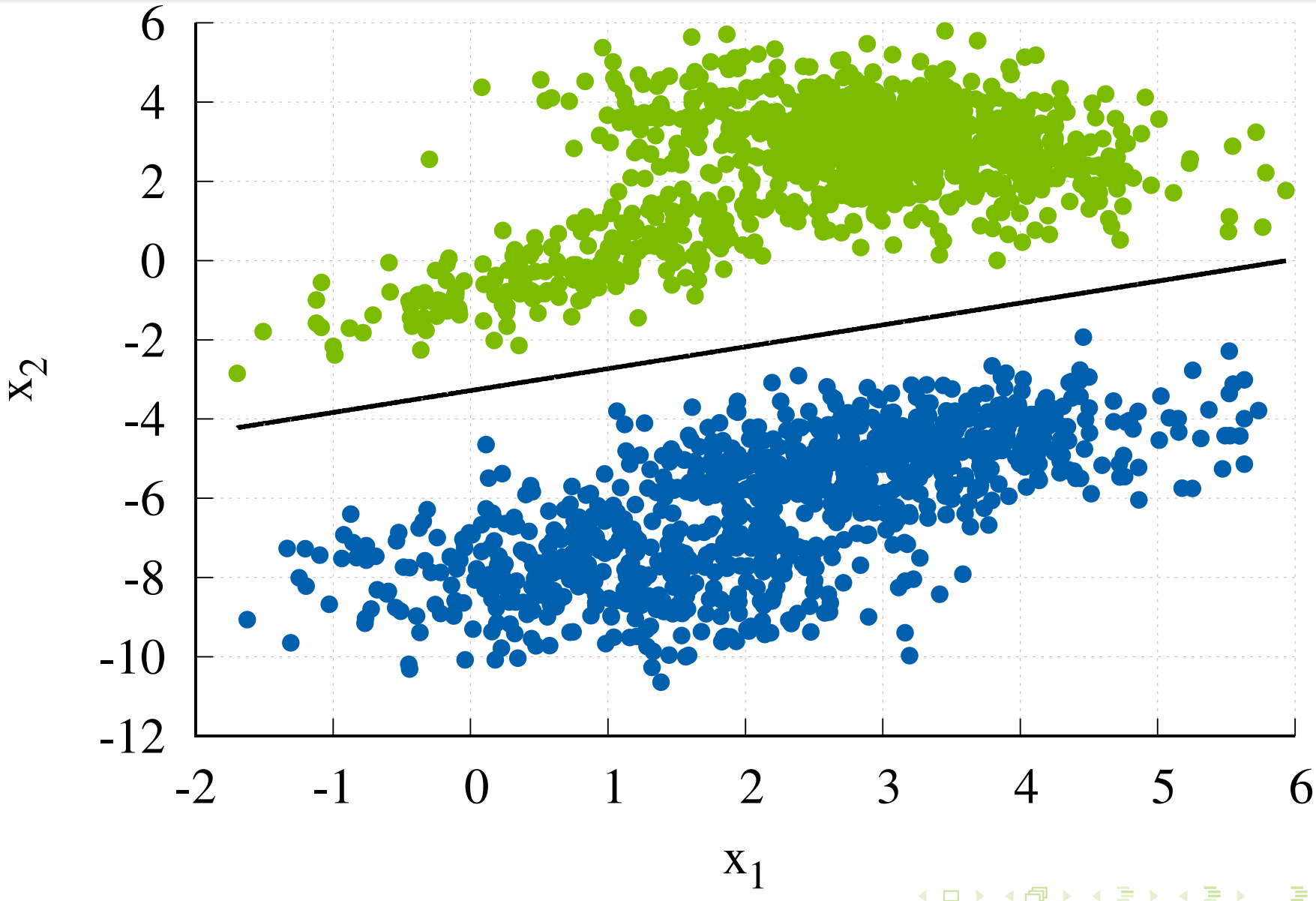
$$\text{s.t.} \quad y^i = \text{sign}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \quad \forall 1 \leq i \leq N$$

Pragmatisch:

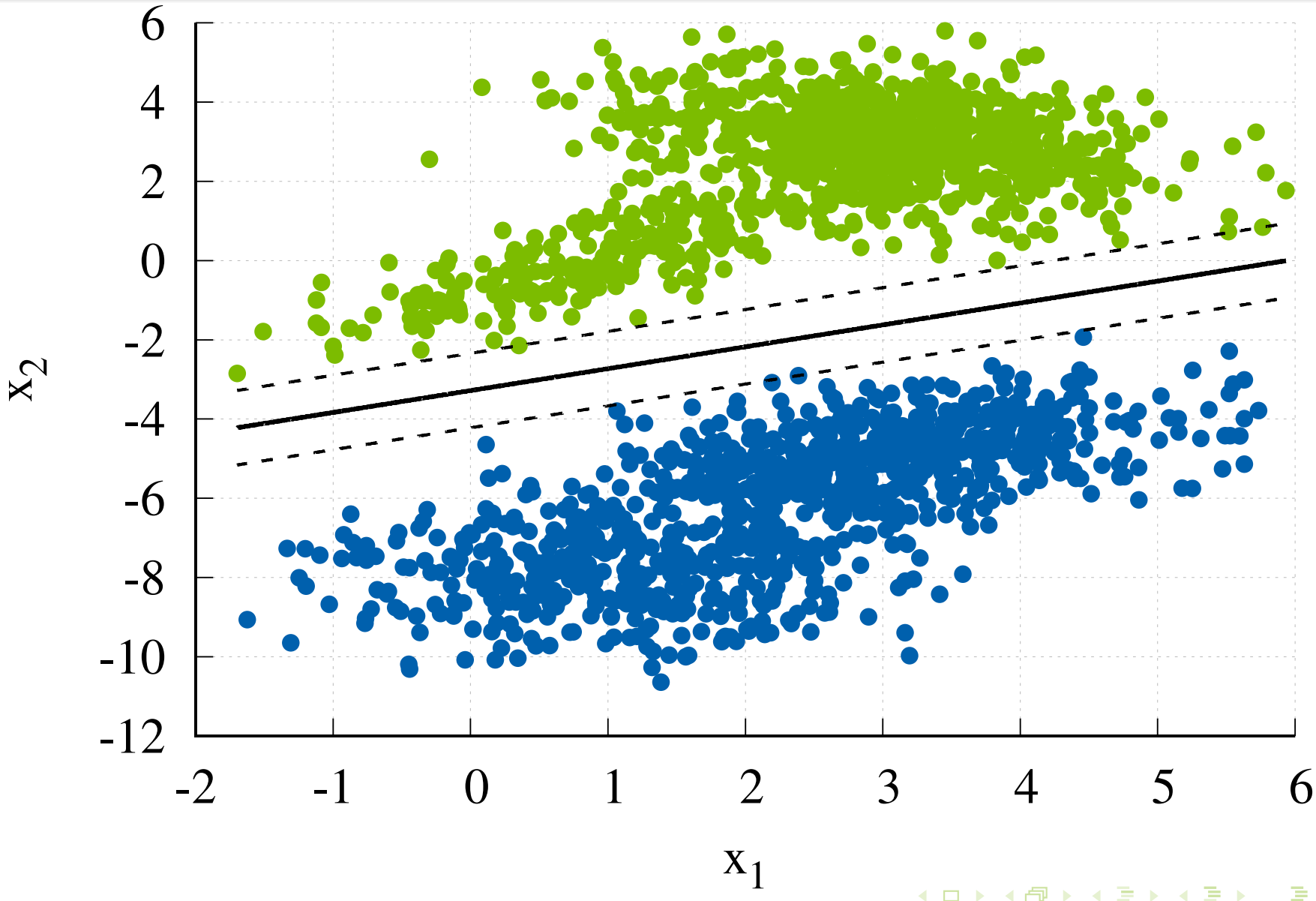
$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \|\boldsymbol{\beta}\|_2^2$$

$$\text{s.t.} \quad y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle) \geq 1 \quad \forall 1 \leq i \leq N$$

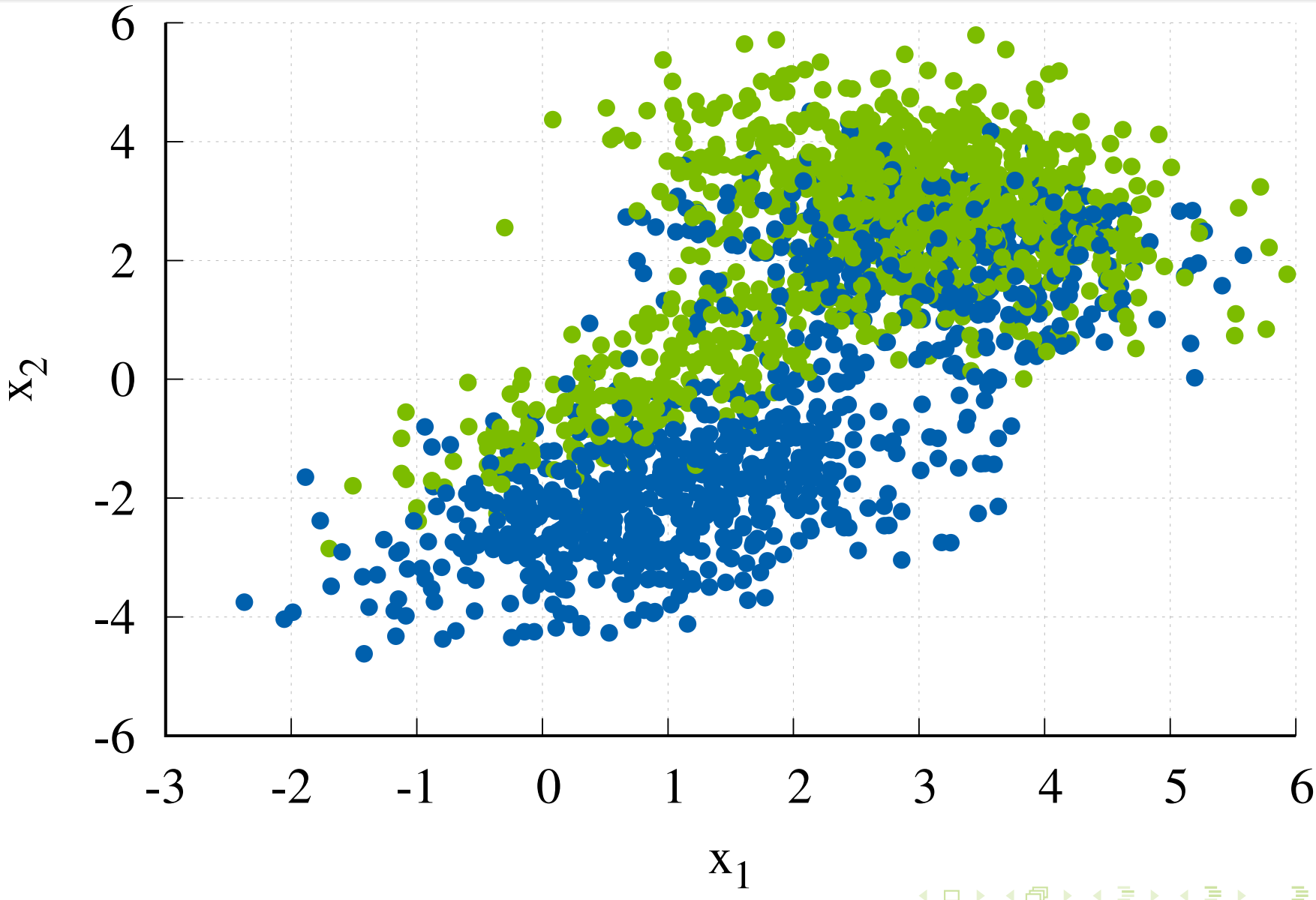
Klassifikation mit Hyperebenen



Klassifikation mit Hyperebenen



Klassifikation mit Hyperebenen (nicht trennbar)



Klassifikation als Optimierungsproblem (II)

$$\min_{\beta_0,\boldsymbol{\beta},\boldsymbol{\zeta}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{N}\boldsymbol{\zeta}_i$$

$$\text{s.t.} \quad y^i(\beta_0 + \langle\boldsymbol{\beta}, \boldsymbol{x}^i\rangle) \geq 1 - \boldsymbol{\zeta}_i, \quad \forall 1 \leq i \leq N$$

$$\boldsymbol{\zeta}_i \geq 0, \quad \forall 1 \leq i \leq N$$

Also suchen wir die kleinsten ζ_i mit

$$\zeta_i \geq 1 - y^i(\beta_0 + \langle\boldsymbol{\beta}, \boldsymbol{x}^i\rangle) \text{ und } \zeta_i \geq 0$$

für jedes $1 \leq i \leq N$ und damit

$$\zeta_i^* = \max\{0, 1 - y^i(\beta_0 + \langle\boldsymbol{\beta}, \boldsymbol{x}^i\rangle)\}$$

Klassifikation als Optimierungsproblem (II)

$$\min_{\beta_0,\boldsymbol{\beta},\boldsymbol{\zeta}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{N}\boldsymbol{\zeta}_i$$

$$\text{s.t.} \quad y^i(\beta_0 + \left\langle \boldsymbol{\beta}, \boldsymbol{x}^i \right\rangle) \geq 1 - \boldsymbol{\zeta}_i, \quad \forall 1 \leq i \leq N$$

$$\boldsymbol{\zeta}_i \geq 0, \quad \forall 1 \leq i \leq N$$

Also suchen wir die kleinsten $\boldsymbol{\zeta}_i$ mit

$$\boldsymbol{\zeta}_i \geq 1 - y^i(\beta_0 + \left\langle \boldsymbol{\beta}, \boldsymbol{x}^i \right\rangle) \text{ und } \boldsymbol{\zeta}_i \geq 0$$

für jedes $1 \leq i \leq N$ und damit

$$\boldsymbol{\zeta}_i^* = \max\{0, 1 - y^i(\beta_0 + \left\langle \boldsymbol{\beta}, \boldsymbol{x}^i \right\rangle)\}$$

Klassifikation als Optimierungsproblem (III)

$$\zeta_i^* = \max\{0, 1 - y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle)\}$$

Einsetzen in Zielfunktion:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{N} \max\{0, 1 - y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle)$$

Minimierung der l_2-regularisierten Hinge-Verlustfunktion

$$\text{Hinge}(\beta_0, \boldsymbol{\beta}; \mathcal{D}) + \lambda R_2(\boldsymbol{\beta}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \max\{0, 1 - y f_{\beta_0, \boldsymbol{\beta}}(\boldsymbol{x})\} + \lambda \|\boldsymbol{\beta}\|_2^2$$

Klassifikation als Optimierungsproblem (III)

$$\zeta_i^* = \max\{0, 1 - y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle)\}$$

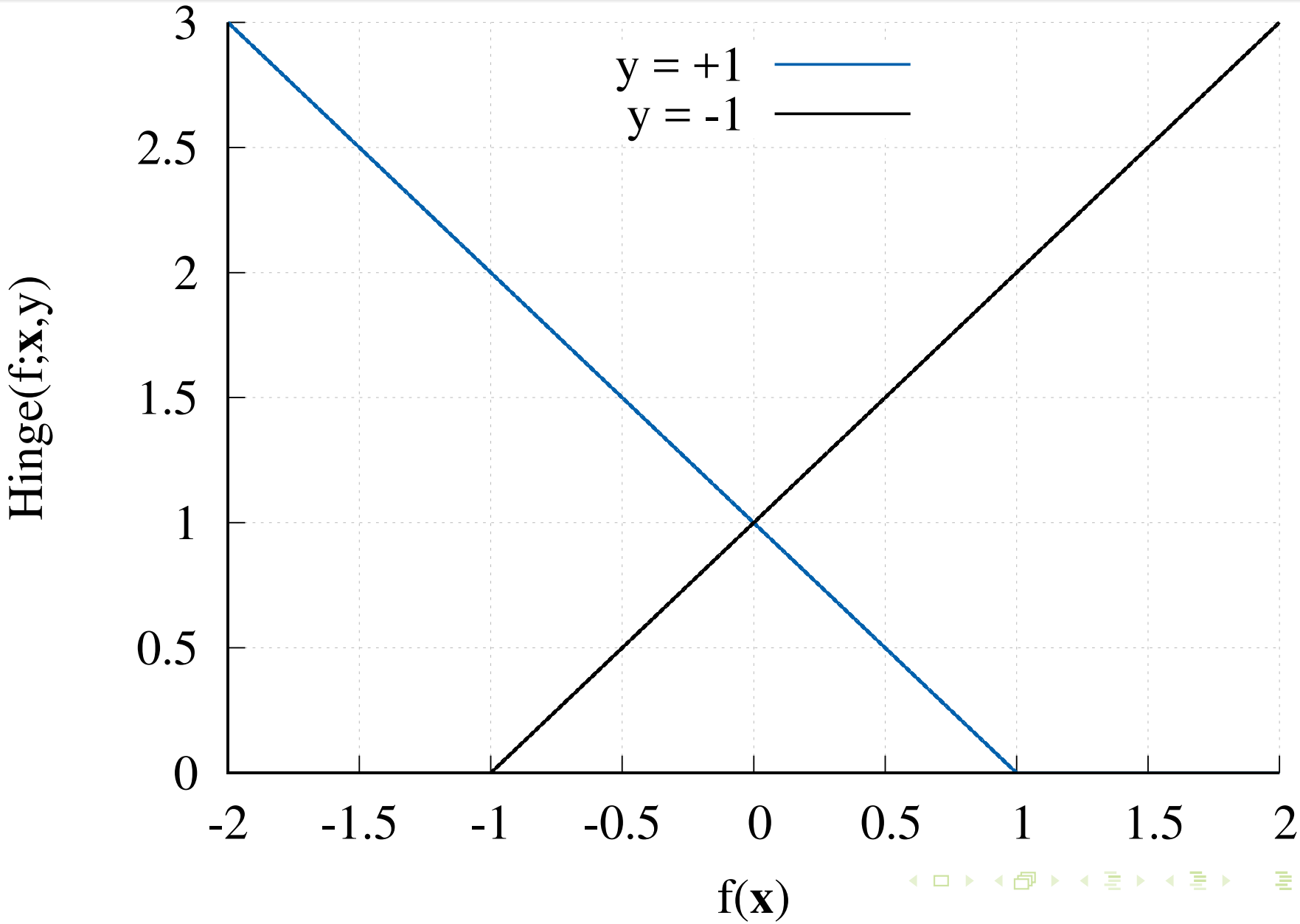
Einsetzen in Zielfunktion:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + C\sum_{i=1}^{N} \max\{0, 1 - y^i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}^i \rangle)$$

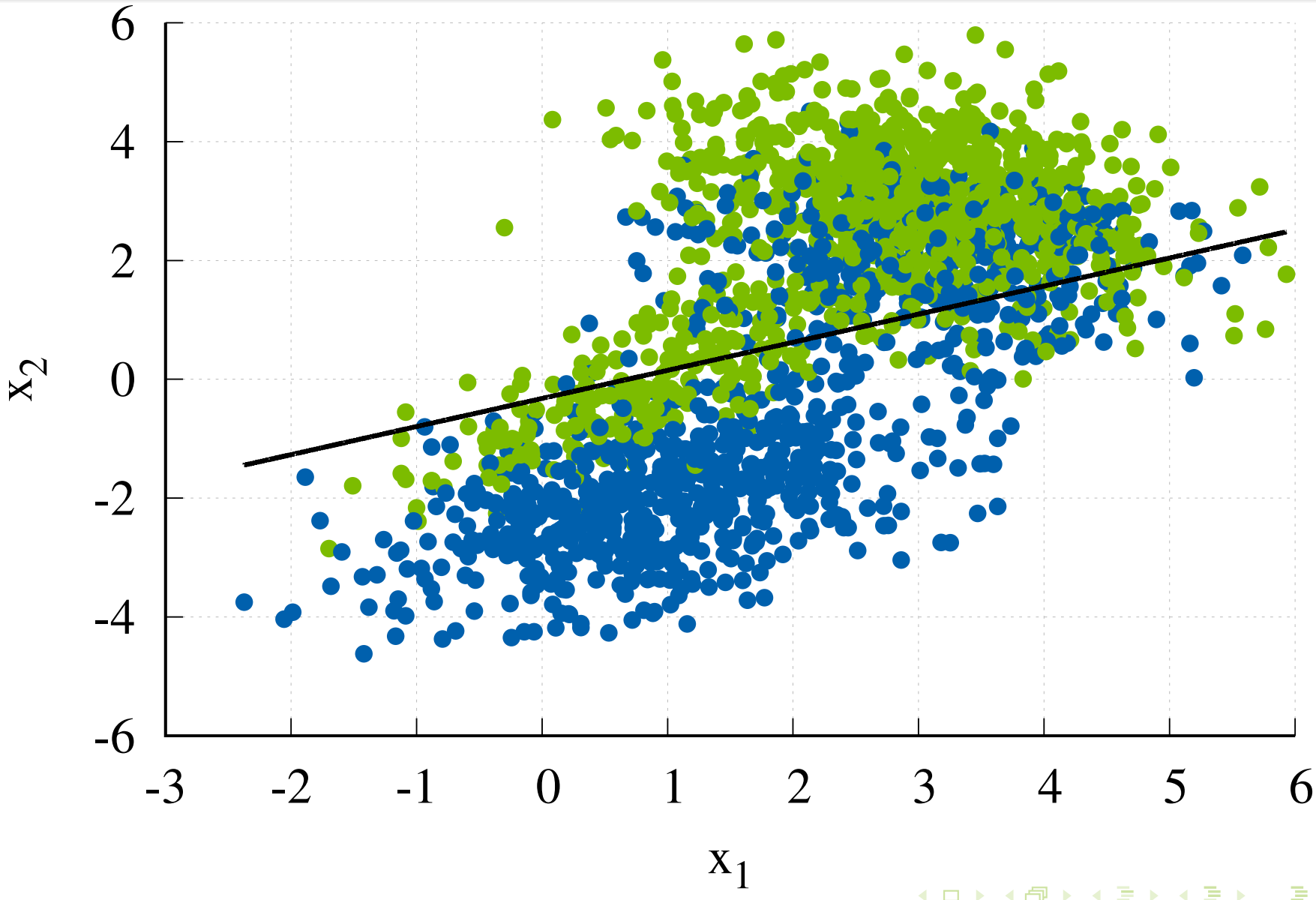
Minimierung der l_2-regularisierten Hinge-Verlustfunktion

$$\text{Hinge}(\beta_0, \boldsymbol{\beta}; \mathcal{D}) + \lambda R_2(\boldsymbol{\beta}) = \frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}} \max\{0, 1 - y f_{\beta_0, \boldsymbol{\beta}}(\boldsymbol{x})\} + \lambda \|\boldsymbol{\beta}\|_2^2$$

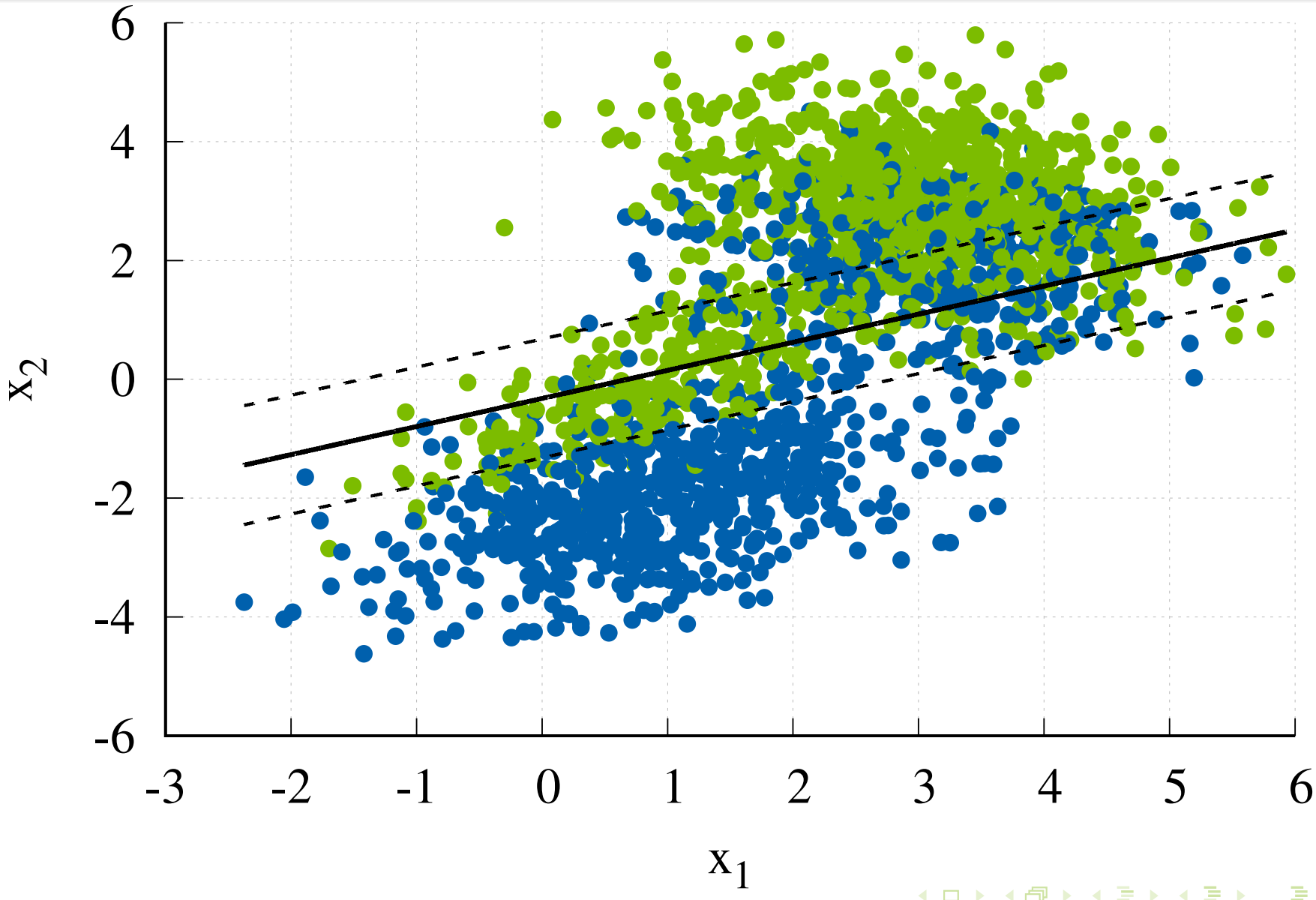
Hinge Verlustfunktion



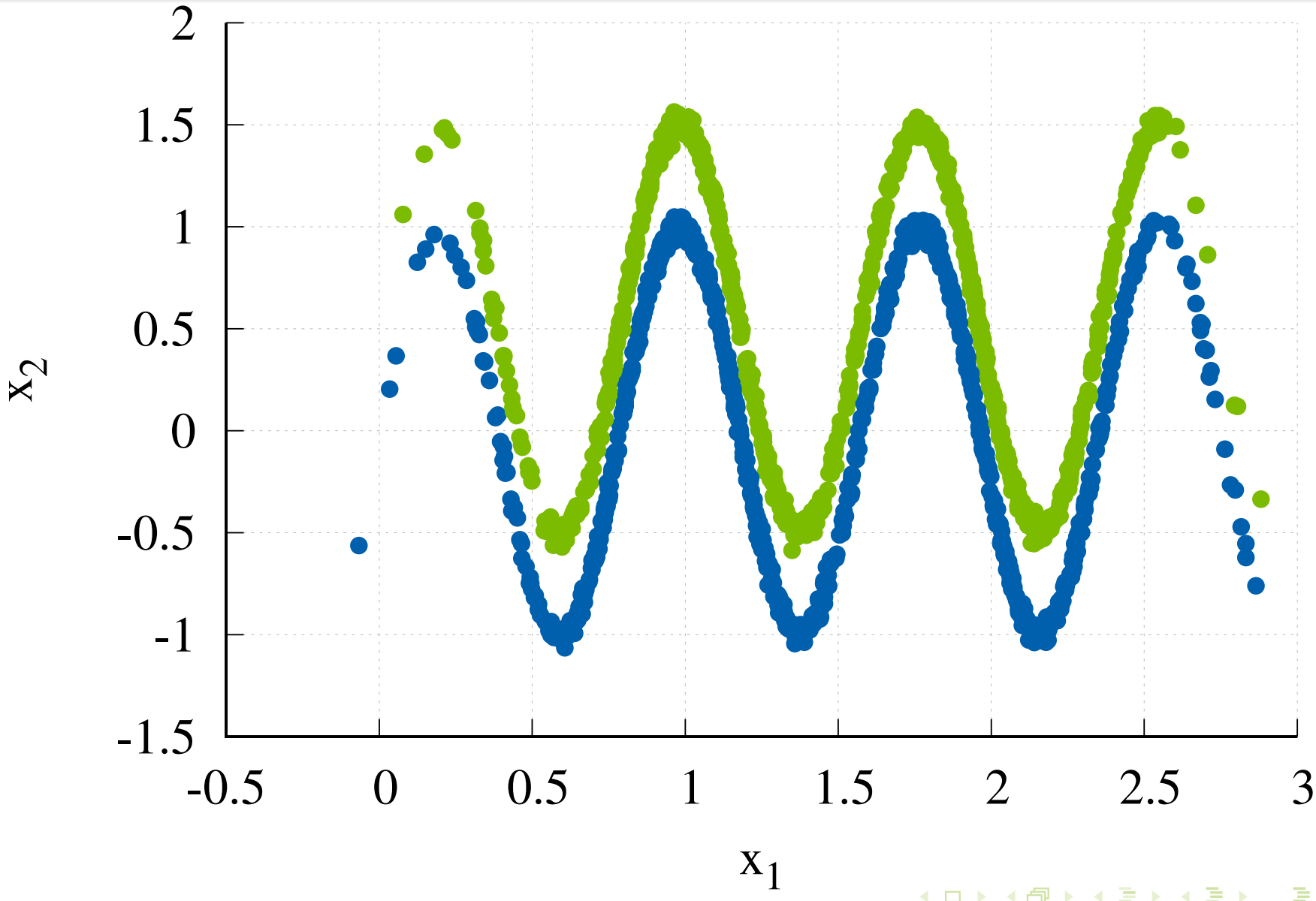
Klassifikation mit Hyperebenen (nicht trennbar)



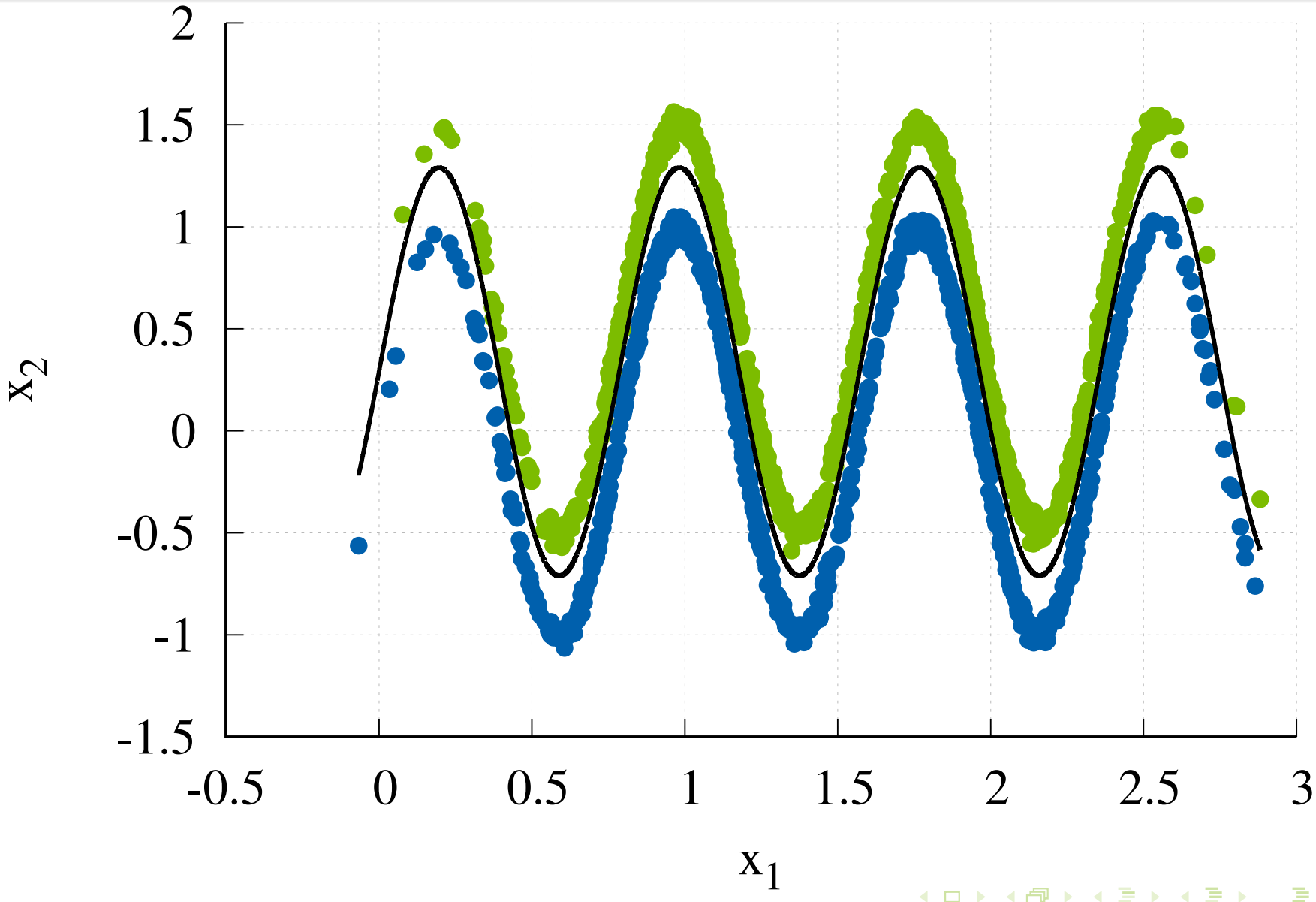
Klassifikation mit Hyperebenen (nicht trennbar)



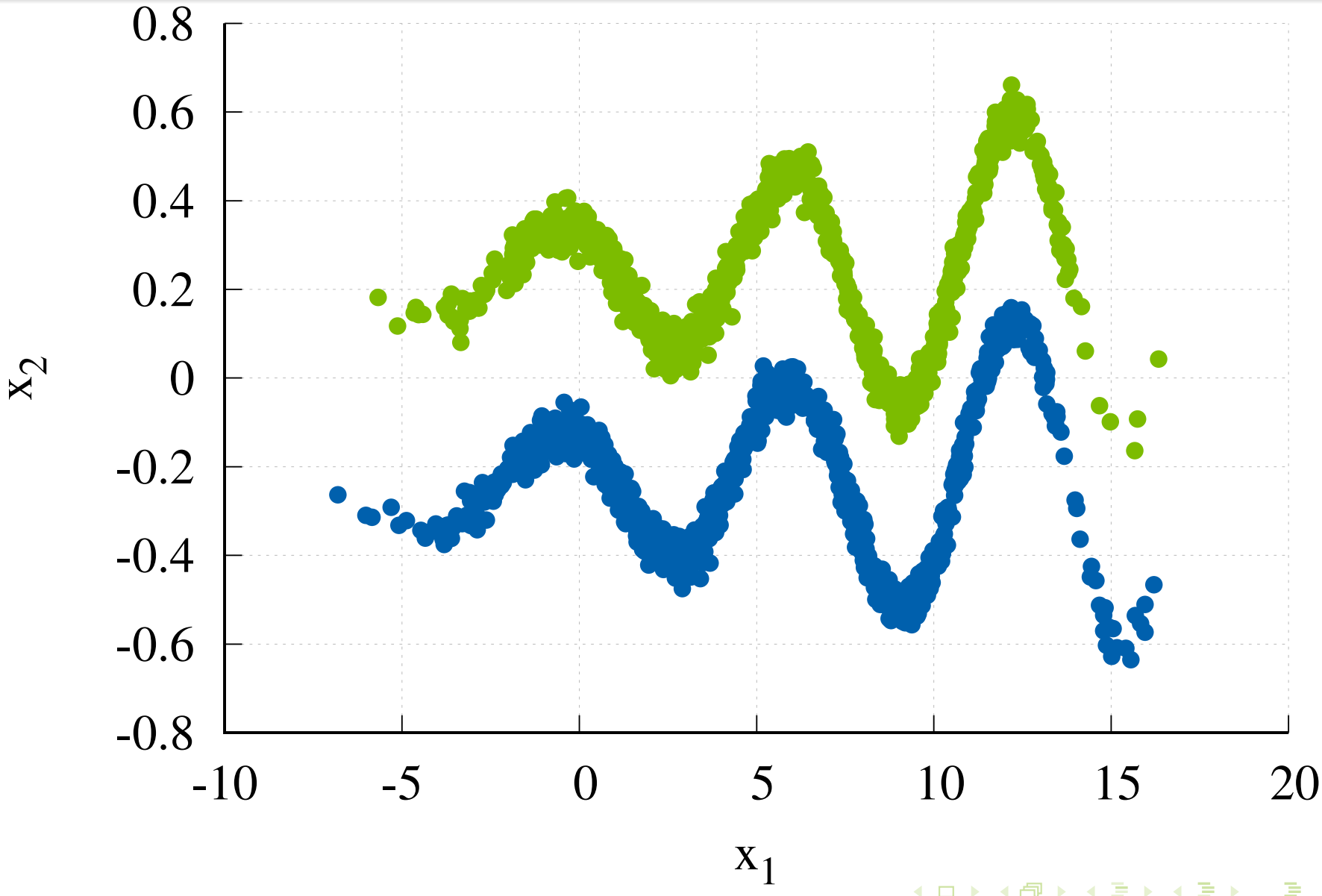
Nicht-linear trennbare Daten



Nicht-linear trennbare Daten



Nicht-linear trennbare Daten



Nicht-lineare Trennbarkeit durch Merkmalstransformation

- Trennung nicht-linearer Daten durch Transformation $\phi : \mathbb{R}^n \to \mathbb{R}^d$
- ϕ kann jede beliebige Funktion sein
- Bild von ϕ kann höher- oder niedrig-dimensionaler als $\mathcal{X}$ sein
- ϕ soll möglichst einfach zu berechnen sein (!!)
- Daten sollen im Bildraum von ϕ möglichst linear trennbar sein

Wo kommt ϕ her??

Oft: “Abstand” im neuen Raum einfacher zu berechnen als ϕ

Nicht-lineare Trennbarkeit durch Merkmalstransformation

- Trennung nicht-linearer Daten durch Transformation $\phi : \mathbb{R}^n \to \mathbb{R}^d$
- ϕ kann jede beliebige Funktion sein
- Bild von ϕ kann höher- oder niedrig-dimensionaler als $\mathcal{X}$ sein
- ϕ soll möglichst einfach zu berechnen sein (!!)
- Daten sollen im Bildraum von ϕ möglichst linear trennbar sein

Wo kommt ϕ her??

Oft: “Abstand” im neuen Raum einfacher zu berechnen als ϕ

Optimierung mit Nebenbedingungen

Lagrange-Dualität:

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$$

$$\text{s.t.} \quad g(\boldsymbol{\beta}) \geq c$$

$\equiv$

$$\min_{\boldsymbol{\beta}} \max_{\alpha} f(\boldsymbol{\beta}) - \alpha(g(\boldsymbol{\beta}) - c)$$

$$\text{s.t.} \quad \alpha \geq 0$$

Optimierung mit Nebenbedingungen

Lagrange-Dualität:

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$$

$$\text{s.t.} \quad g(\boldsymbol{\beta}) \geq c$$

$\equiv$

$$\min_{\boldsymbol{\beta}} \max_{\alpha} f(\boldsymbol{\beta}) - \alpha(g(\boldsymbol{\beta}) - c)$$

$$\text{s.t.} \quad \alpha \geq 0$$

Duales Problem und Stützvektoren

Lagrange-Dual der SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \boldsymbol{\alpha}_i - \frac{1}{2} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \sum_{i=1}^{N} \sum_{j=1}^{N} y^i y^j \left\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \right\rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i = 0$$

$$0 \leq \boldsymbol{\alpha}_i \leq C, \forall 1 \leq i \leq N$$

Neues Problem hat “nur” N Parameter (ist das gut?)

Original Gewichte: $\boldsymbol{\beta} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i \boldsymbol{x}^i$

Datenpunkte mit $\alpha_i > 0$ heißen Stützvektor (Support-Vector)

Duales Problem und Stützvektoren

Lagrange-Dual der SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \boldsymbol{\alpha}_i - \frac{1}{2} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \sum_{i=1}^{N} \sum_{j=1}^{N} y^i y^j \left\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \right\rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i = 0$$

$$0 \leq \boldsymbol{\alpha}_i \leq C, \forall 1 \leq i \leq N$$

Neues Problem hat “nur” N Parameter (ist das gut?)

Original Gewichte: $\boldsymbol{\beta} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i \boldsymbol{x}^i$

Datenpunkte mit $\alpha_i > 0$ heißen Stützvektor (Support-Vector)

Duales Problem und Stützvektoren

Lagrange-Dual der SVM:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \boldsymbol{\alpha}_i - \frac{1}{2} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \sum_{i=1}^{N} \sum_{j=1}^{N} y^i y^j \left\langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \right\rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i = 0$$

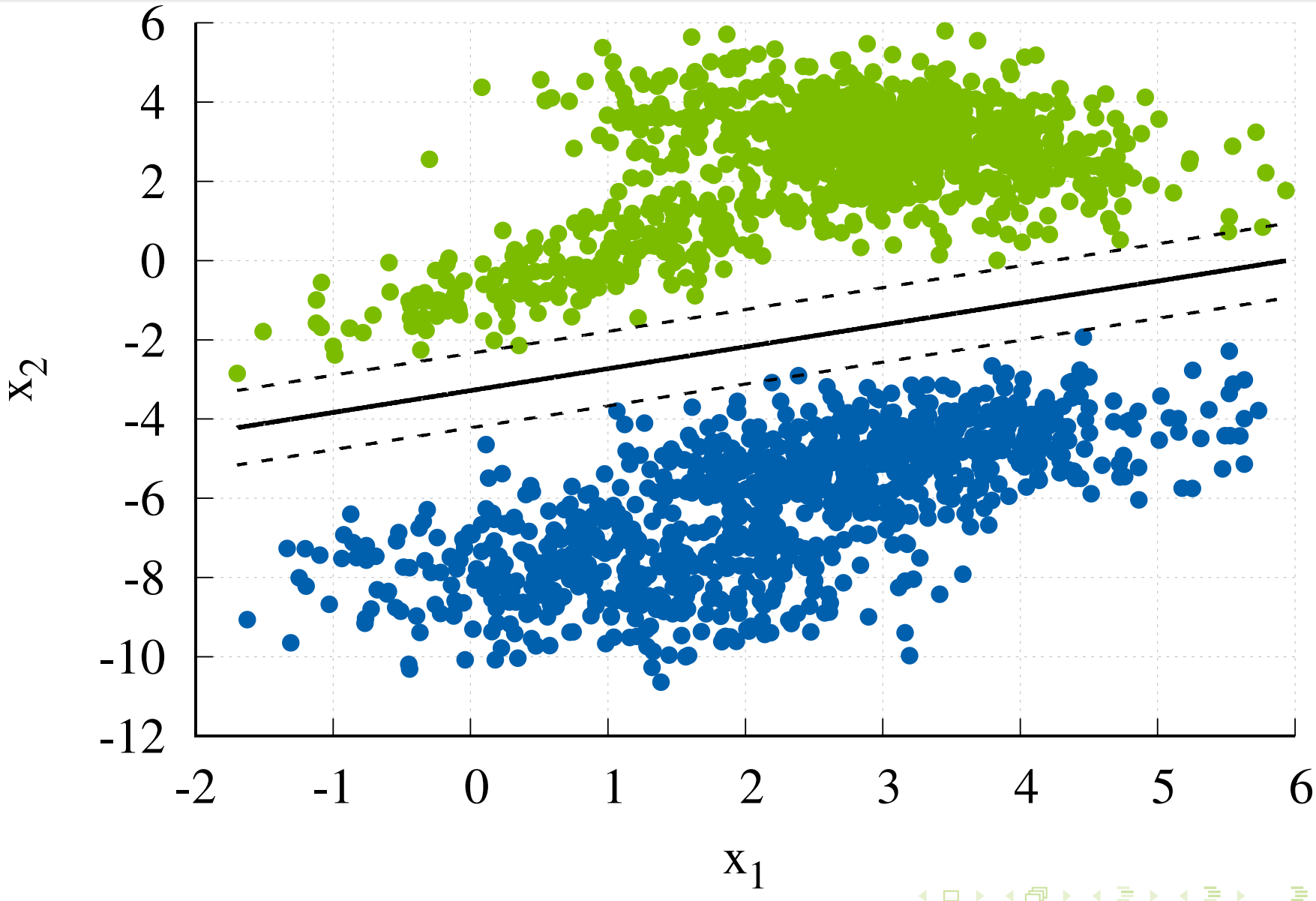
$$0 \leq \boldsymbol{\alpha}_i \leq C, \forall 1 \leq i \leq N$$

Neues Problem hat “nur” N Parameter (ist das gut?)

Original Gewichte: $\boldsymbol{\beta} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i y^i \boldsymbol{x}^i$

Datenpunkte mit $\boldsymbol{\alpha}_i > 0$ heißen Stützvektor (Support-Vector)

Klassifikation mit Hyperebenen



Kernel “Trick”

Anstatt $\phi(\boldsymbol{x})$ für jeden $\boldsymbol{x}$ auszurechnen,

- Finde Funktion K, mit $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$
- K heißt *Kernel* oder *Kernfunktion*
- $\phi(\boldsymbol{x})$ (und $\phi(\boldsymbol{x}')$) muss nie explizit berechnet werden!
- ϕ muss nicht bekannt sein!!
- $K_{\gamma,\delta,\ldots}(x, x')$ kann Parameter haben

Kernel “Trick”

Anstatt $\phi(\boldsymbol{x})$ für jeden $\boldsymbol{x}$ auszurechnen,

- Finde Funktion K, mit $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$
- K heißt *Kernel* oder *Kernfunktion*
- $\phi(\boldsymbol{x})$ (und $\phi(\boldsymbol{x}')$) muss nie explizit berechnet werden!
- ϕ muss nicht bekannt sein!!
- $K_{\gamma,\delta,\ldots}(\boldsymbol{x}, \boldsymbol{x}')$ kann Parameter haben

Kernfunktionen

$$K_{\text{Linear}}(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle$$

$$K_{\text{Poly},k,c}(\boldsymbol{x}, \boldsymbol{x}') = \left(\left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle + c\right)^k$$

$$K_{\text{Gauss},\gamma}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{\gamma}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2\right)$$

- Graph Kernel (common subgraphs, Bäume, Pfade)
- String Kernel (gemeinsame Zeichenketten, gemeinsame Parsebäume)