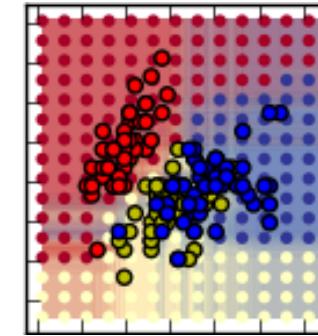
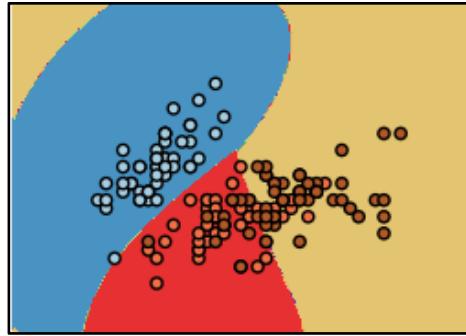


Bildanalyse und Visualisierung in Diagnostik und Therapie

Support Vector Machines & Random Decision Forests



M.Sc. Oskar Maier

Prof. Dr. rer. nat. habil. Heinz Handels

Institut für Medizinische Informatik

Universität zu Lübeck

Inhalt

- Klassifizierung
- Support Vector Machines (SVM)
 - Maximum-Margin Klassifizierer
 - Nicht-lineare Probleme
 - Kernel-Trick
 - Soft-Margin Klassifizierer
- Entscheidungsbäume
 - Training und Anwendung
 - Informationsgewinn als Gütemaß
- Random Decision Forests (RDF)
 - Mehr ist besser
 - Das Random in Random Decision Forests
- Anwendungsbeispiele



Klassifizierung

Definitionen

- **Klassifizierung:** „Die Einordnung einer neuen Observation in eine von mehreren Kategorien anhand ihrer Merkmale“
- **Merkmale:** individuelle, messbare Eigenschaften einer Observation
 - Unterscheidend
 - Unabhängig
- **Klassifizierer:** Ein auf einer Trainingsmenge erlerntes Problemlösungsverfahren
- **Trainingsmenge:** Eine Menge von Stichproben-Observationen mit bekannten Klassenzugehörigkeiten zum Trainieren eines Klassifizierers
- Im Maschinellen Lernen: **supervised training**

Ein Beispiel

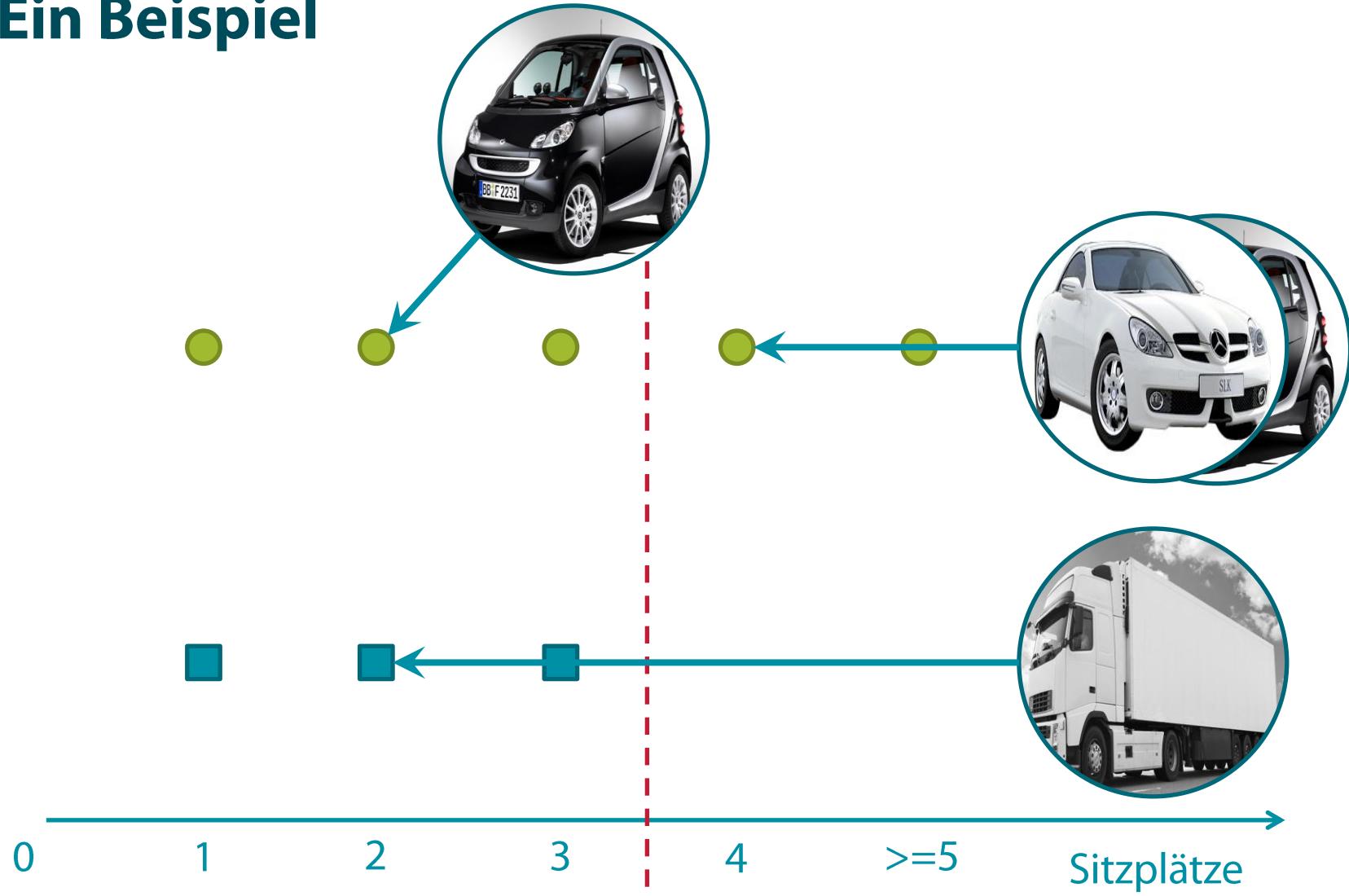


LKW

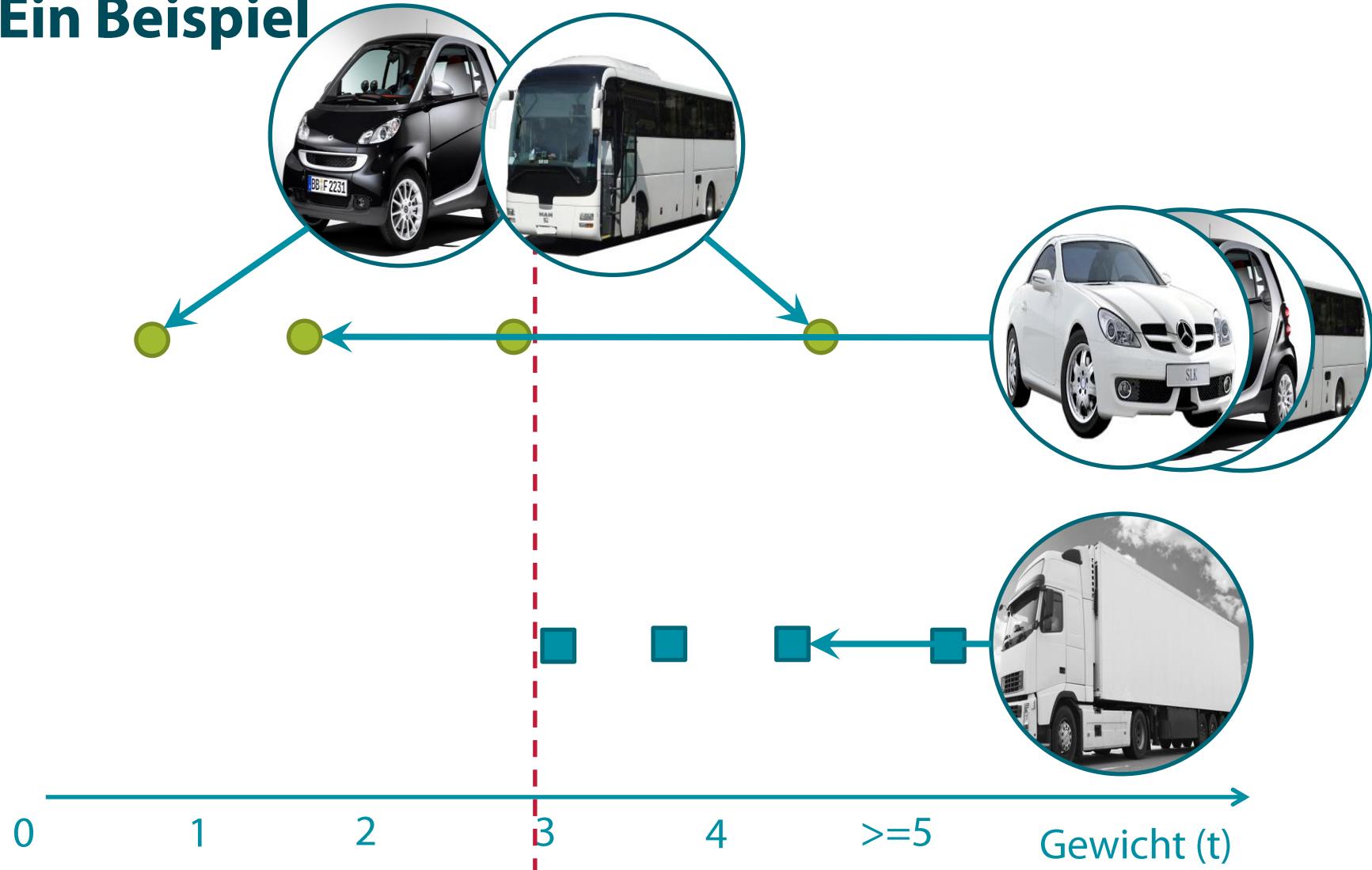


**Andere
Kraftfahrzeuge**

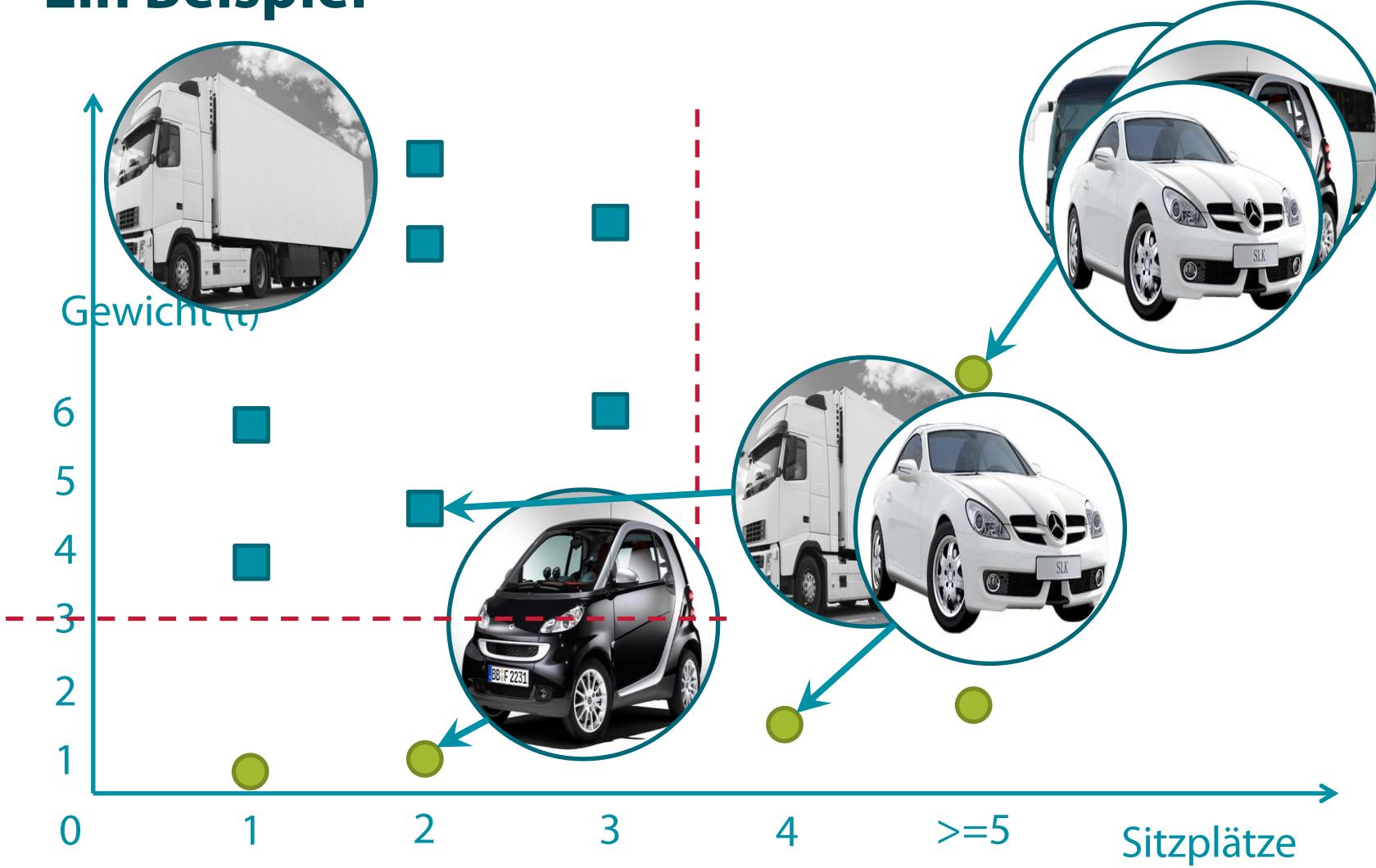
Ein Beispiel



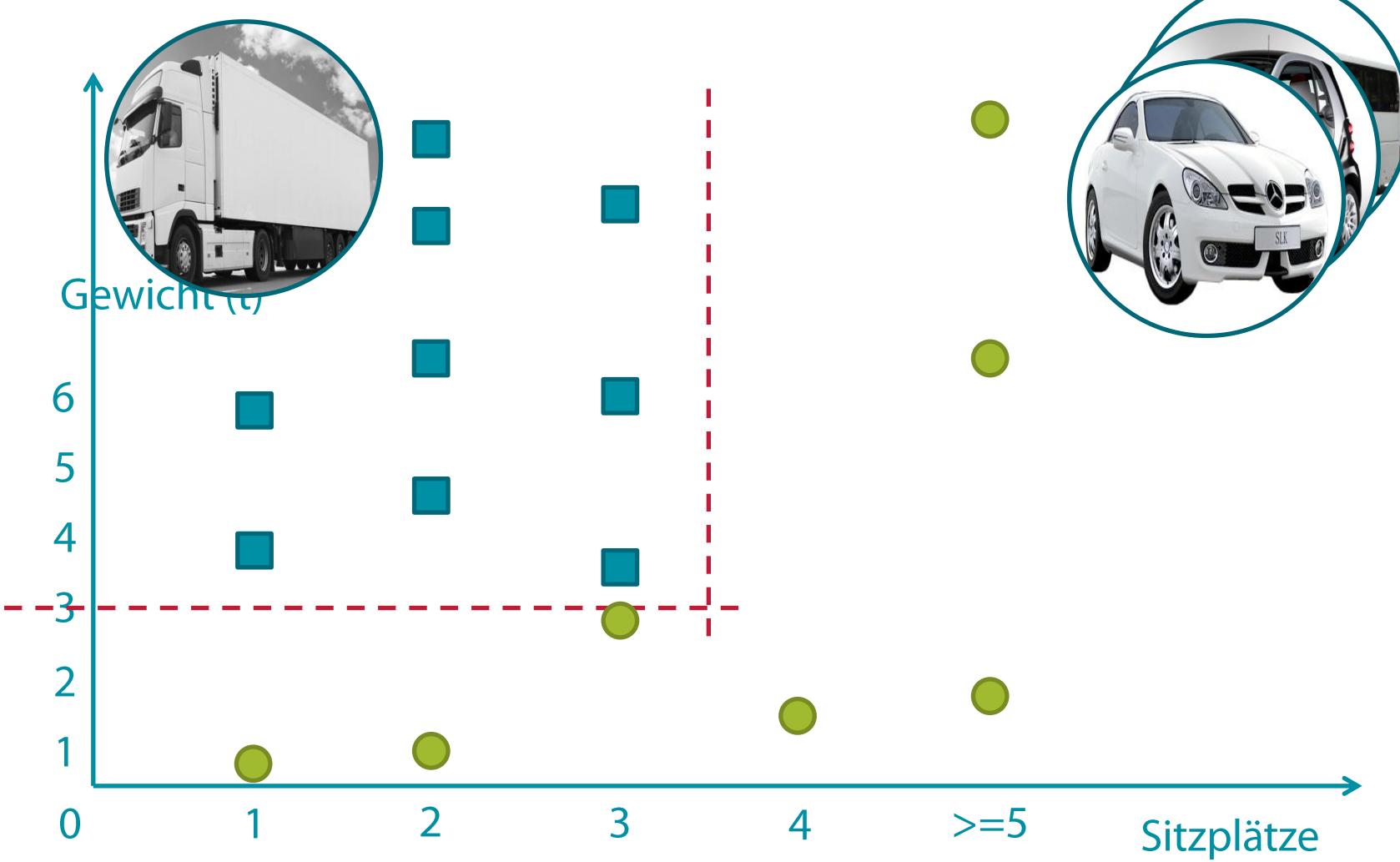
Ein Beispiel



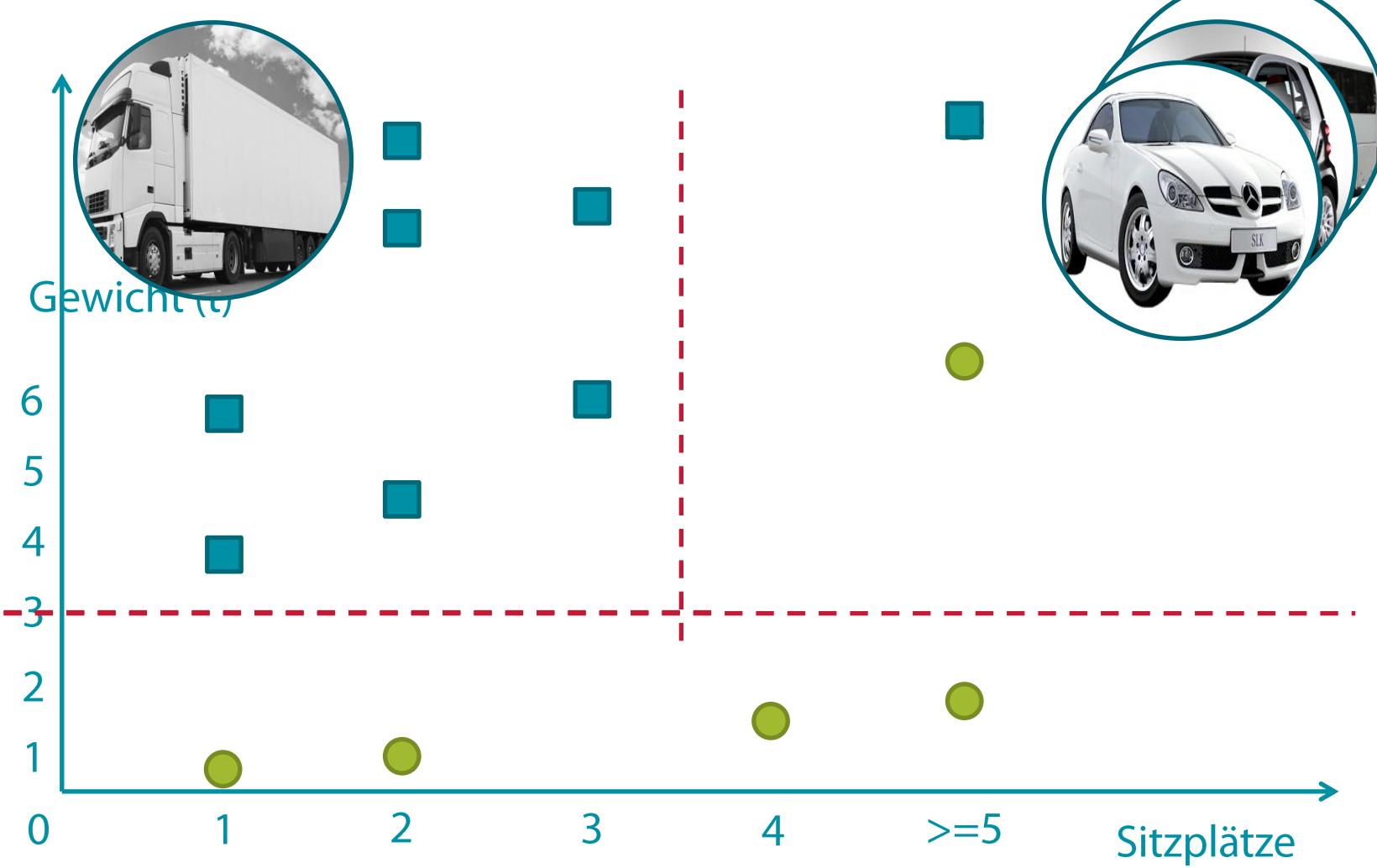
Ein Beispiel



Ein Beispiel - Anwendung



Ein Beispiel – Trainingsset



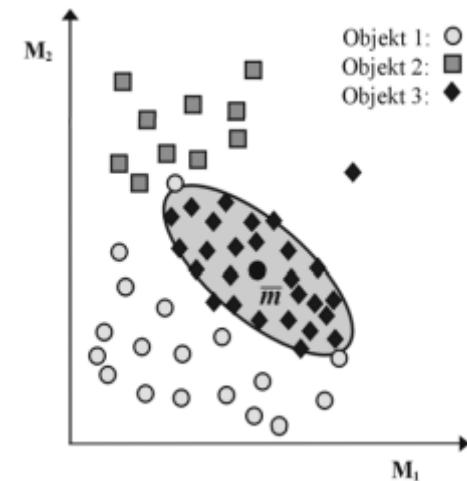
Formales Vorgehen

1. Klassifizierungsproblem bestimmen
2. Menge von Stichproben-Observationen mit zugehöriger Klasse sammeln => Trainingsmenge
 - Repräsentativ?
 - Verrauscht?
 - Nicht zu wenig (repräsentieren Modell nicht)
 - Evaluationsmenge zu Seite legen
3. Geeignete Merkmale extrahieren
 - Unterscheidend und unabhängig?
 - Nicht zu viele (Fluch der Dimensionalität)
 - Nicht zu wenig (keine Abgrenzung möglich)
4. Geeigneten Klassifizierer auswählen
 - Wie sieht der Problemraum aus?
5. Klassifizierer anlernen
6. Genauigkeit und Generalisierung des Klassifizierers evaluieren

Rückblick

„Algorithmus zur ROI-basierten Pixelklassifikation in multi-spektralen Bilddaten“, 2te Vorlesung

1. Auswahl der ROI => Trainingsdaten
2. Modell lernen => Klassifizierer anlernen
(Stichwort: Mahalanobis-Distanz / CovM)
3. Verbleibende Voxel klassifizieren =>
Klassifizierer anwenden



Unsupervised learning

- Stichwort: **k-means clustering**
- Beim unsupervised learning sind die Klassenzugehörigkeiten (kurz: Klassen) nicht bekannt
- Zusammengehörigkeit wird über die (meist euklidische) Nähe der Merkmale im Merkmalsraum definiert: Mustererkennung



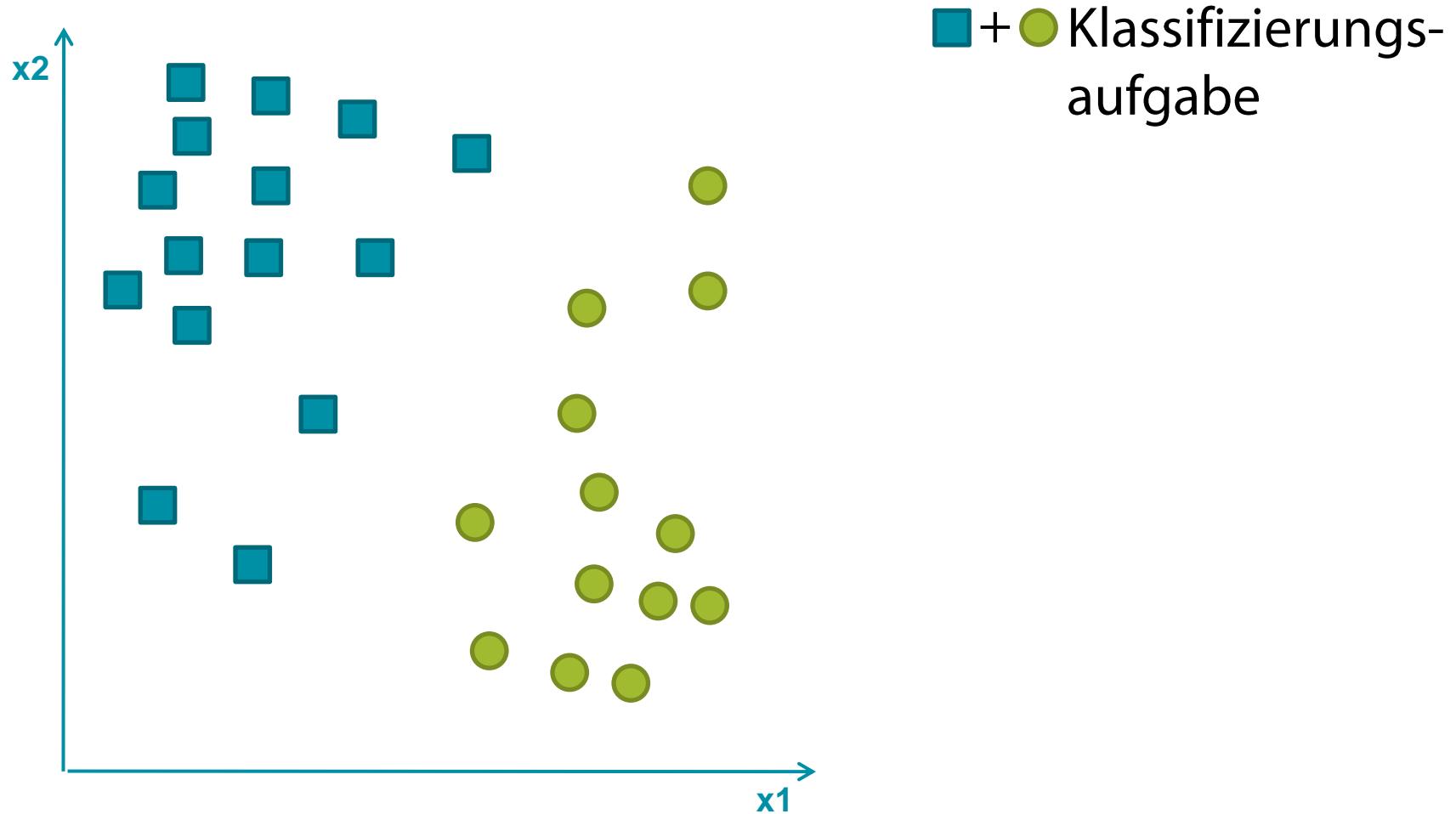
Klassifizierung

Q&A

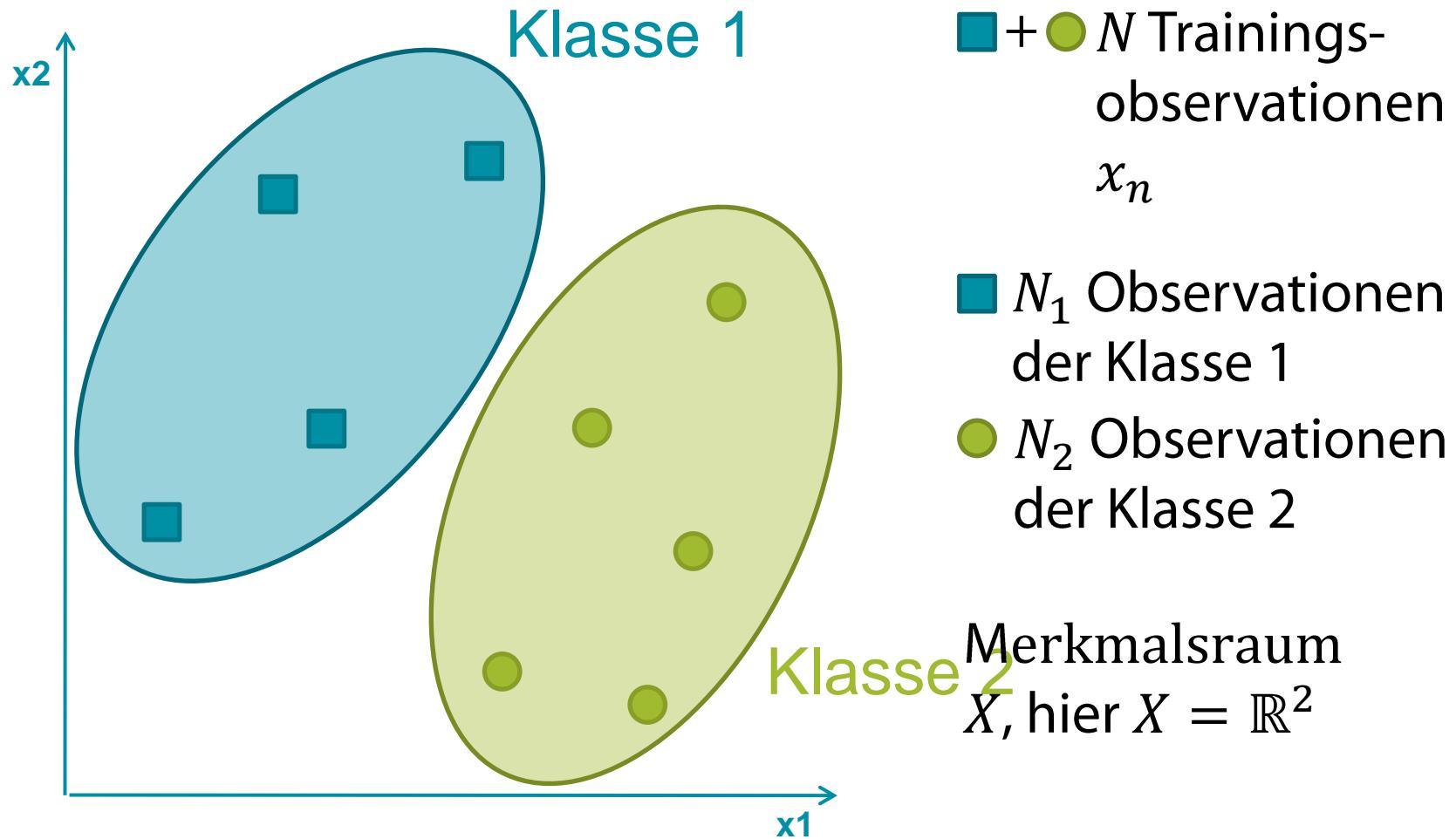


Support Vector Machines (**SVM**)

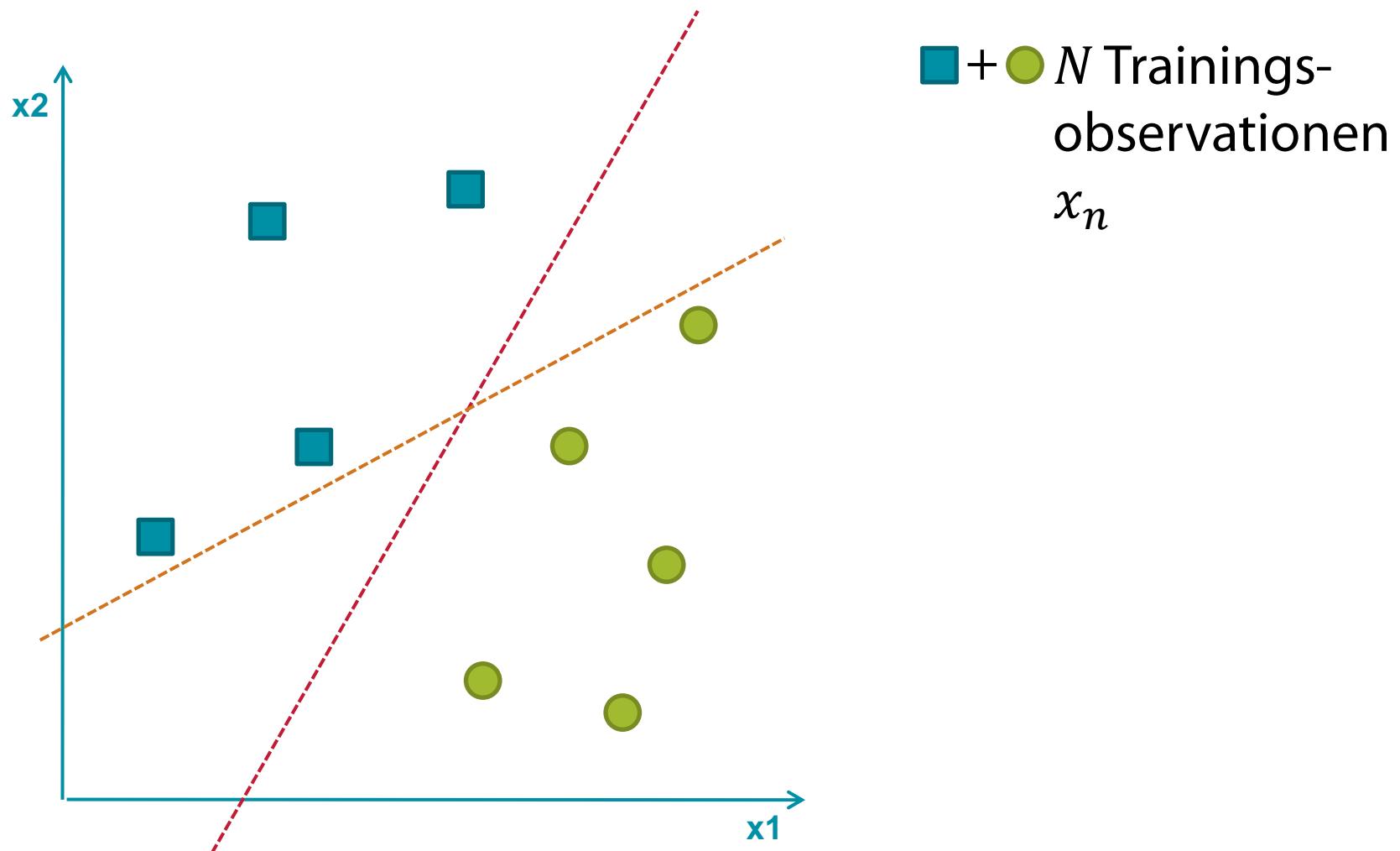
Support Vector Machines (SVM)



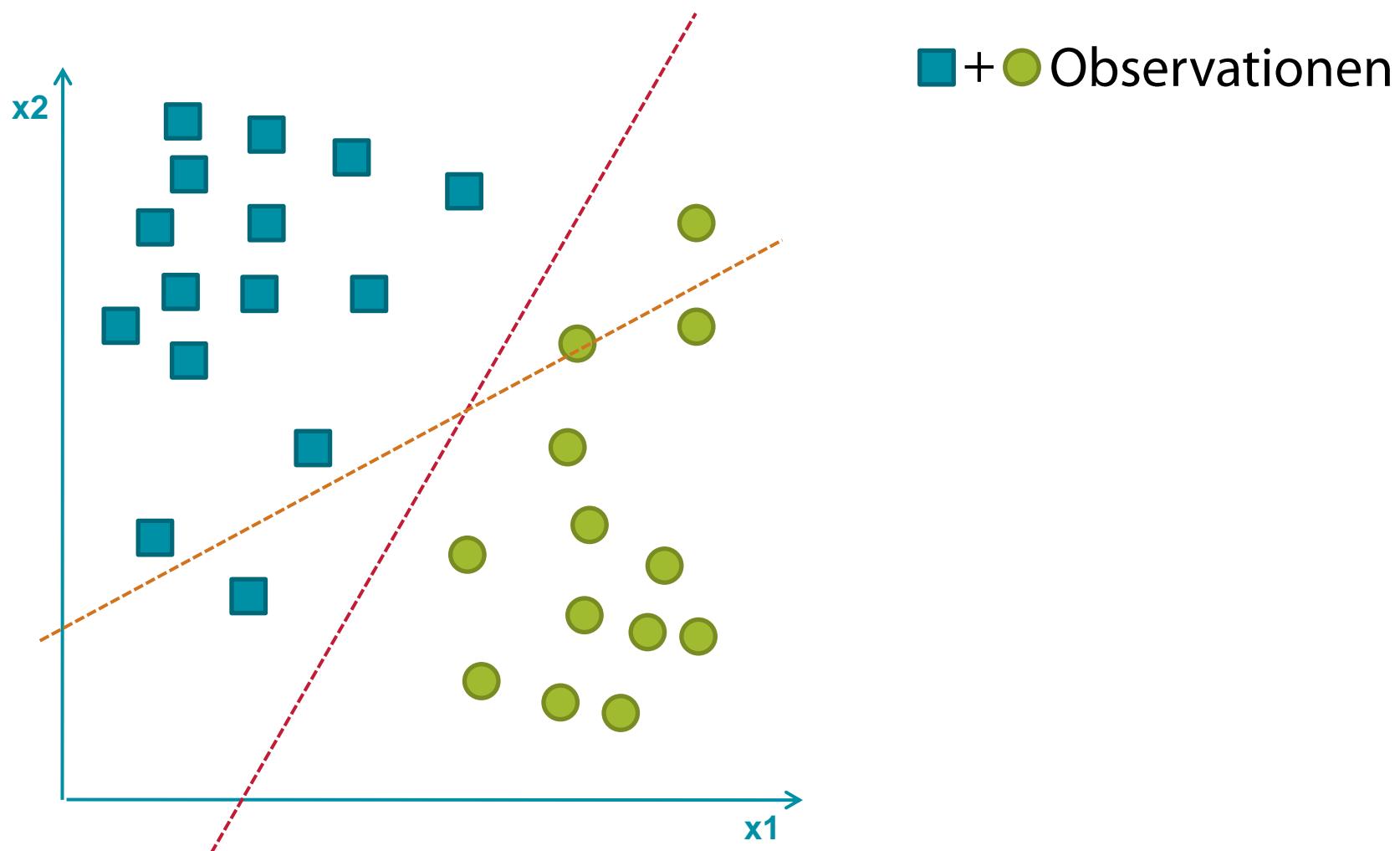
Support Vector Machines (SVM)



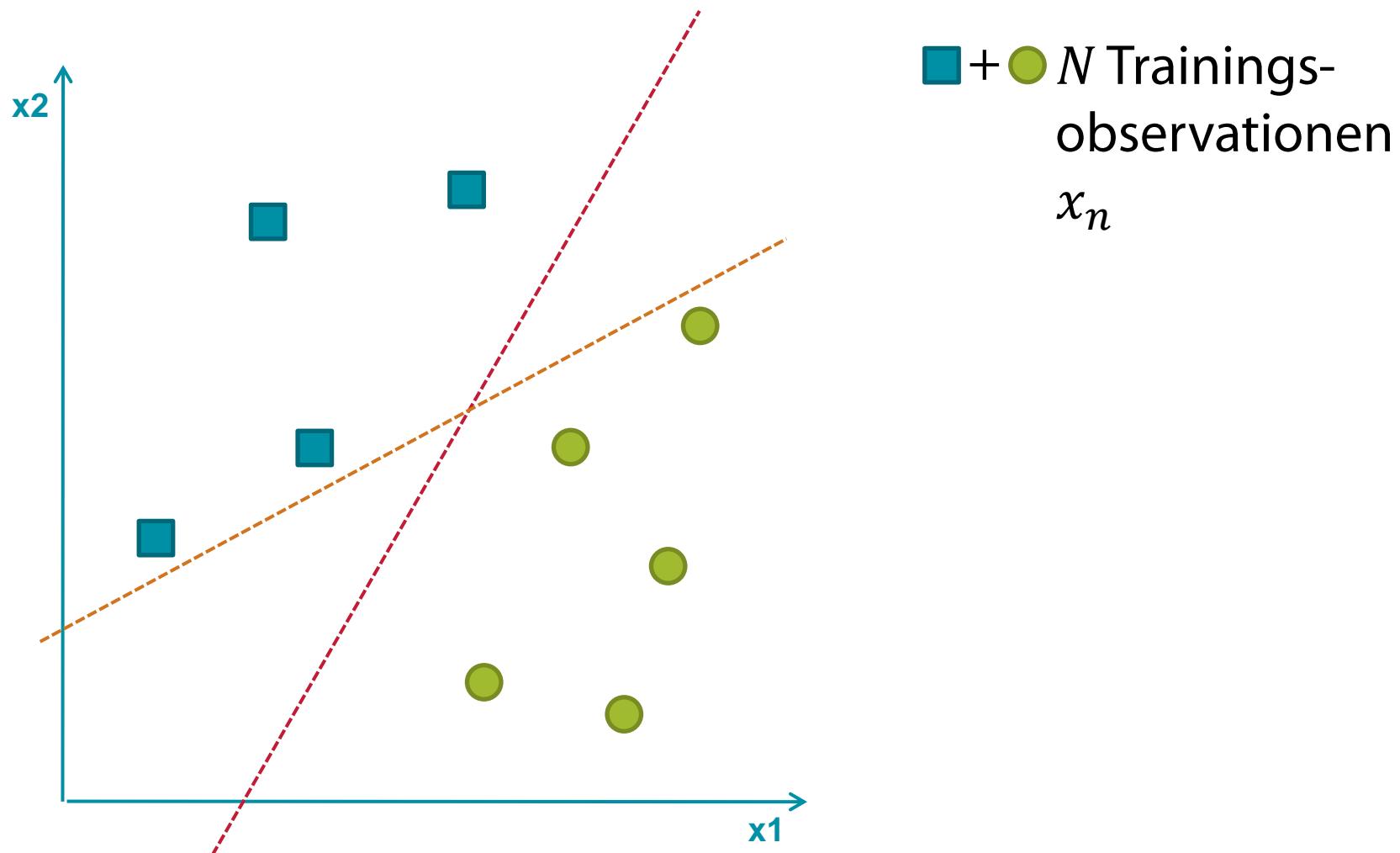
SVM: Partitionieren der Observationen



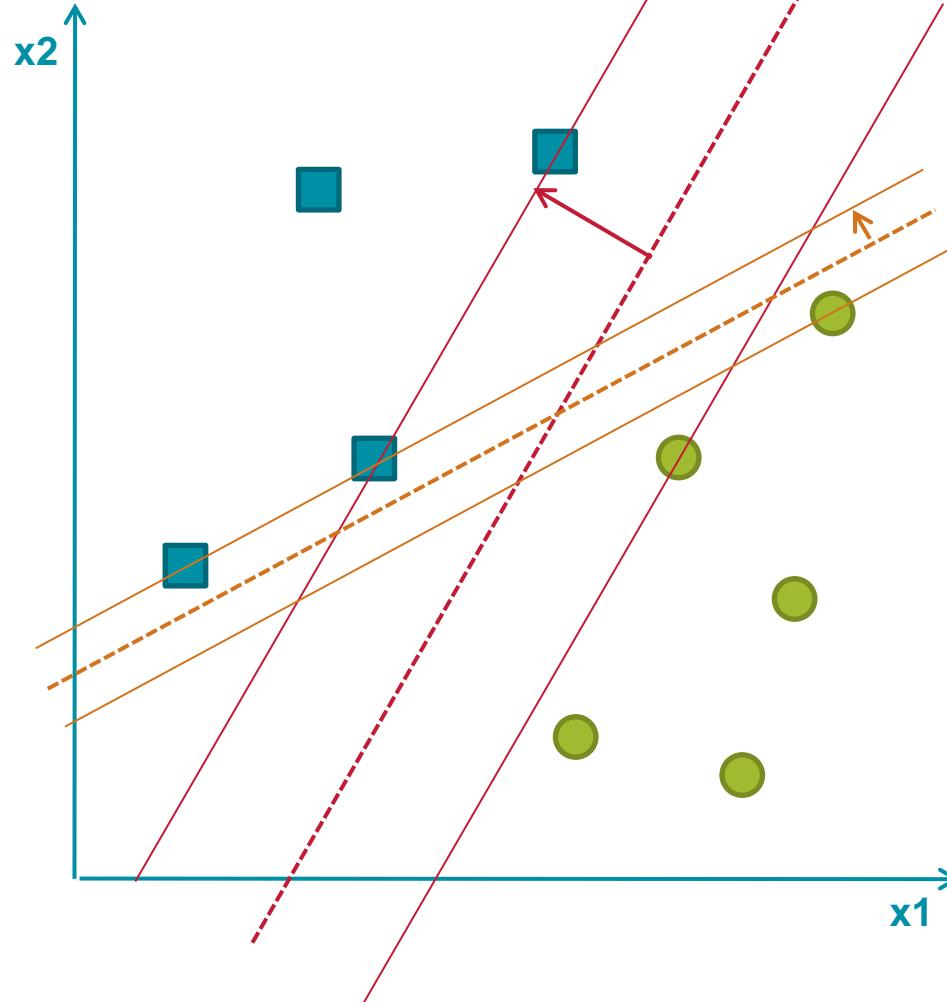
SVM: Partitionieren der Observationen



SVM: Partitionieren der Observationen

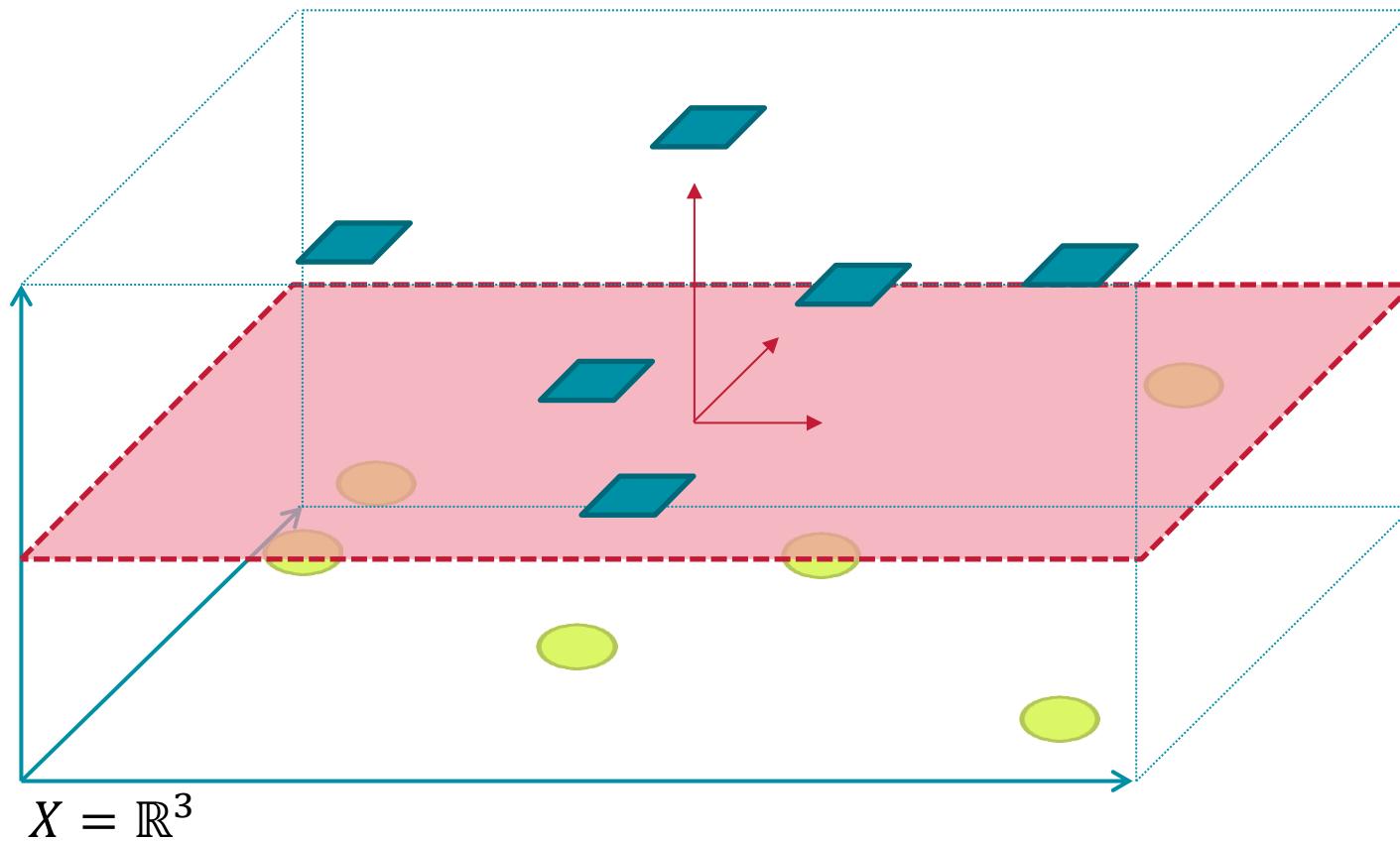


SVM: Maximum Margin Klassifikator



■ + ● N Trainings-
observationen
 x_n

SVM: Die partitionierende Hyperebene



SVM

Wir suchen die Hyperebene, die

1. die Trainingsdaten perfekt trennt und
2. den maximalen Abstand zu den nächsten Trainingsobservationen beider Klassen aufweist

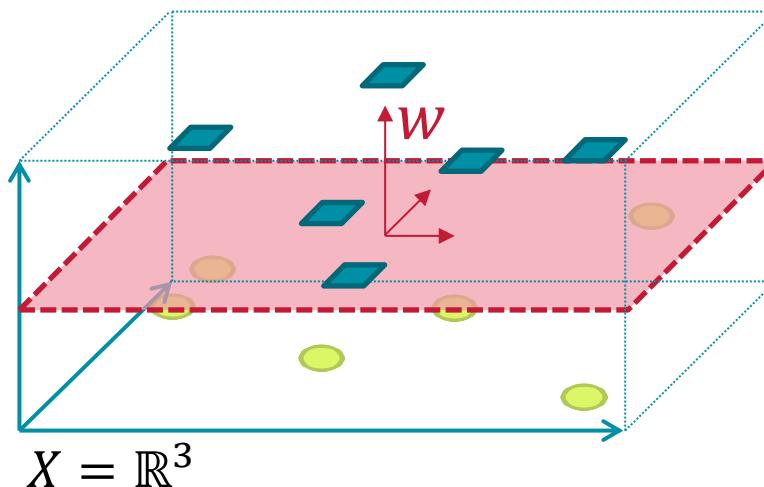
Wir benötigen für eine rechnerische Lösung:

- Mathematische Beschreibung der Hyperebene
- Mathematische Beschreibung des Abstandes
- Mathematische Beschreibung des Optimierungsproblems

SVM: Die partitionierende Hyperebene

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

(Normalenform)



$$b \in \mathbb{R}$$

~ Schwellenwert

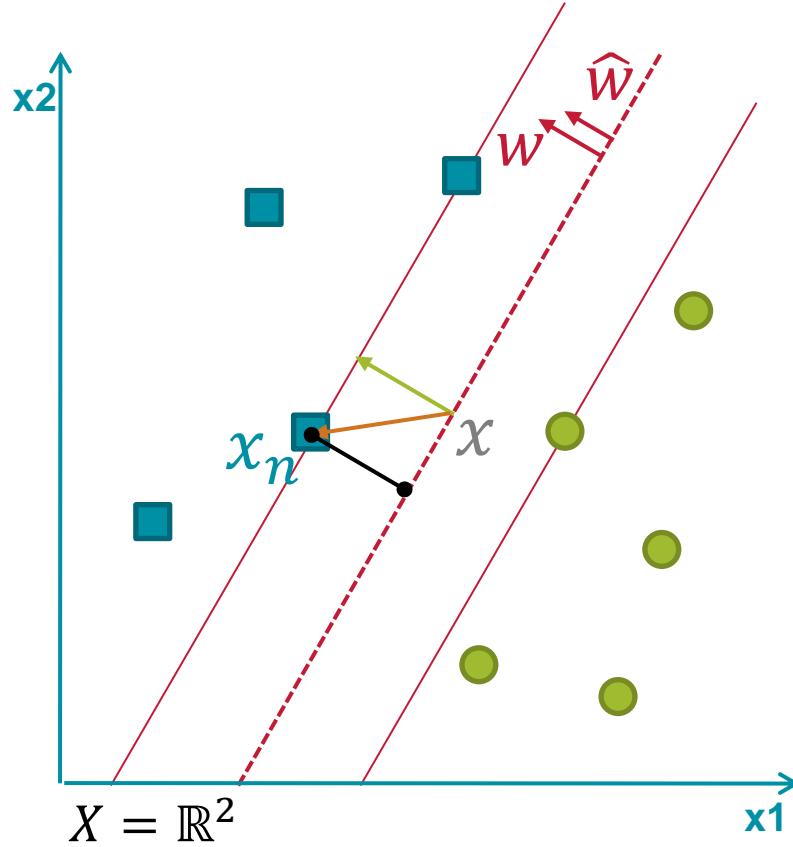
$$\mathbf{w} \in \mathcal{X}$$

Ausrichtung der Ebene
Skalierung beliebig

$$\mathbf{w}, b$$

Unbekannte

SVM: Definition des Abstandes



Observation

x_n

Ebene

$$w^T x + b = 0$$

Einheitsvektor

$$\hat{w} = \frac{w}{\|w\|}$$

Differenzvektor

$$x_n - x$$

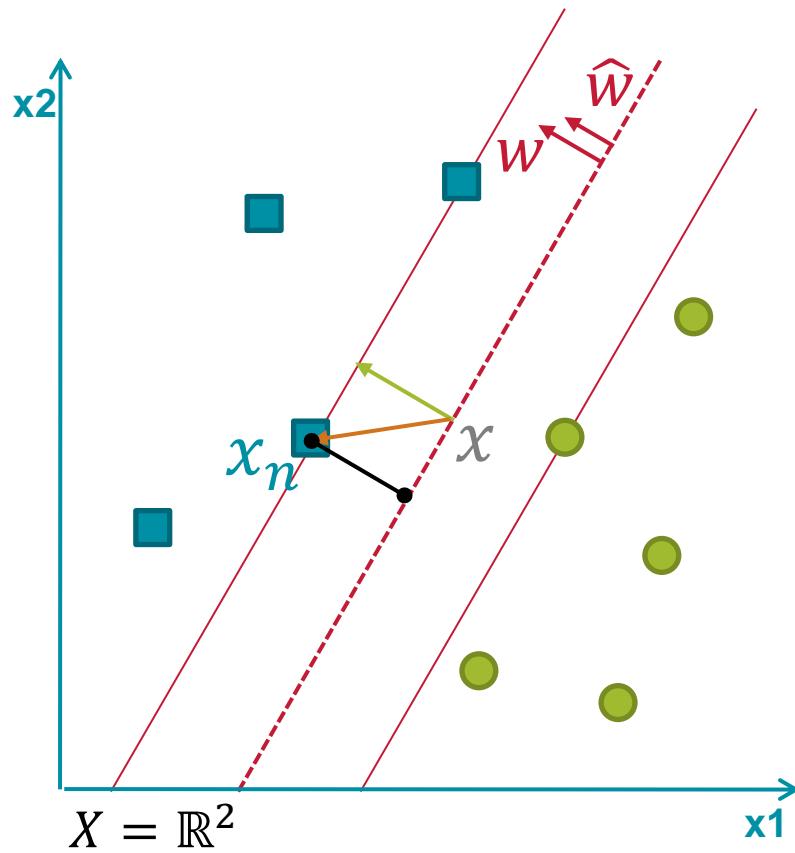
Projektion

$$\hat{w}^T(x_n - x)$$

Abstand

$$|\hat{w}^T(x_n - x)|$$

SVM: Definition des Abstandes



Abstand

$$|\hat{w}^T(x_n - x)|$$

Einheitsvektor

$$\hat{w} = \frac{w}{\|w\|}$$

Abstand

$$\frac{1}{\|w\|} |w^T x_n - w^T x|$$

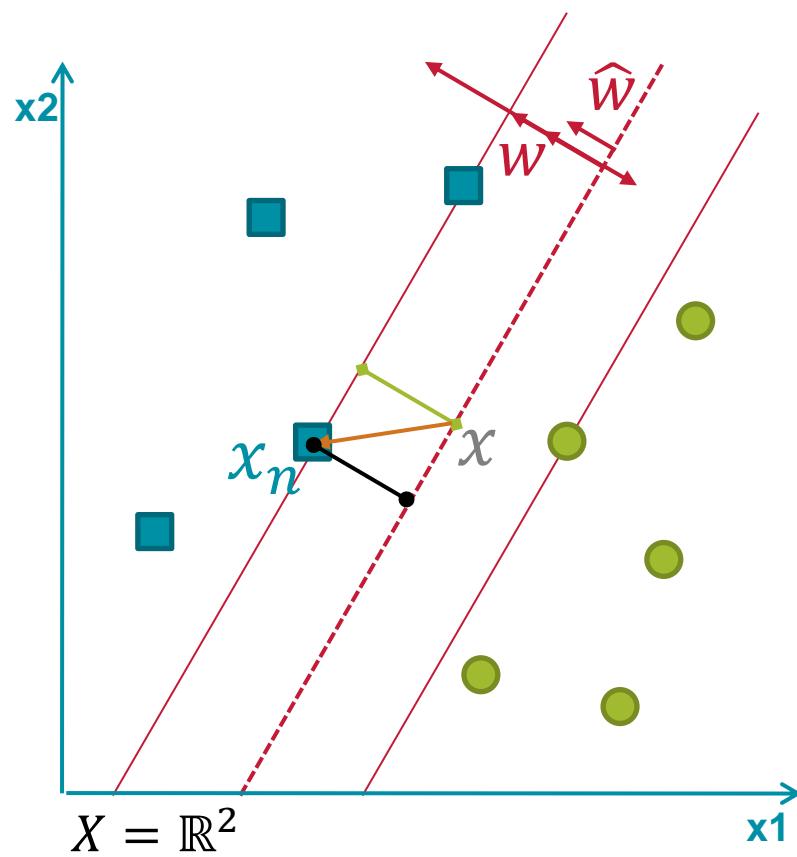
Ebene

$$w^T x + b = 0$$

Abstand

$$\frac{1}{\|w\|} |w^T x_n + b - w^T x + b|$$

SVM: Definition des Abstandes



Abstand

$$\frac{1}{\|w\|} |w^T x_n + b - w^T x - b|$$

Ebene

$$w^T x + b = 0$$

Abstand

$$\frac{1}{\|w\|} |w^T x_n + b - 0|$$

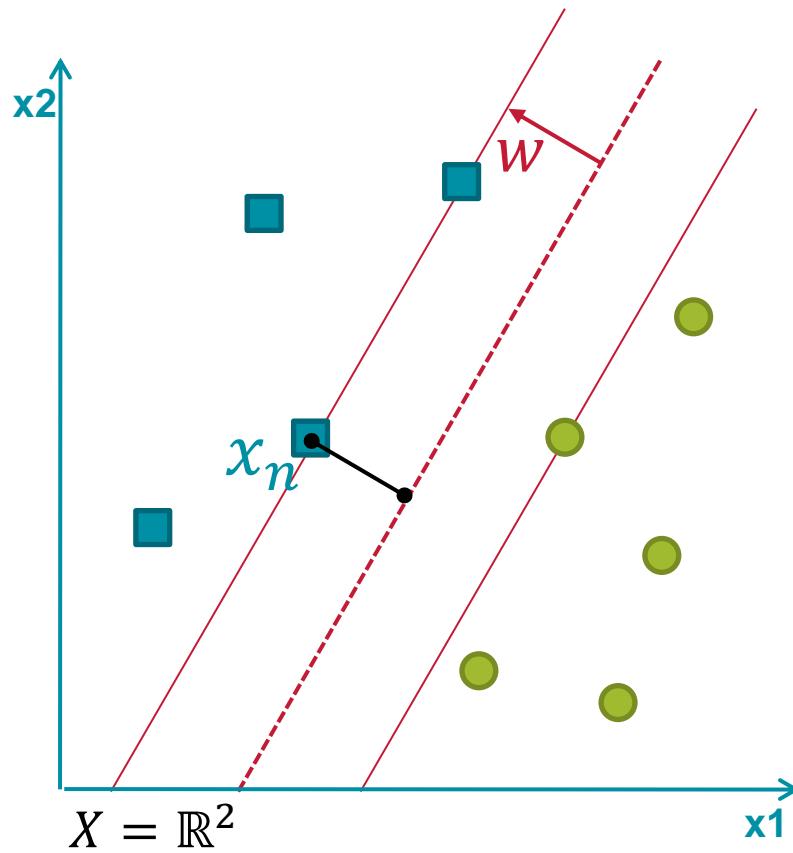
Festlegung (für nächstes x_n)

$$w^T x_n + b = 1$$

Abstand

$$\frac{1}{\|w\|} |1 - 0| = \frac{1}{\|w\|}$$

SVM: Das Optimierungsproblem



Abstand / Margin maximieren

$$\max \frac{1}{\|w\|}$$

Randbedingung

$$|w^T x_n + b| \geq 1$$

wobei $n = 1, \dots, N$

SVM: Das Optimierungsproblem

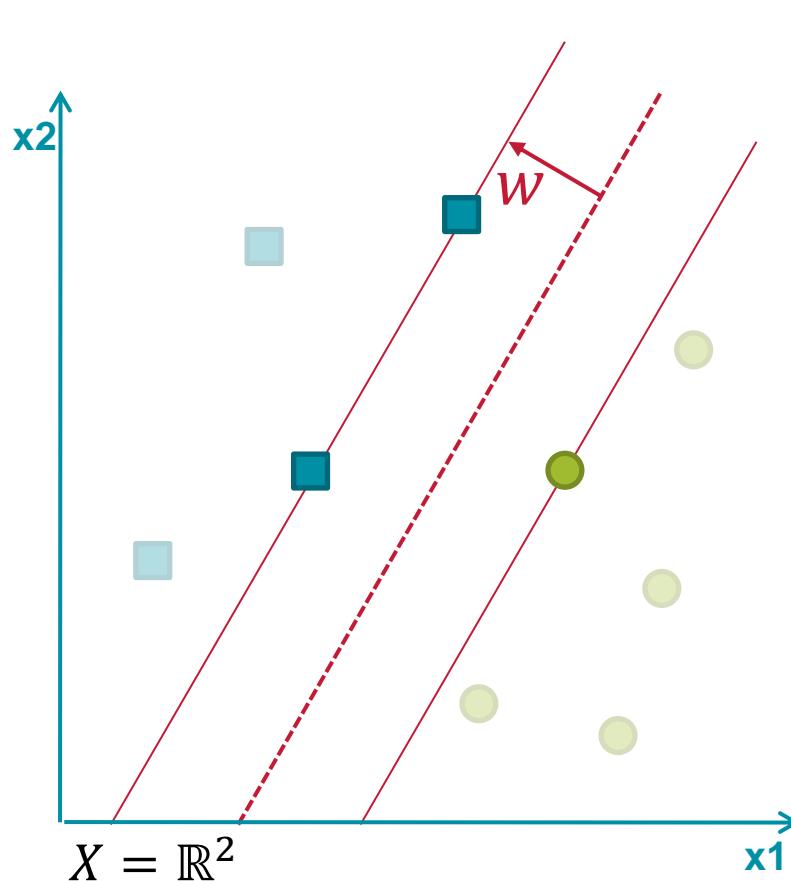
- Karush-Kuhn-Tucker (KKT) Bedingungen: *ineq. -> equality*
- Lagrangian Multipliers: *Maximierung mit Randbedingungen*
- Quadratic Programming: *Lösung des Optimierungsproblems*

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N y_n y_m \alpha_n \alpha_m \mathbf{x}_n^T \mathbf{x}_m - \sum_{n=1}^N \alpha_n$$

- Komplexität: $\mathcal{O}(N^2)$, bis auf $\mathbf{x}_n^T \mathbf{x}_m$ unabhängig von \mathcal{X}

Ergebnis des Ganzen: Support Vectors / Stützvektoren

SVM: Support Vectors



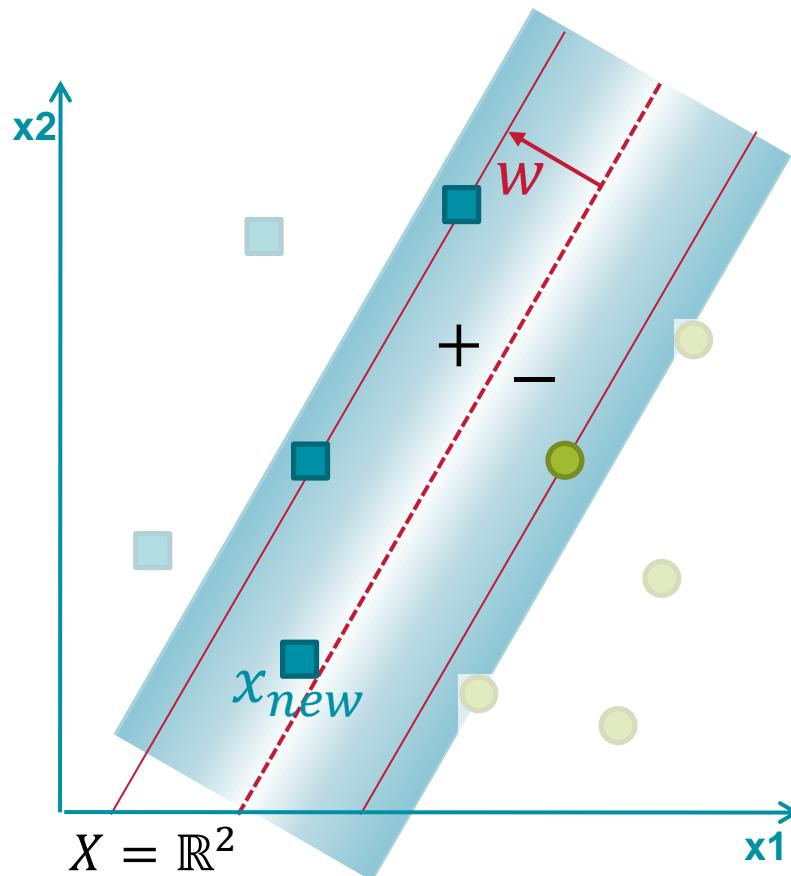
■ + ● Support Vectors

Partitionierende Hyperbene

$$w^T x + b = 0$$

w und b können berechnet werden

SVM: Der Klassifikator



■ + ● Support Vectors

Partitionieren Hyperbene

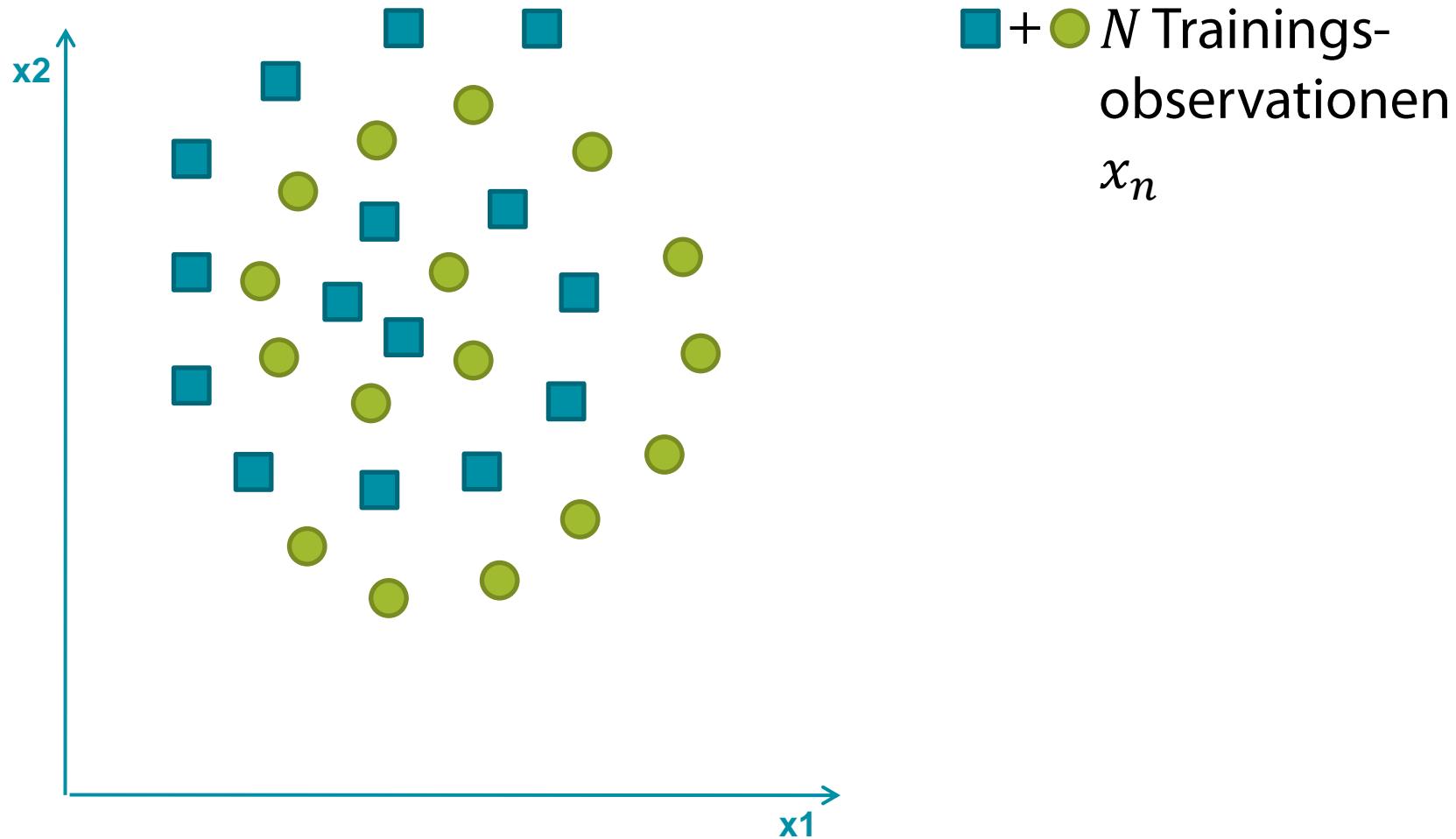
$$\mathbf{w}^T \mathbf{x} + b = 0$$

\mathbf{w} und b können einfach berechnet werden

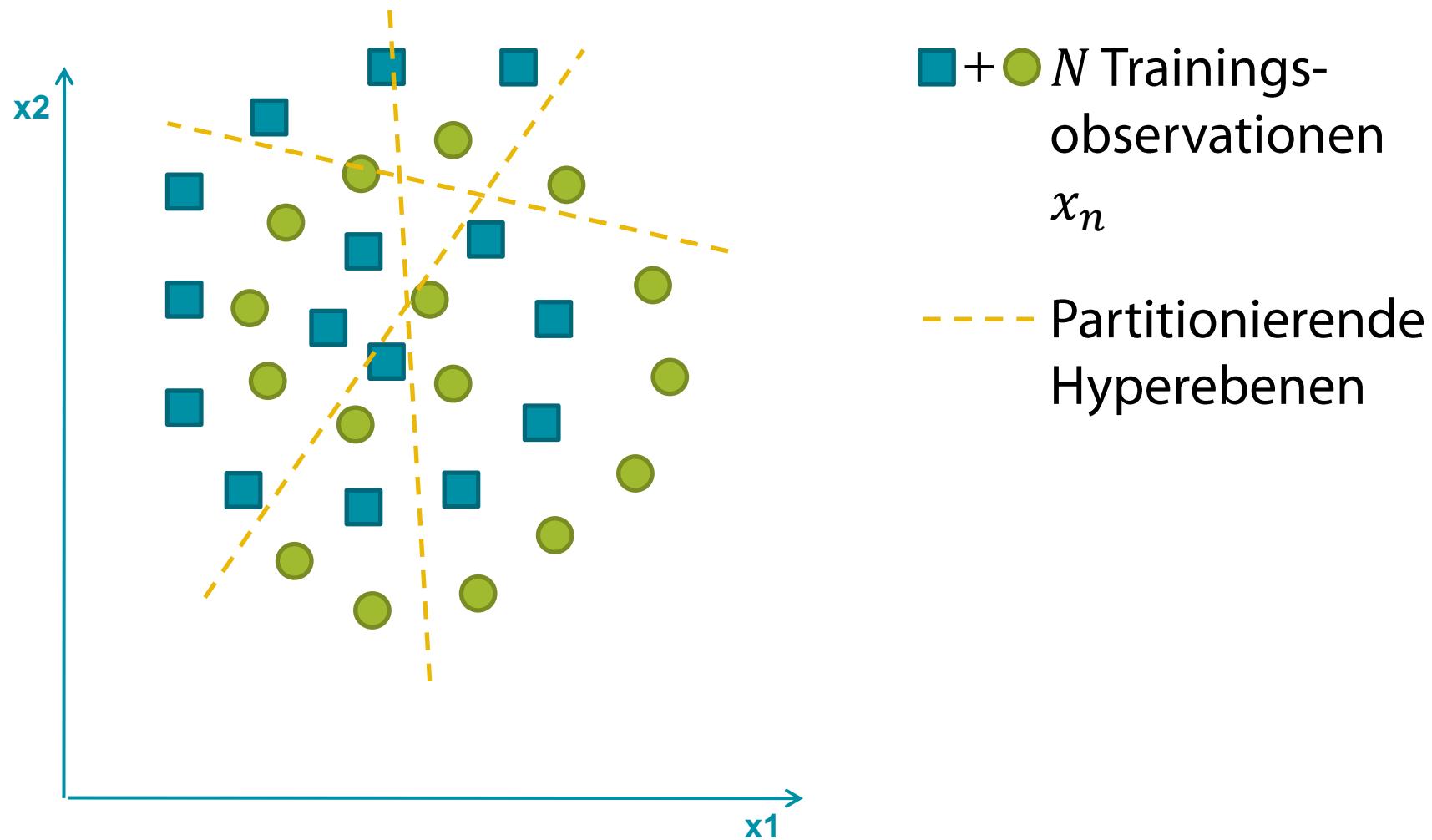
Neue Observation x_{new}

$$\text{sgn}(\mathbf{w}^T \mathbf{x}_{new} + b)$$

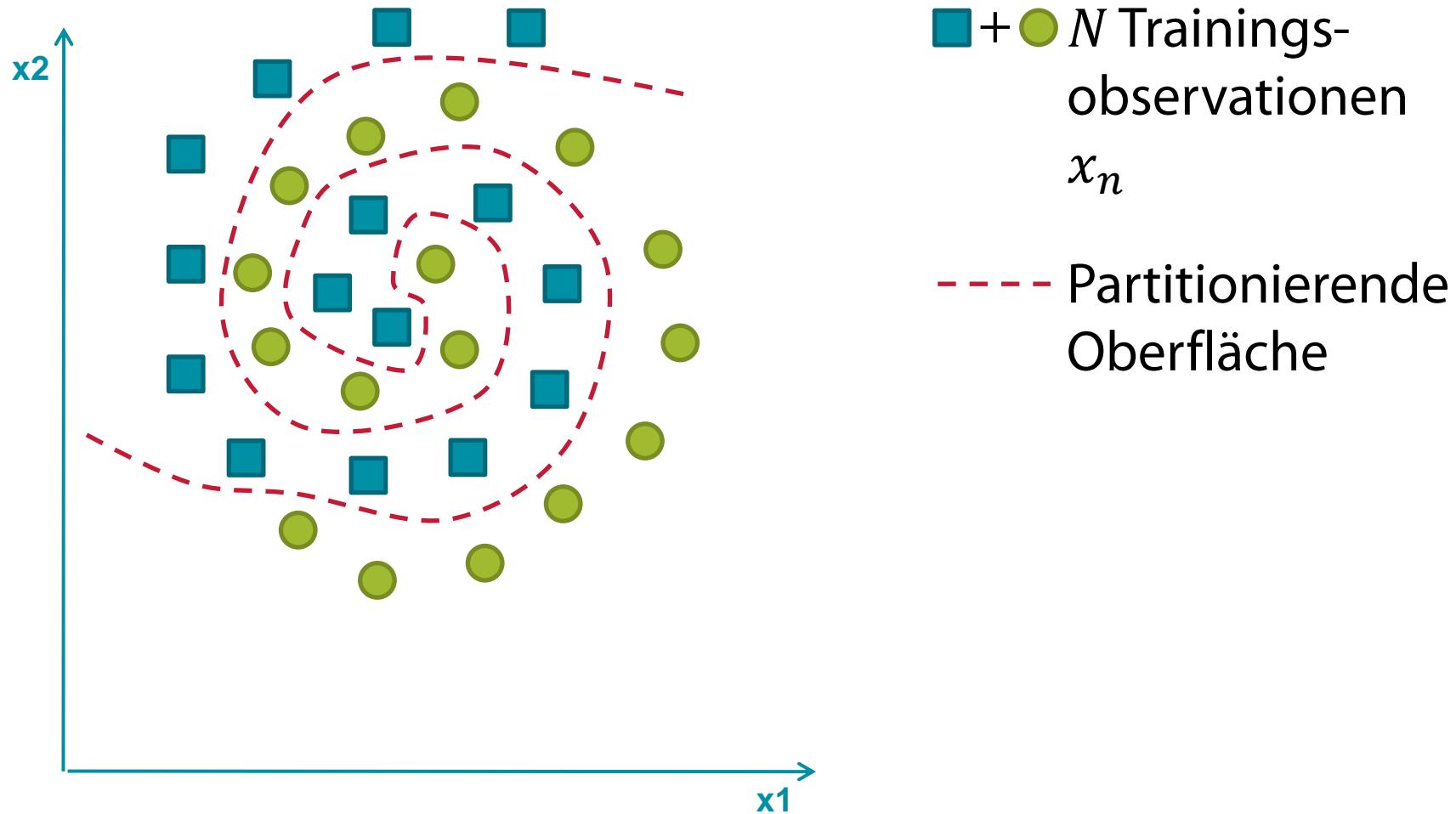
SVM: Nicht-lineare Klassifizierungsprobleme



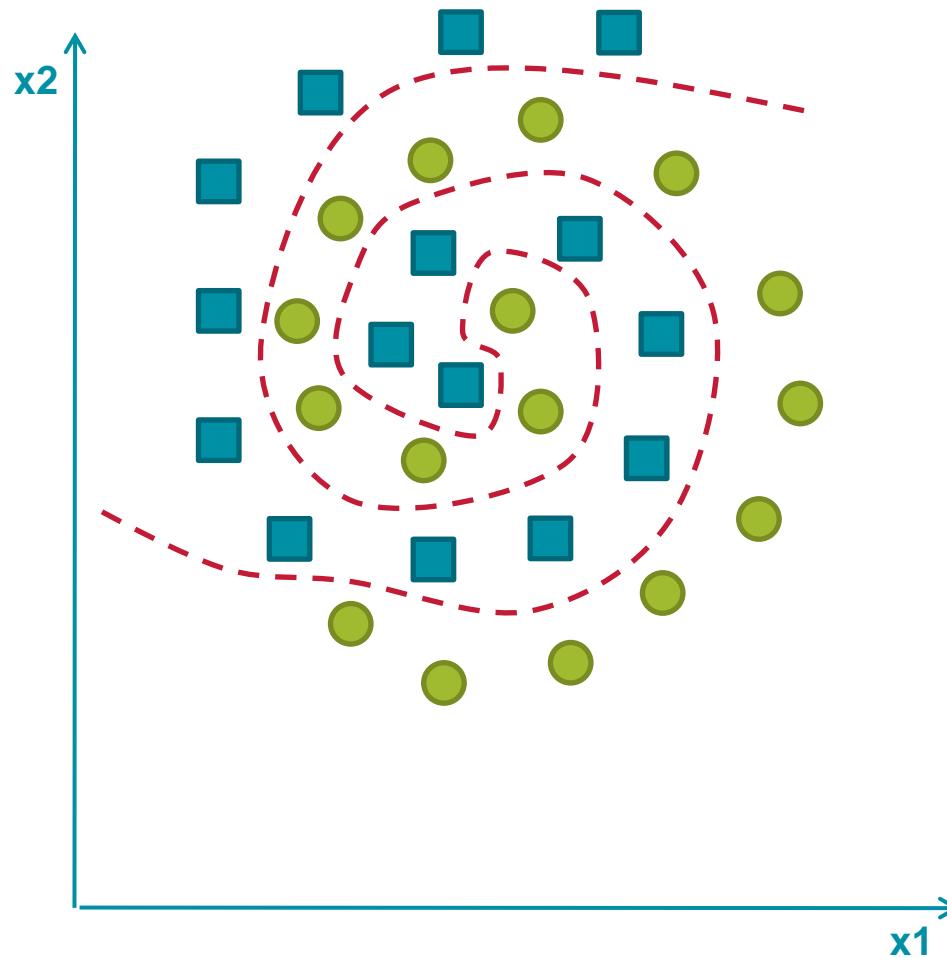
SVM: Nicht-lineare Klassifizierungsprobleme



SVM: Nicht-lineare Klassifizierungsprobleme



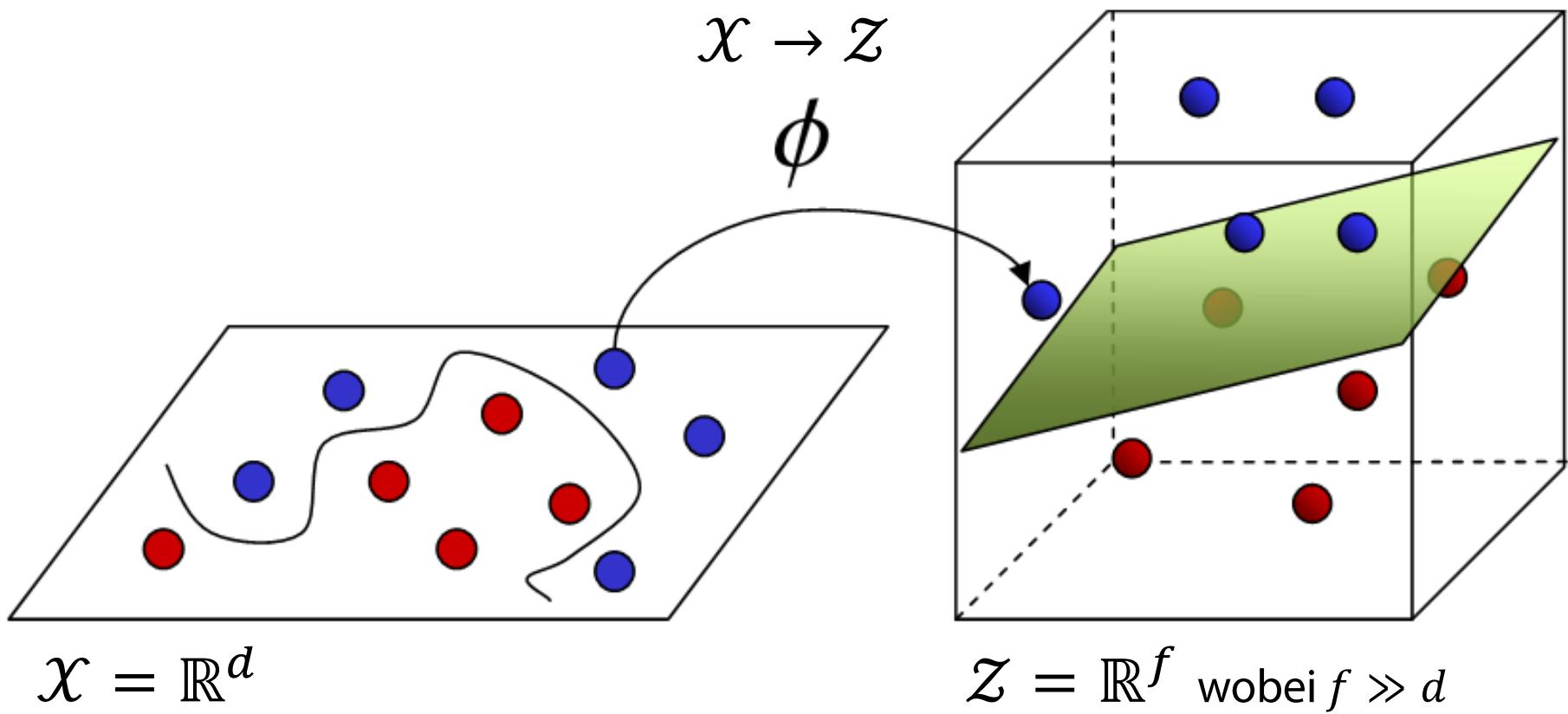
SVM: Nicht-lineare Klassifizierungsprobleme



- + ● N Trainings-observationen
- x_n
- - - Partitionierende Oberfläche

Aber:
SVM finden nur
lineare Lösung!

SVM: Transformation in den \mathcal{Z} -Raum



$$\mathcal{X} = \mathbb{R}^d$$

$$\mathcal{Z} = \mathbb{R}^f \text{ wobei } f \gg d$$

Wobei z.B. $\phi(x) = (x_1, x_2, x_1 + x_2)$

SVM: Das Optimierungsproblem

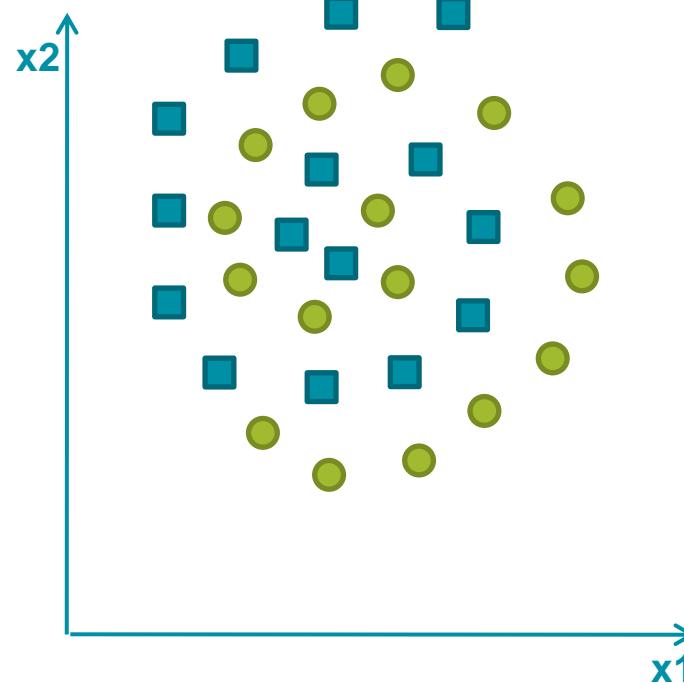
$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N y_n y_m \alpha_n \alpha_m \mathbf{x}_n^T \mathbf{x}_m \sum_{n=1}^N \alpha_n$$

Komplexität: $\mathcal{O}(N^2)$, bis auf $\mathbf{x}_n^T \mathbf{x}_m$ unabhängig von \mathcal{X}

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N y_n y_m \alpha_n \alpha_m \mathbf{z}_n^T \mathbf{z}_m \sum_{n=1}^N \alpha_n$$

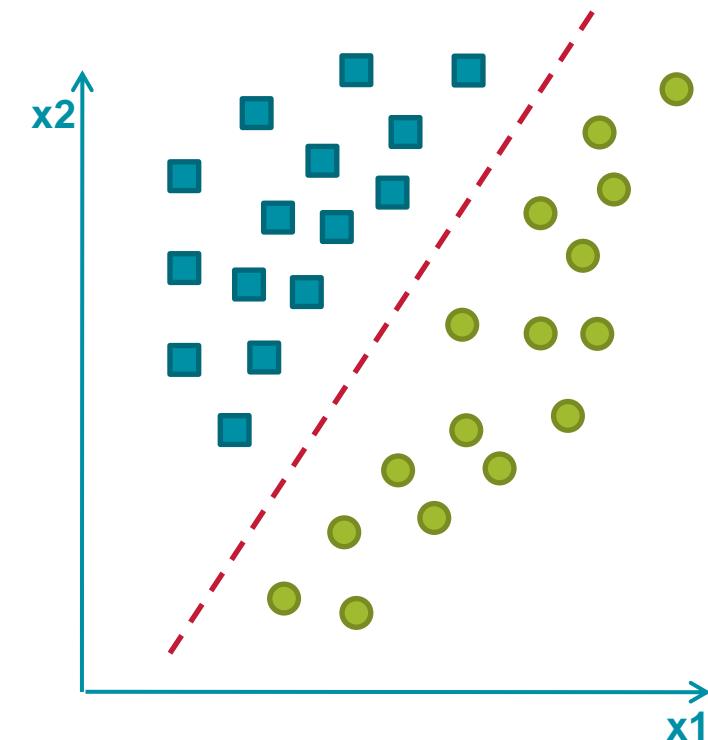
Komplexität: Dimensionalität von \mathcal{Z} beeinflusst nur $\mathbf{z}_n^T \mathbf{z}_m$

SVM: Transformation in den \mathcal{Z} -Raum



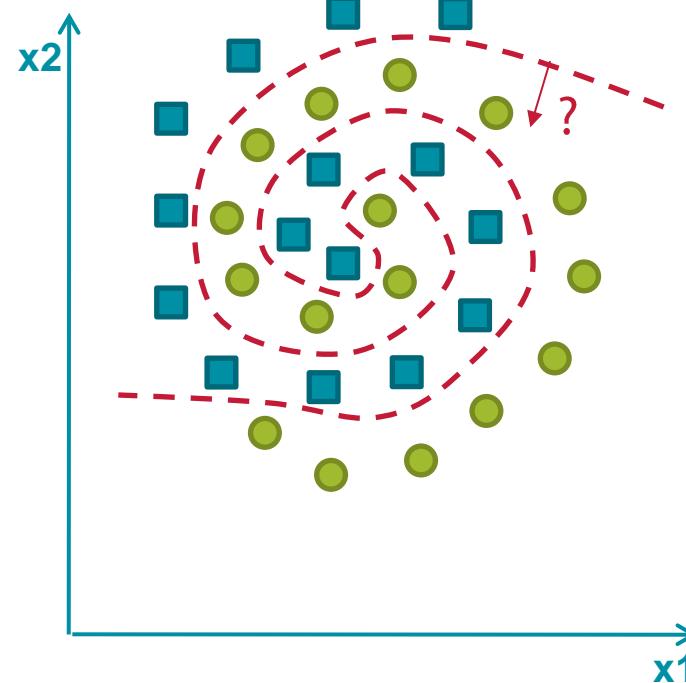
$$\mathcal{X} \rightarrow \mathcal{Z}$$

$$\mathcal{X} = \mathbb{R}^d$$

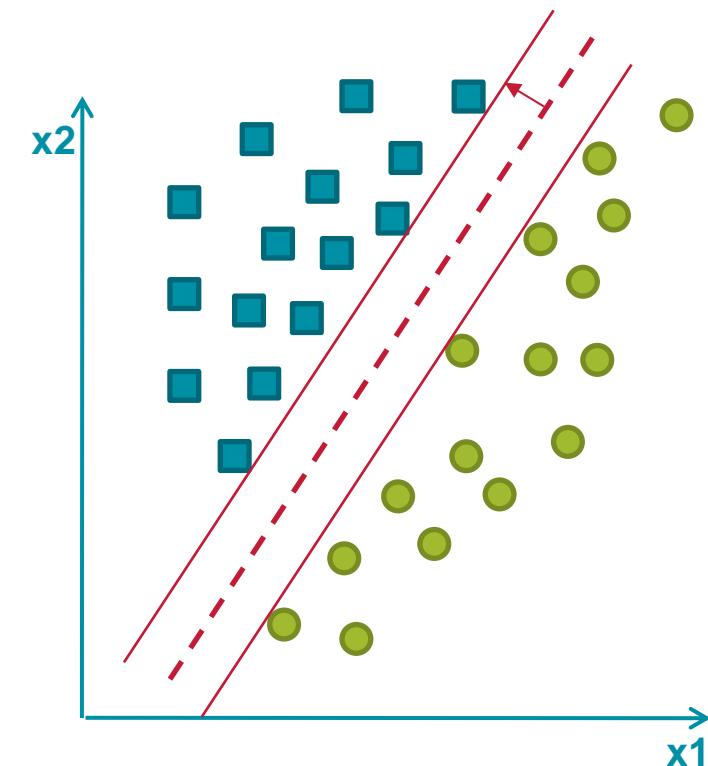


$$\mathcal{Z} = \mathbb{R}^f \text{ wobei } f \gg d$$

SVM: Transformation in den \mathcal{Z} -Raum

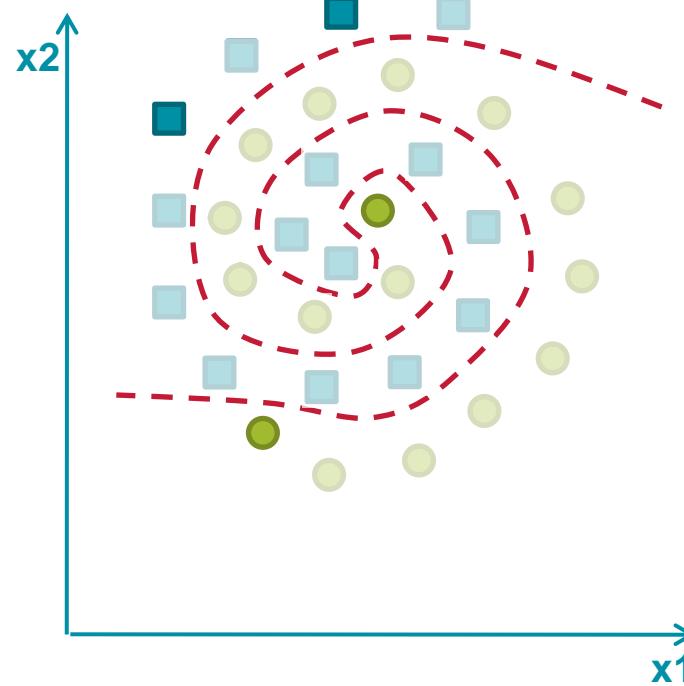

$$\mathcal{X} \rightarrow \mathcal{Z}$$

$$\mathcal{X} = \mathbb{R}^d$$

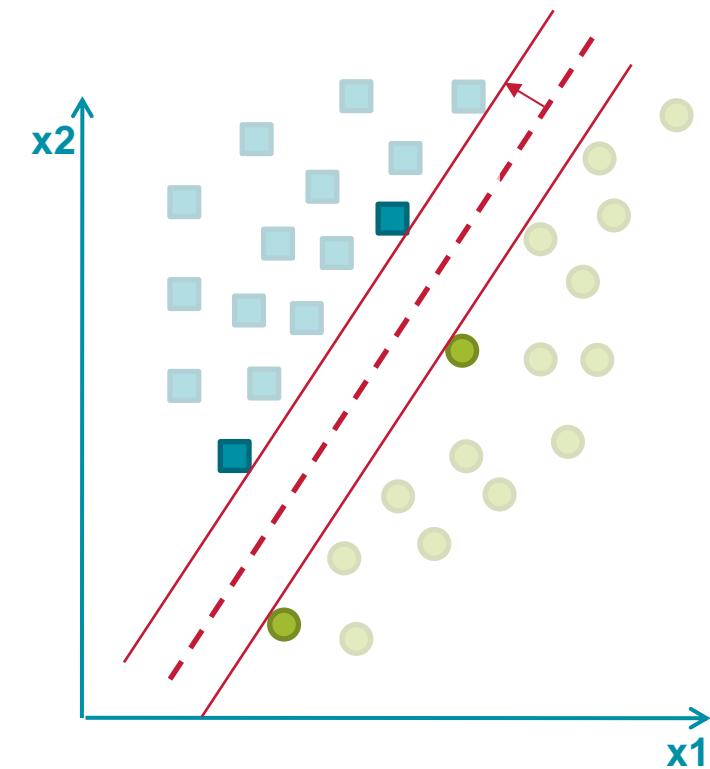


$$\mathcal{Z} = \mathbb{R}^f \text{ wobei } f \gg d$$

SVM: Transformation in den \mathcal{Z} -Raum

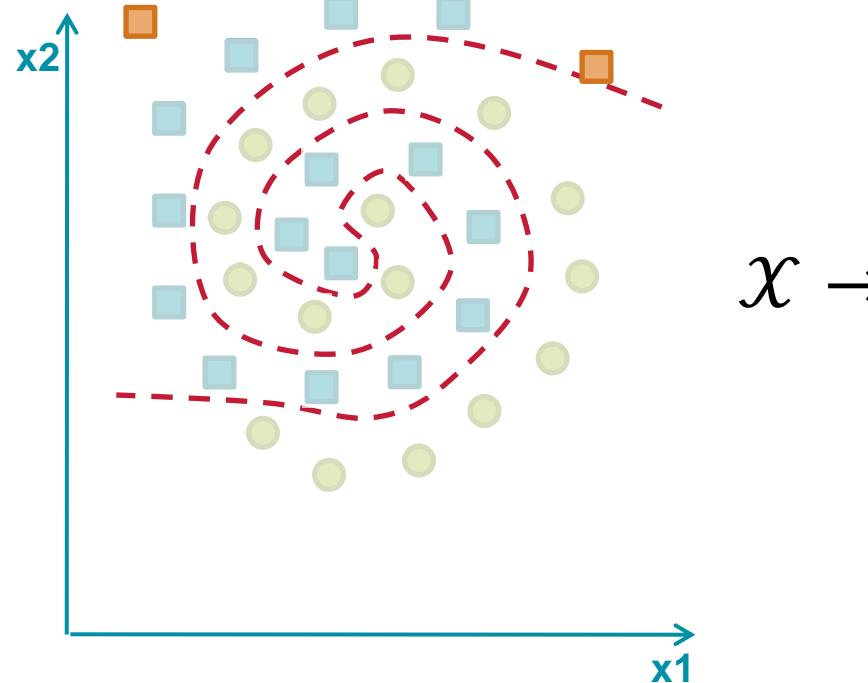

$$\mathcal{X} \rightarrow \mathcal{Z}$$

$$\mathcal{X} = \mathbb{R}^d$$

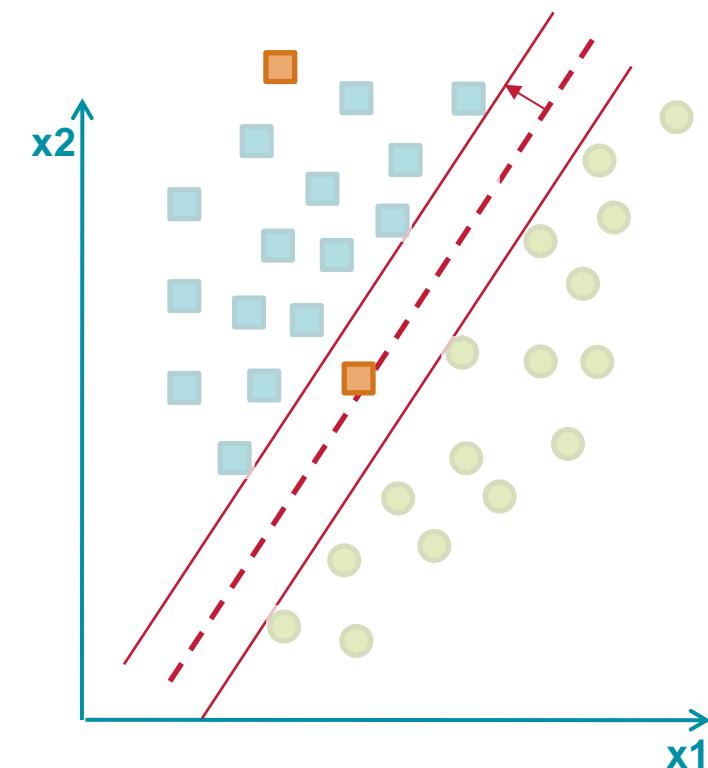


$$\mathcal{Z} = \mathbb{R}^f \text{ wobei } f \gg d$$

SVM: Klassifizierung im \mathcal{Z} -Raum



$$\mathcal{X} \rightarrow \mathcal{Z}$$



Skalarprodukt $w^T z$ resp. $z^T z$ muss berechnet werden

SVM: Vorläufige Zusammenfassung

- Hypothese: Maximaler Margin entspricht bester Generalisierung
- SVM findet partitionierende Hyperebene mit maximalem Margin
- Komplexität des SVM Trainings https. abhängig von N
- SVM kann nur lineare Separierung
- Transformation der Trainingsdaten in höherdimensionalen Raum
- Dort kann eine lineare Separierung gefunden werden
- Erfordert N^2 Skalarprodukte während des Trainings und jeweils eines pro Klassifizierung in \mathcal{Z} -Raum

SVM: Die Magie

- Die Berechnung des Skalarproduktes ist vergleichbar schnell
 - **Aber:** lineare Separierbarkeit nur bei sehr hoch-dimensionalem \mathcal{Z} -Raum garantiert
 - Komplexität des Skalarproduktes: $\mathcal{O}(\dim(\mathcal{Z}))$
-
- Training: $\mathbf{z}_n^T \mathbf{z}_m$
 - Anwendung: $\mathbf{z}^T \mathbf{z}$
- => Nur das Skalarprodukt, nicht \mathbf{z} wird benötigt
- Gibt es eine Möglichkeit das Skalarprodukt implizit zu berechnen ohne die eigentliche Transformation auszuführen?

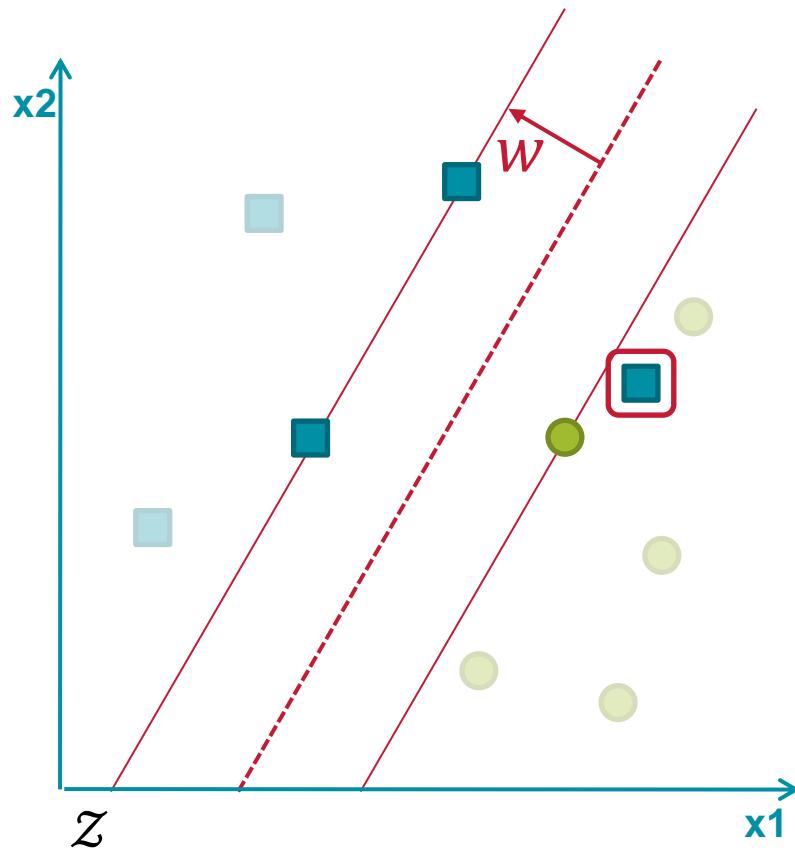
SVM: Der Kernel-Trick

- $\mathcal{X} = \mathbb{R}^2, \mathcal{Z} = \mathbb{R}^6, x = (x_1, x_2), x' = (x'_1, x'_2)$
- $z = \phi(x) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)$
- $z^T z' = \phi(x)\phi(x')$
- $z^T z' = \boxed{1 + x_1 x'_1 + x_2 x'_2 + x_1^2 x'^2_1 + x_2^2 x'^2_2 + x_1 x'_1 x_2 x'_2}$
- $K(x, x') = (1 + x^T x')^2$ 
- $K(x, x') = (1 + (x_1 x'_1 + x_2 x'_2))^2$
- $K(x, x') = \boxed{1 + \boxed{2}x_1 x'_1 + \boxed{2}x_2 x'_2 + x_1^2 x'^2_1 + x_2^2 x'^2_2 + \boxed{2}x_1 x'_1 x_2 x'_2}$

SVM: Kernels

- Müssen mathematische Eigenschaften erfüllen
- Einige bekannte Kernel Funktionen
 - Linear $K(x, x') = xx'$
 - Polynomial $K(x, x') = (\gamma * xx' + r)^d$
 - Radiale Basisfunktion $K(x, x') = \exp(-\gamma|x - x'|^2)$
 - Sigmoid $K(x, x') = \tanh(\gamma * xx' + r)$
- Berechnung auf x , d.h. völlig unabhängig von $\dim(\mathcal{Z})$
- Der \mathcal{Z} -Raum ist unbekannt
- Der \mathcal{Z} -Raum der RBF ist unendlich
- $\exp\left(-\frac{1}{2}|x - x'|^2\right) = \sum_{j=0}^{\infty} \frac{(x^T x')^j}{j!} \exp(-\frac{1}{2}|x|^2) \exp(-\frac{1}{2}|x'|^2)$

SVM: Soft-Margin



- Anzahl der Support-Vektoren bestimmt Fehler
- Ausreißer / Rauschen erhöht benötigte Support-Vektoren
- **Lösung:** Soft-Margin
- Neuer Parameter C
 - Je kleiner C, desto „weicher“ der Margin
- Non-Margin Support-Vektoren

SVM: Zusammenfassung

- Soft-Margin Klassifikator berücksichtigt Rauschen
- Observationen werden in ∞ -dimensionalen Raum abgebildet
- Nutzt linear Methode um nicht-lineare Klassifizierungsprobleme in einem höherdimensionalen Raum mit dem Kernel-Trick zu lösen
- Komplexität des Trainings abhängig von der Anzahl Trainingsobservationen
- Komplexität der Klassifizierung abhängig von der Anzahl der Support-Vektoren
- Generalisierbarkeit / Zu erwartender Fehler abhängig von der Anzahl der Support-Vektoren
- Black-box Klassifizierer

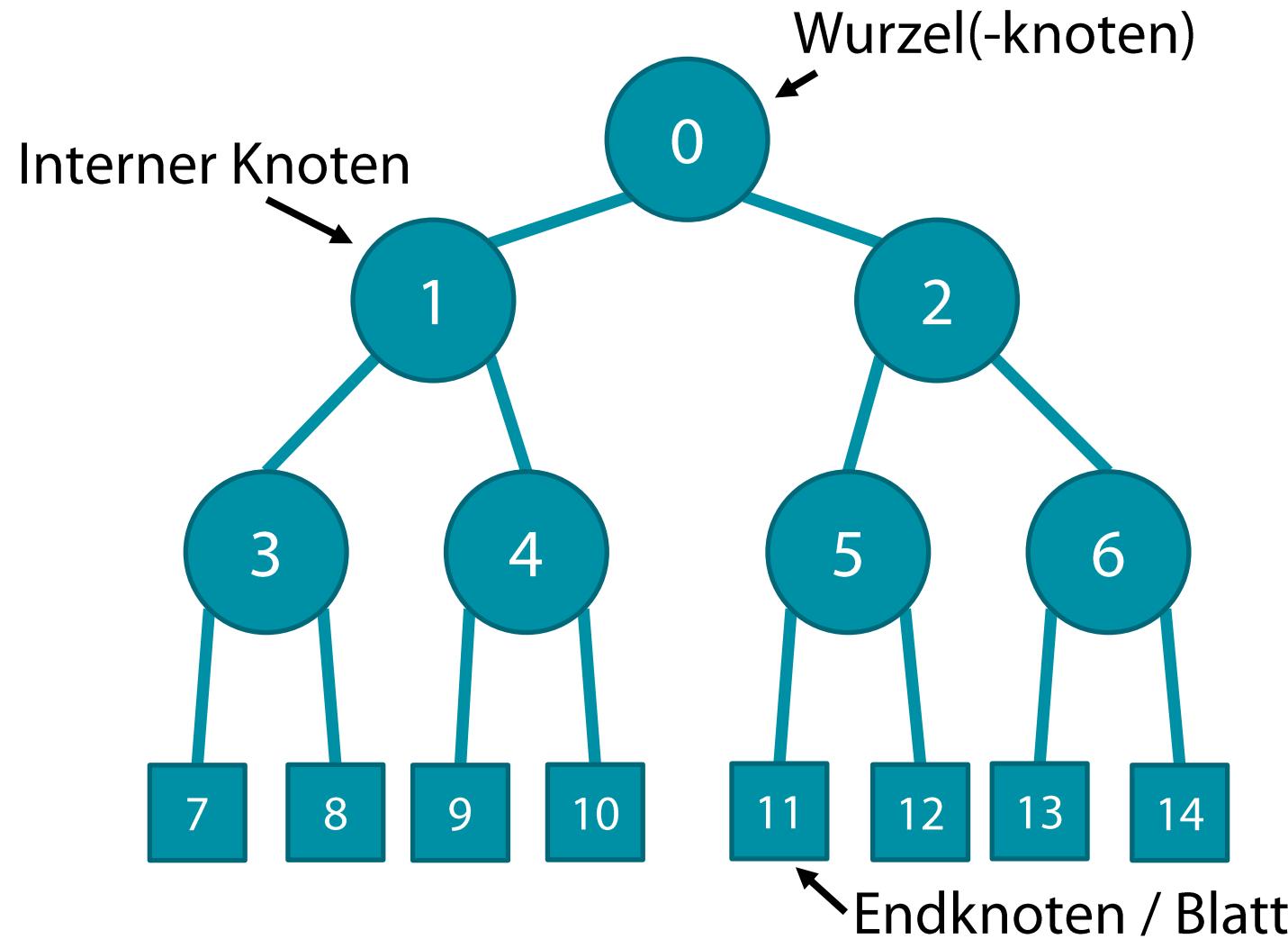


SVM: Q&A

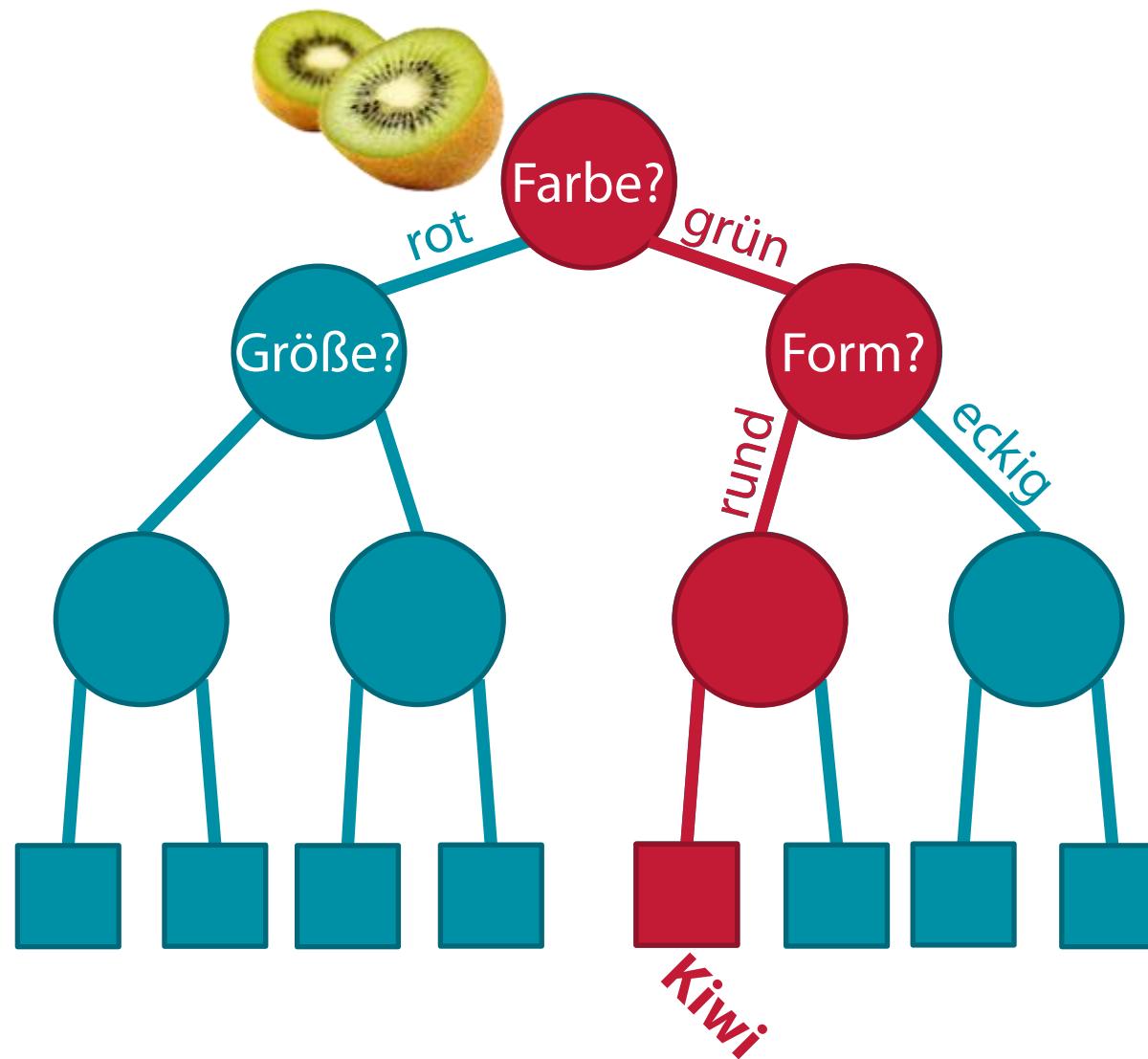


Entscheidungs- bäume

Entscheidungsbaum: Struktur

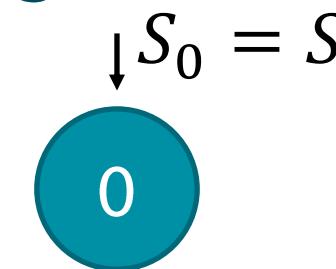
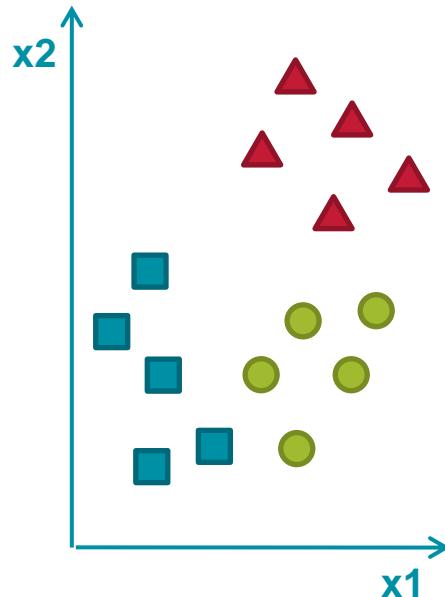


Entscheidungsbaum: Einfaches Beispiel



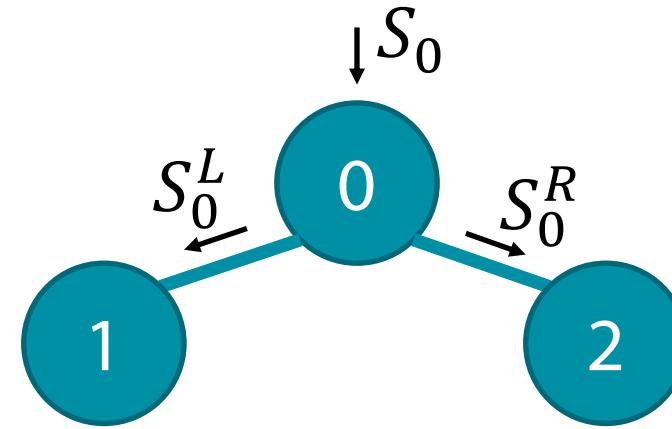
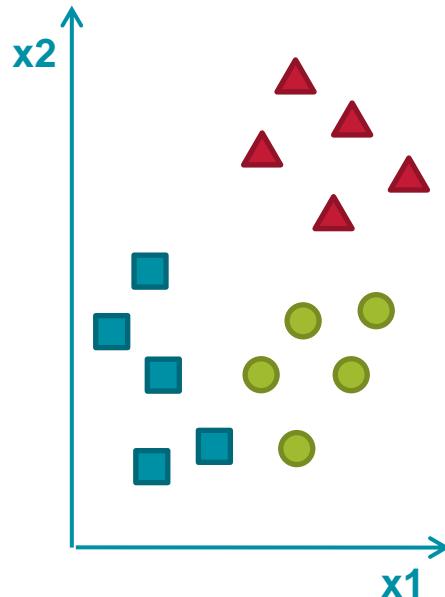
Entscheidungsbaum: Training

- Trainingsobservationen
- $S = \{x_0, \dots, x_N\}$



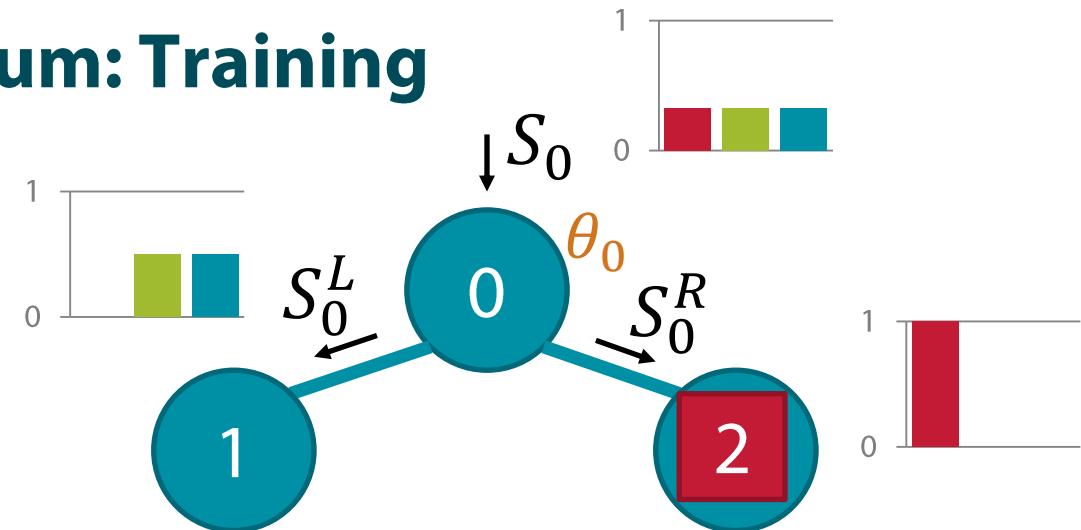
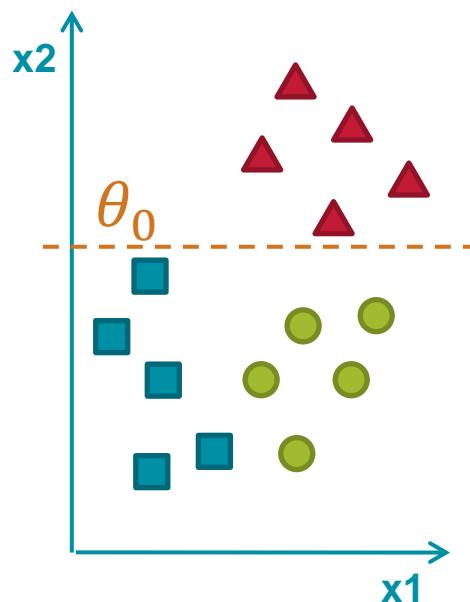
Entscheidungsbaum: Training

- Trainingsobservationen
- $S = \{x_0, \dots, x_N\}$



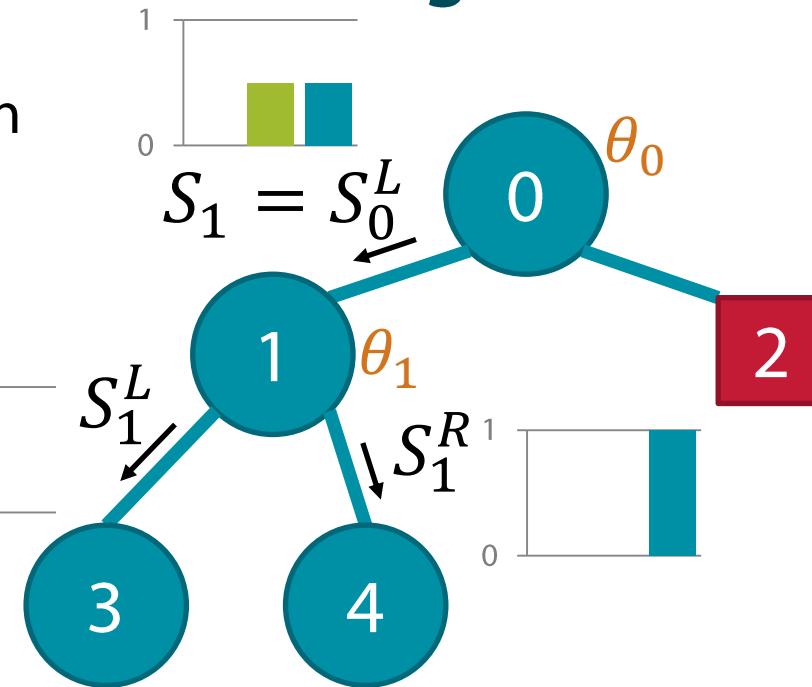
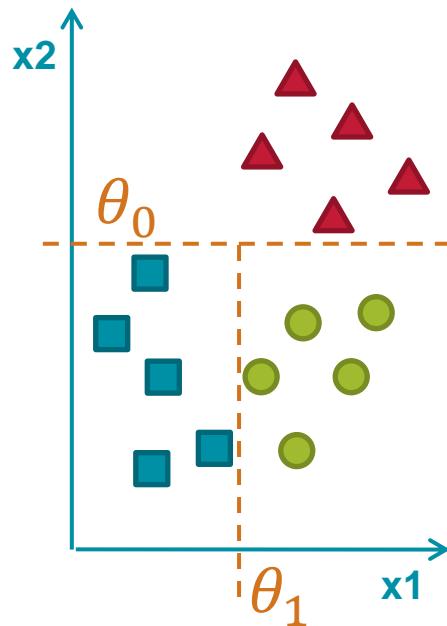
Entscheidungsbaum: Training

- Trainingsobservationen
- ▲ $S = \{x_0, \dots, x_N\}$



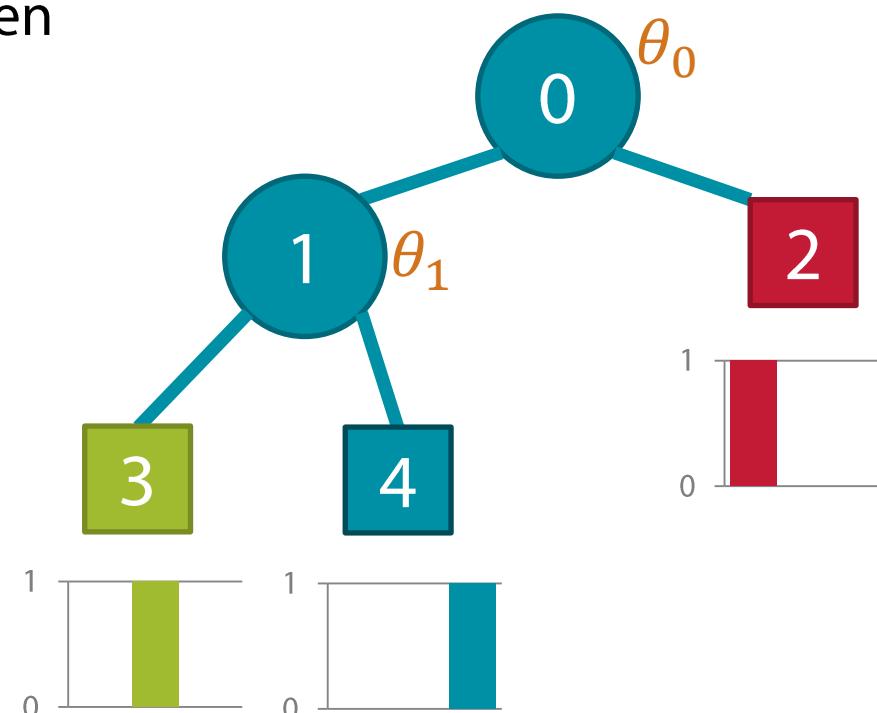
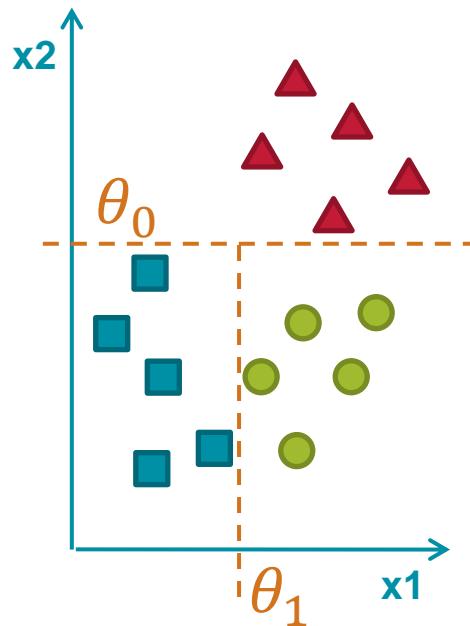
Entscheidungsbaum: Training

- Trainingsobservationen
- ▲ $S = \{x_0, \dots, x_N\}$



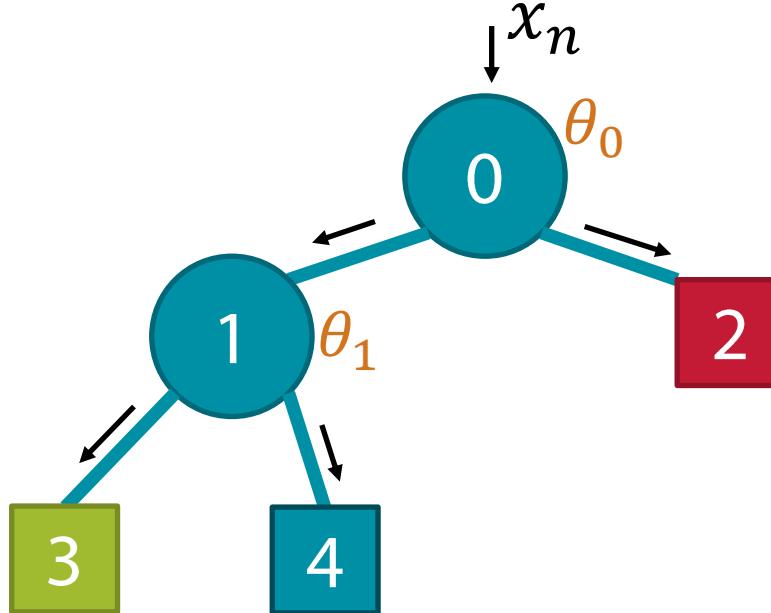
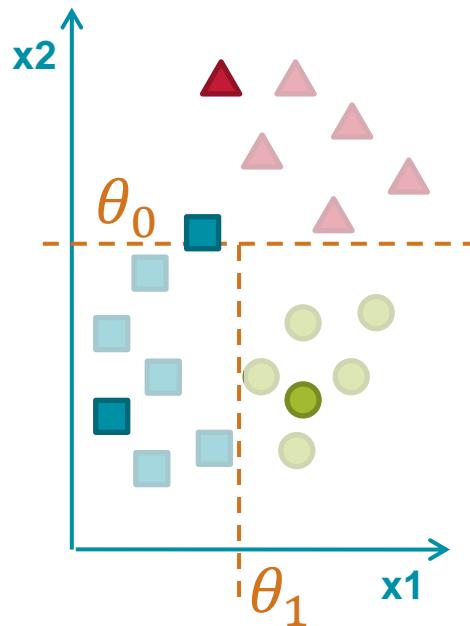
Entscheidungsbaum: Training

- Trainingsobservationen
- $S = \{x_0, \dots, x_N\}$

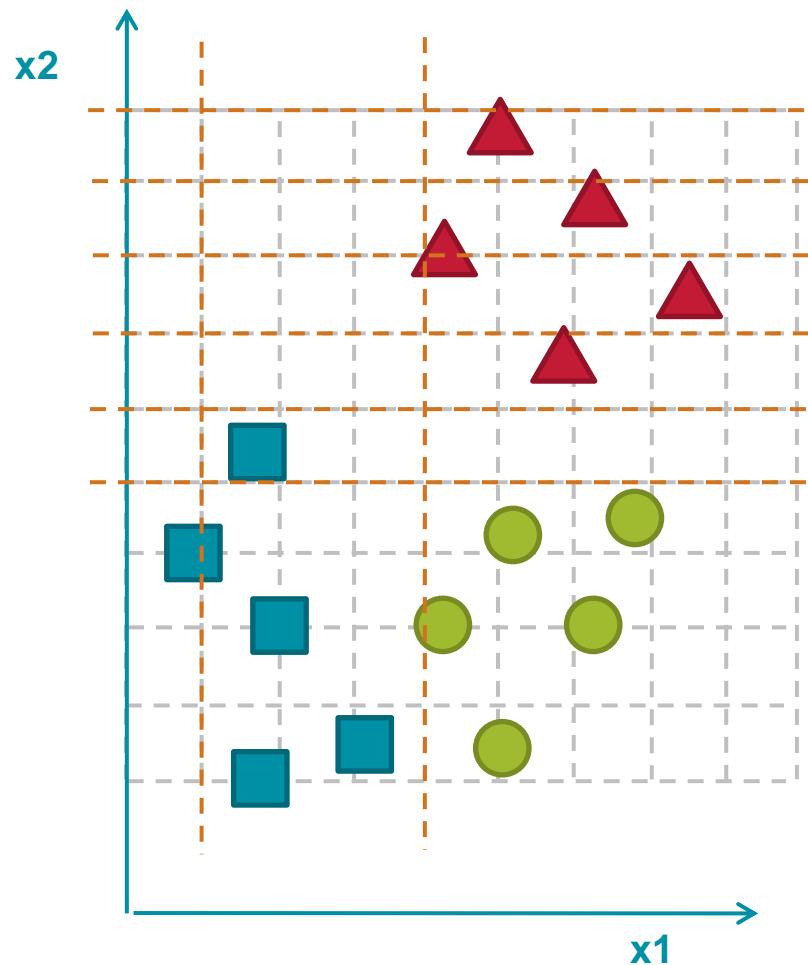


Entscheidungsbaum: Anwendung

- Trainingsobservationen
- $S = \{x_0, \dots, x_N\}$

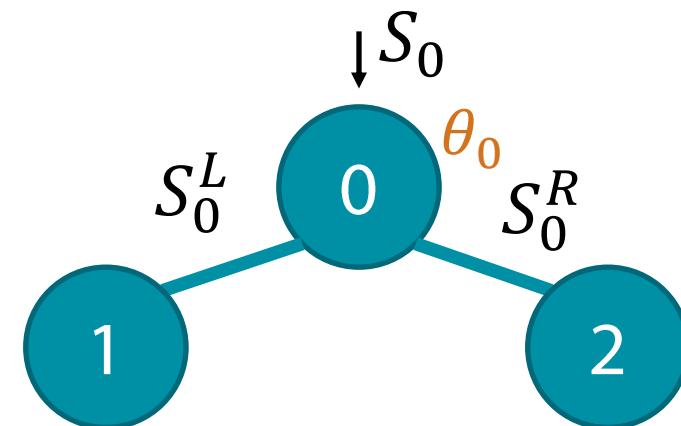


Entropy / Informationsgewinn

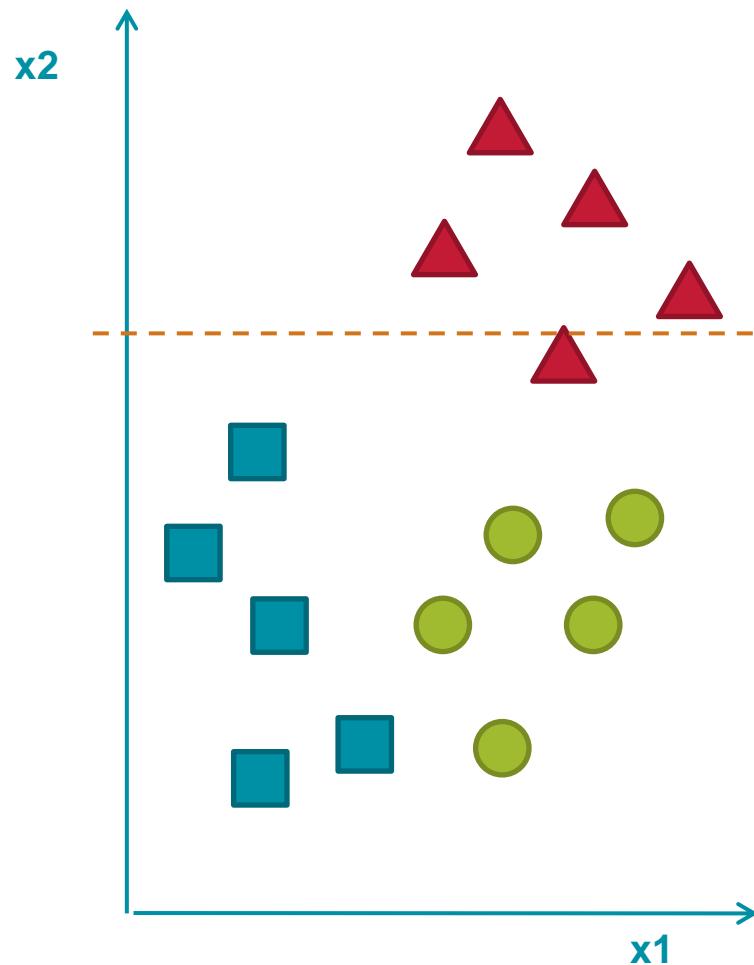


Wie finde ich die beste Partition über ein Merkmal?

1. Diskretisierung des Merkmalraumes
2. Abtasten

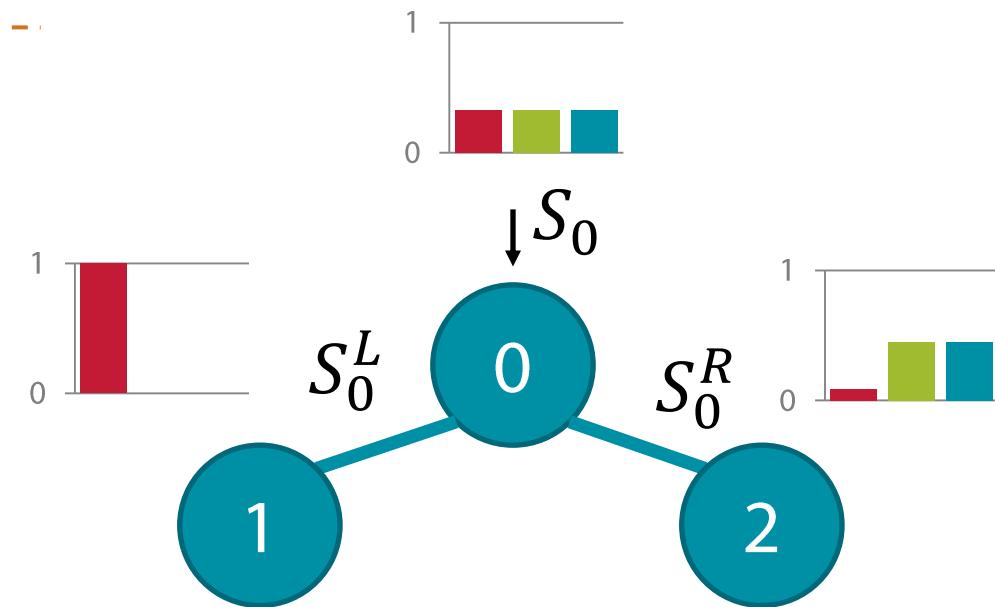


Entropy / Informationsgewinn

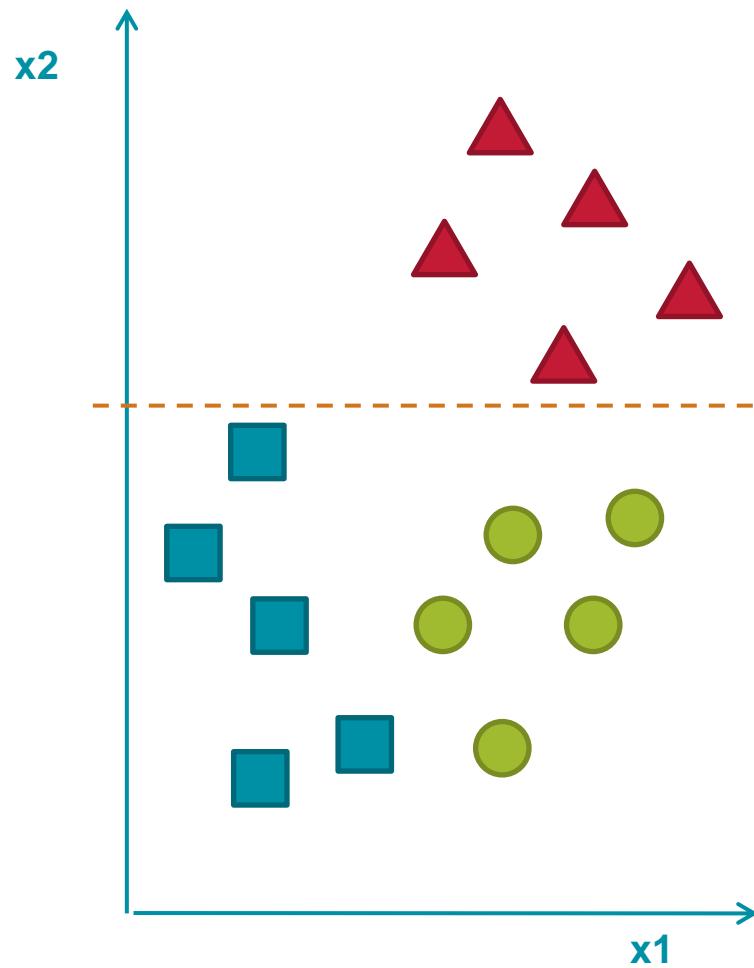


Wie messe ich die Güte einer Partitionierung?

1. Informationsgewinn

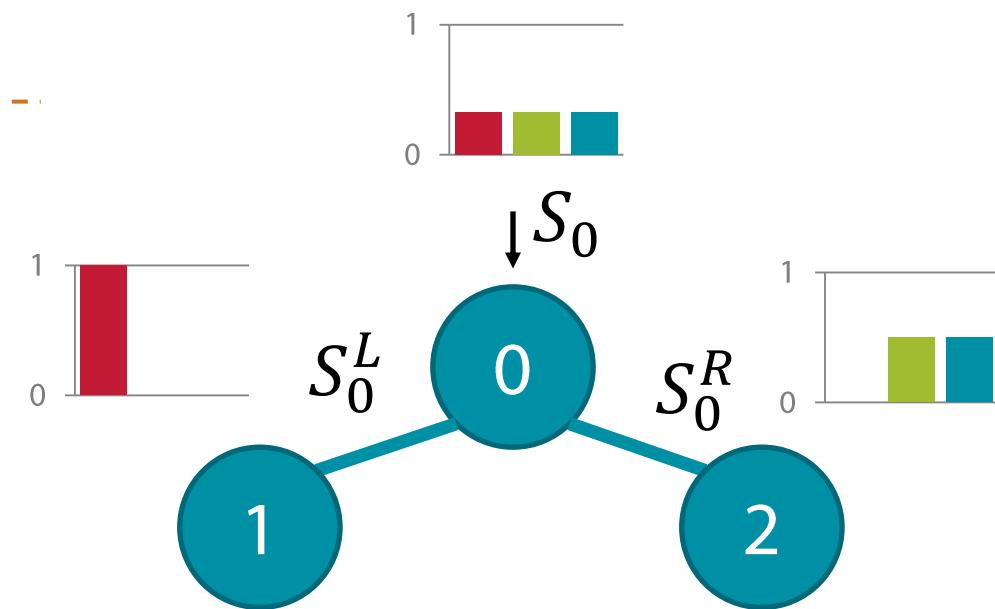


Entropy / Informationsgewinn



Wie messe ich die Güte einer Partitionierung?

1. Informationsgewinn



Entropy / Informationsgewinn

Informationsgewinn I

$$I = H(S) - \frac{|S^L|}{|S|} H(S^L) - \frac{|S^R|}{|S|} H(S^R)$$

Shannon-Entropie $H(S)$

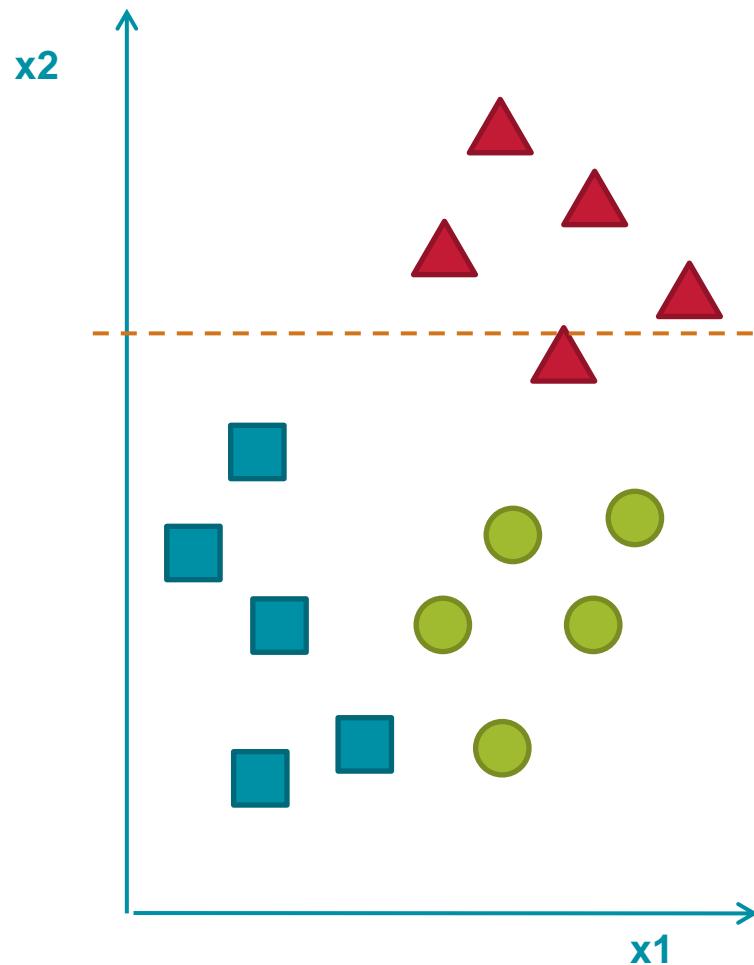
$$H(S) = - \sum_{c \in C} p(c) \log_2 p(c)$$

Wobei $C = \{\blacksquare, \bullet, \blacktriangle\}$ und $p(c)$ die Wahrscheinlichkeit c in der Menge S anzutreffen.

Entropie:

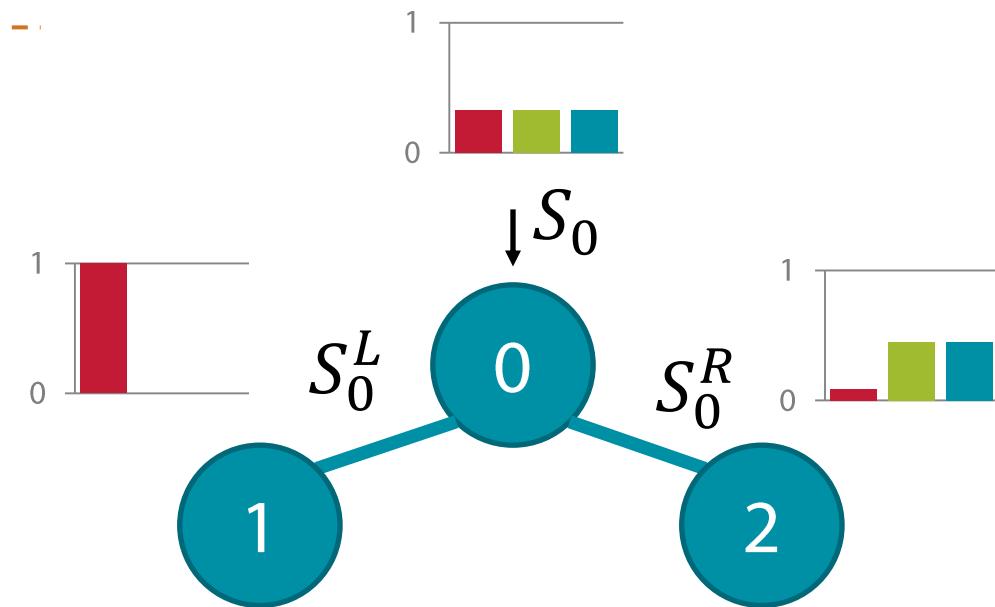
- Maß für den mittleren Informationsgehalt.
- Je geringer die Entropie, desto höher der Informationsgehalt.
- Entropie (~ Unordnung) nimmt monoton zu.

Entropy / Informationsgewinn

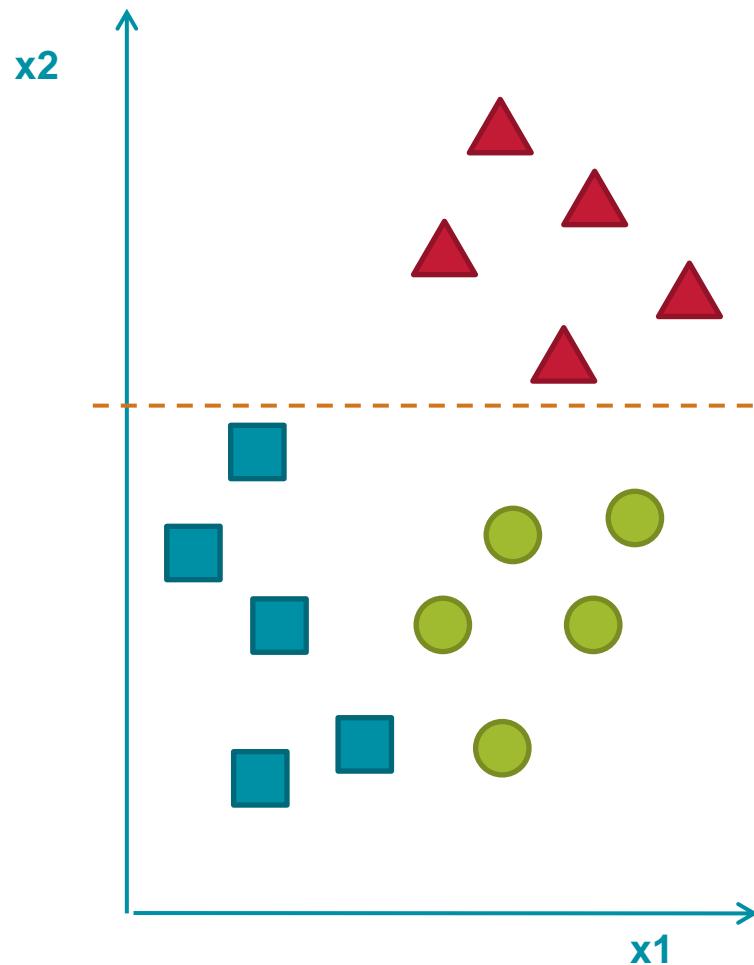


Wie messe ich die Güte einer Partitionierung?

1. Informationsgewinn $I = 0.18$

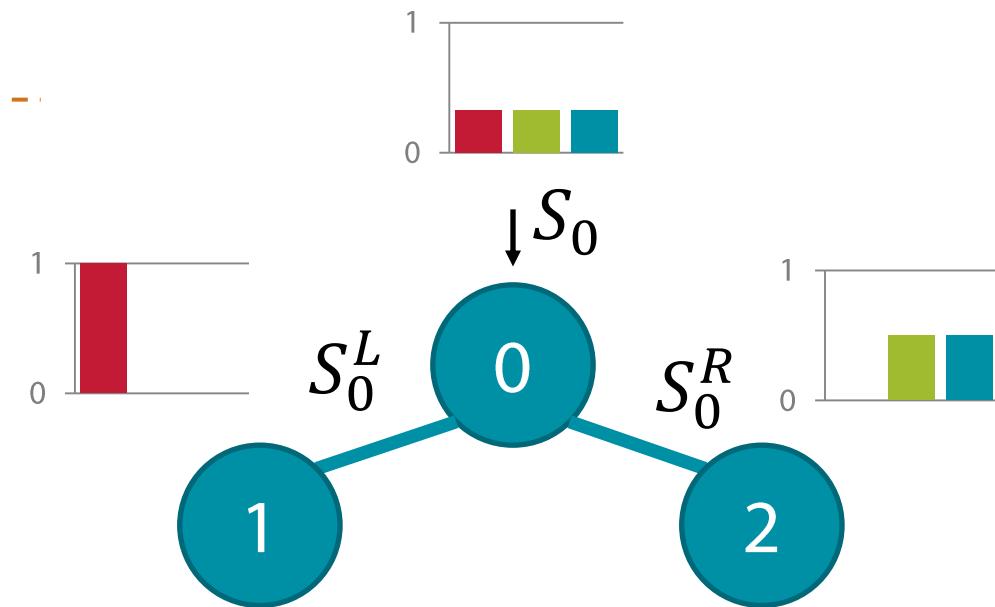


Entropy / Informationsgewinn



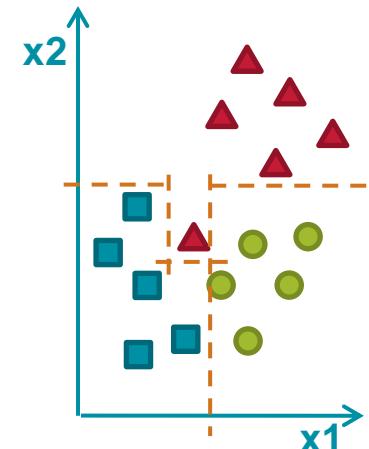
Wie messe ich die Güte einer Partitionierung?

1. Informationsgewinn $I = 0.28$



Entscheidungsbaum: Zusammenfassung

- Automatische, deterministische Klassifizierer
- Optimale Partitionierung nach Trainingsdaten
- „Wichtige“ Merkmale befinden sich höher im Baum
- Sehr abhängig von der Güte der Trainingsdaten
- Hang zur Überanpassung





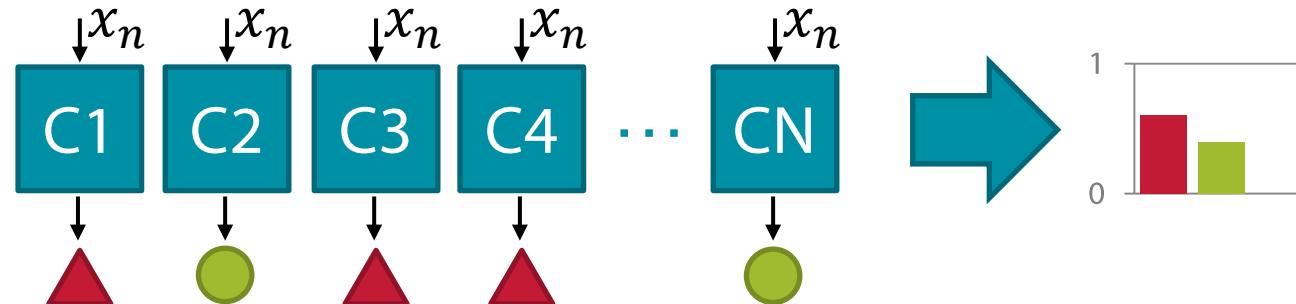
Entscheidungs-Bäume: Q&A



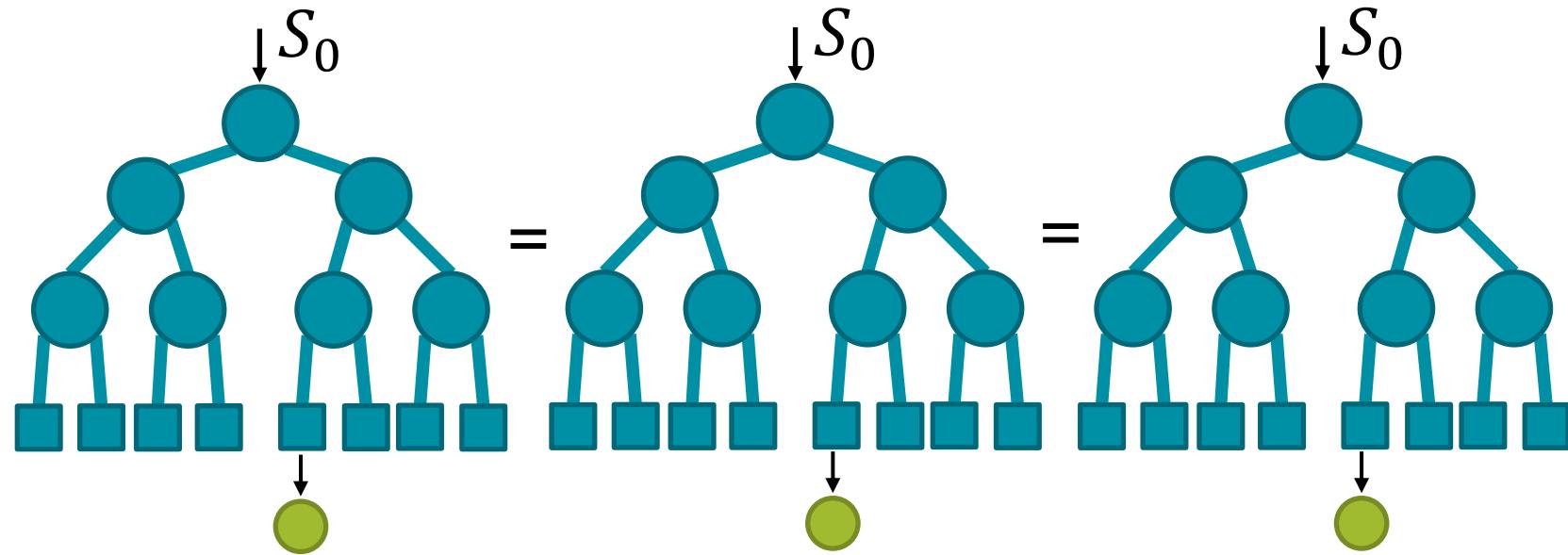
Random Decision Forests (RDF)

Mehr ist besser: Ensemble Methoden

- Bekannte Probleme vieler Klassifizierer
 - Große Abhängigkeit von den Trainingsdaten
 - Überanpassung / Overfitting
 - Empfindlich gegenüber Rauschen und Ausreißern
 - Fluch der Dimensionalität
- Idee: **Ensemble Methoden**
 - Ein ganzes Ensemble sog. schwacher Klassifizierer wird trainiert
 - Abstimmung über die Klassenzugehörigkeit bzw. Wahrscheinlichkeit



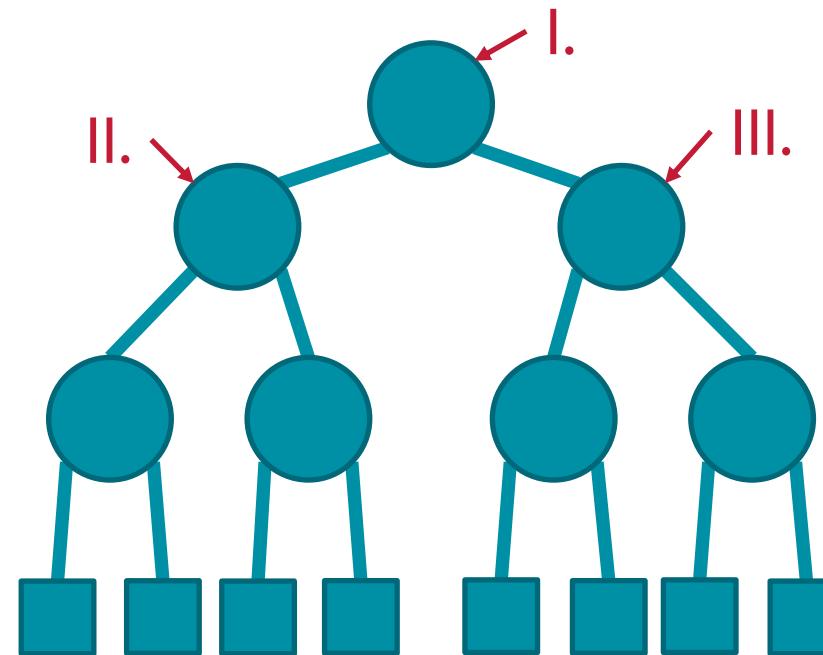
Random Decision Forests



- **Problem:** Entscheidungsbäume trainieren deterministisch
- **Deshalb:** **Random** Decision Forests

Das Random in Random Decision Forests

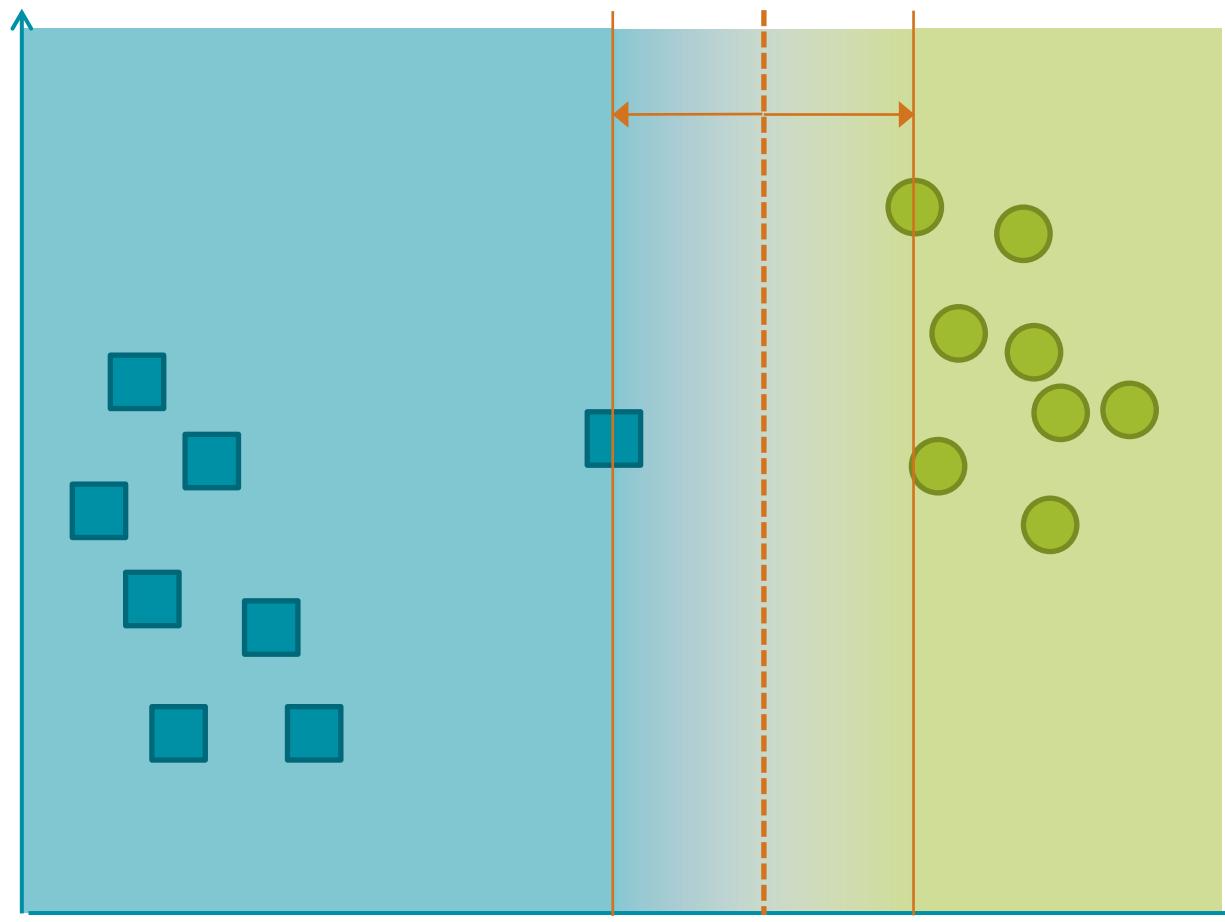
- I. Zufällige Auswahl der Trainingsmenge
- II. Zufällige Auswahl einer Untermenge der Merkmale
- III. Zufällige Auswahl des Schwellwertes



Randomness I: Bagging

- Jeder Baum t wird mit einer zufällig gesampelten Untermenge S_t der Gesamttrainingsmenge S trainiert
 - Häufig gleichmäßiges Sampeln mit Zurücklegen
-
- Jeder Baum wird ein wenig anders
 - Große Cluster im Merkmalsraum wirken sich weiterhin stark aus
 - Einzeln stehende Observationen werden runtergewichtet

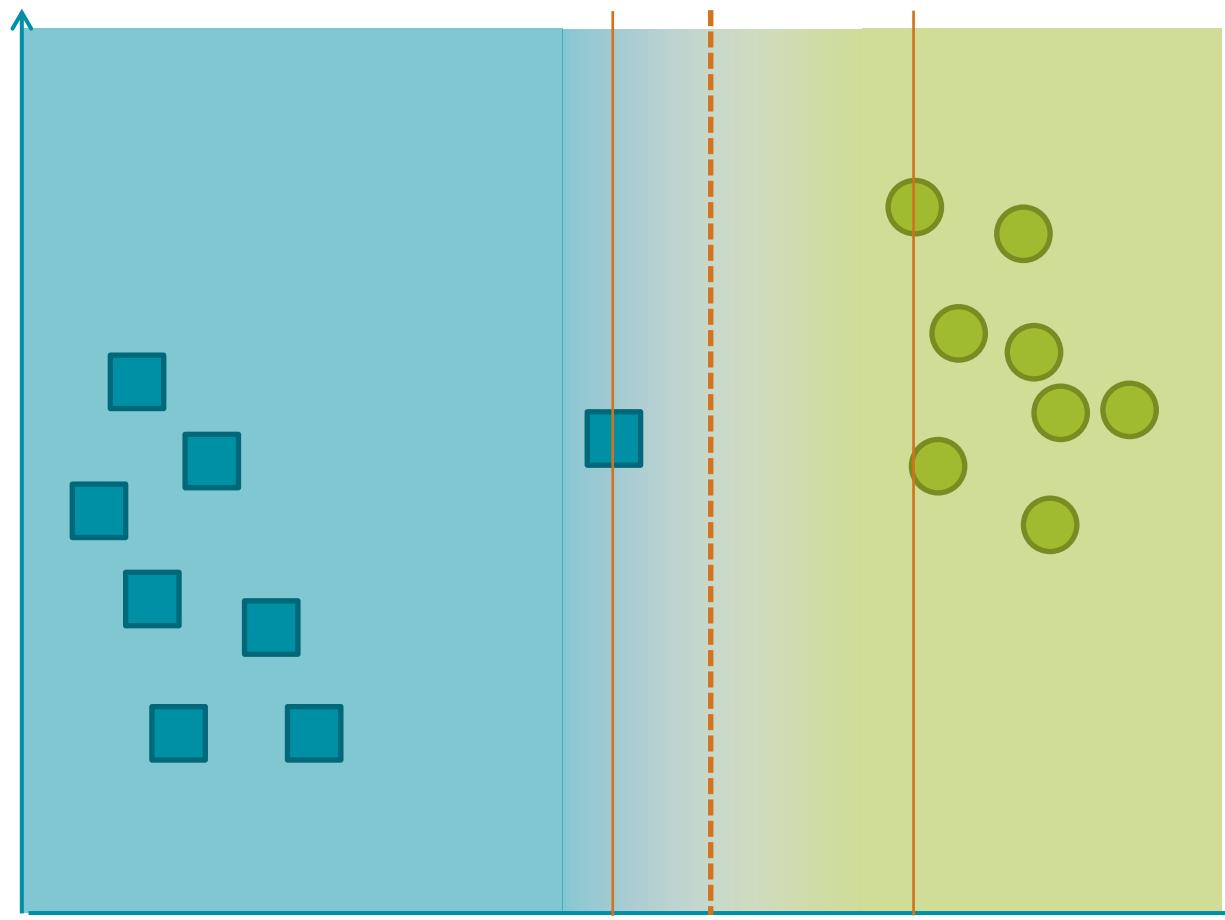
Randomness I: Bagging



Beispiel ohne
Bagging:

- Wie Maximaler Margin
- Blauer Ausreißer hat großen Einfluss

Randomness I: Bagging



Beispiel mit
Bagging:

- Wie Soft-Margin SVM
- Manche Bäume trainieren ohne den Ausreißer
- Ausreißer hat weniger Einfluss

Einschub: Formalisierung des Trainings

Binäre Partitionierung

$$h(x, \theta_j) = [\tau > \phi(x)]$$

, wobei ϕ ein einzelnes Merkmal aus einer Observation x extrahiert, τ einen Schwellwert darstellt und $\theta_j = \{\tau, \phi\}$ ihre Menge darstellt.

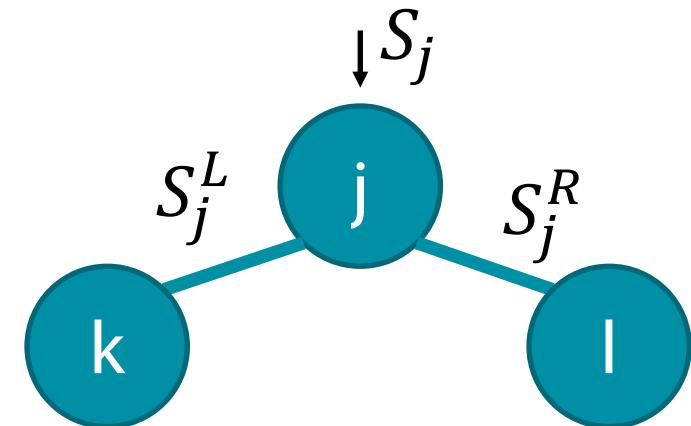
Informationsgewinn

$$I_j = I(S_j, S_j^L, S_j^R, \theta_j)$$

Optimierungsproblem

$$\theta_j^* = \arg \max_{\theta_j \in \Upsilon} I_j$$

, wobei Υ die Menge aller möglichen Parameterkombinationen darstellt.



Randomness II: Zufällige Merkmale

Optimierungsproblem

$$\theta_j^* = \arg \max_{\theta_j \in \Upsilon_j} I_j$$

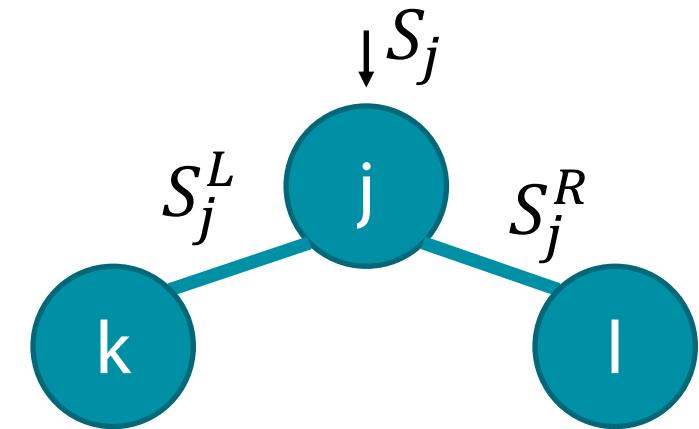
, mit Menge der unbekannten Parameter $\theta_j = \{\tau, \phi\}$.

Zufällige Merkmale

$M = |x|$ ist die Menge aller Merkmale (d.h. Dimensionalität des Merkmalsraumes \mathcal{X}).

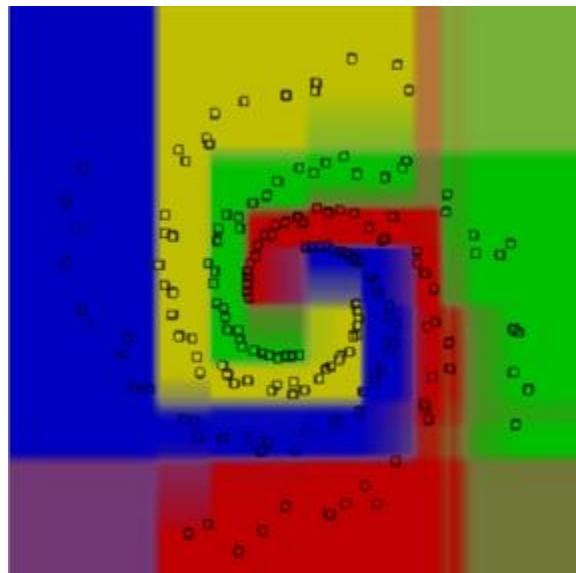
$M_j \subset M$ ist eine Untermenge von p zufällig gesampelten Merkmalen für Knoten j .

$\Upsilon_j \subset \Upsilon$ ist dann die Menge aller möglichen Parameterkombinationen θ_j deren Merkmalselektor ϕ ein $\mu \in M_j$ selektiert.

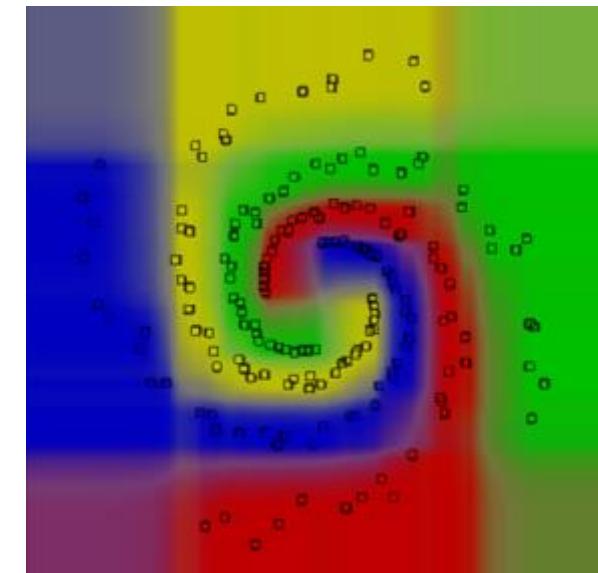


Randomness II: Zufällige Merkmale

- Schnelleres Training, da der Suchraum kleiner ist
- Reflektiert verschiedene Herangehensweisen an das Problem
- Robuster gegen schlecht gewählte Merkmale
- Klassengrenzen werden unschärfer => geringere Überanpassung



Large ρ



Small ρ

Randomness III: Zufälliger Schwellwert

Optimierungsproblem

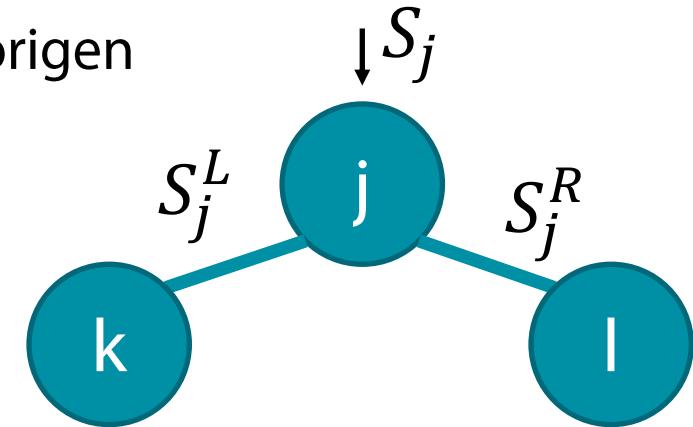
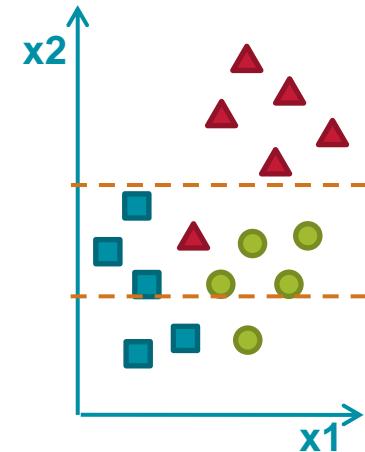
$$\theta_j^* = \arg \max_{\theta_j \in \Gamma} I_j$$

, mit Menge der unbekannten Parameter $\theta_j = \{\tau_\phi, \phi\}$.

Zufälliger Schwellwert

Γ ist die diskrete Menge aller Schwellwerte τ .

Aus den zu jedem Merkmalsselektor ϕ gehörigen Schwellwerten $\Gamma_\phi \in \Gamma$ wird ein zufälliger Schwellwert τ_ϕ gewählt.

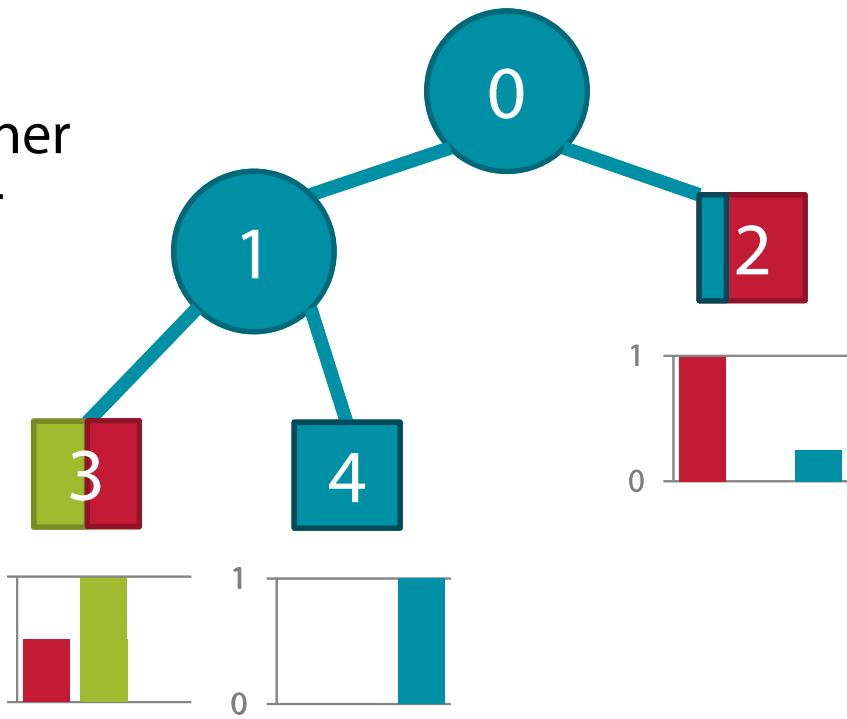


Randomness III: Zufälliger Schwellwert

- Schnelleres Training, da sehr viel kleinerer Suchraum
- Sub-optimale und schlechte Partitionen häufig
- Ermöglicht nicht-intuitive Herangehensweisen an das Problem
- Größere / tiefere Bäume
- Robuster gegen Rauschen und Ausreißer

Wahrscheinlichkeitsblätter

- Mehrere Bäume stimmen ab => Ausgabe muss nicht mehr „crisp“ sein
- Observationen verschiedener Klassen in einem Blatt erlauben
- Baum gibt Wahrscheinlichkeit einer Klassenzugehörigkeit anstatt der Klassenzugehörigkeit zurück



Forest-Parameter

size (Größe)

Bestimmt die Anzahl der Bäume im Ensemble (z.B. $T = 200$)

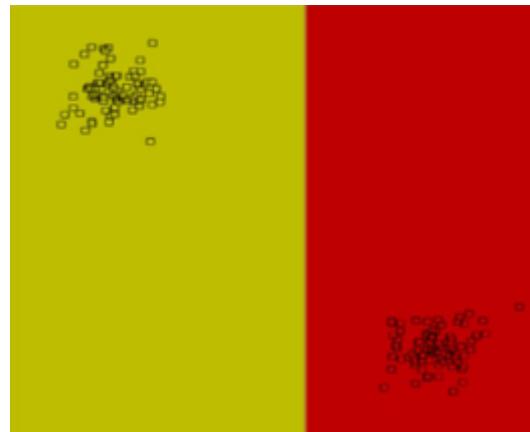
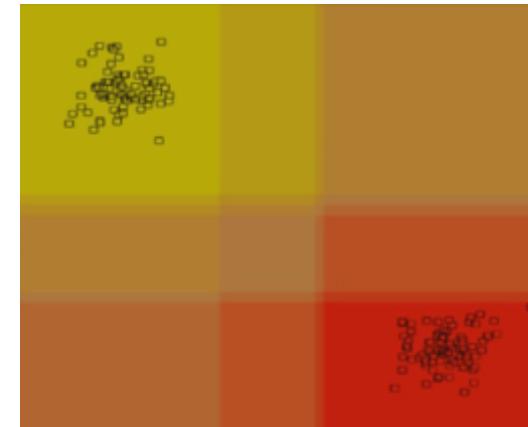
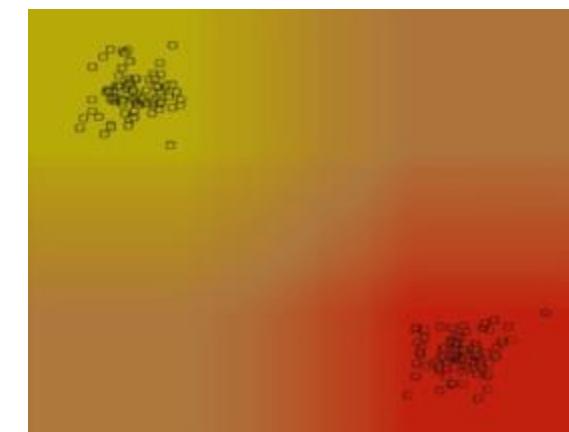
depth (Tiefe)

Bestimmt die maximale Tiefe der Bäume. (z.B. $D = 25$)

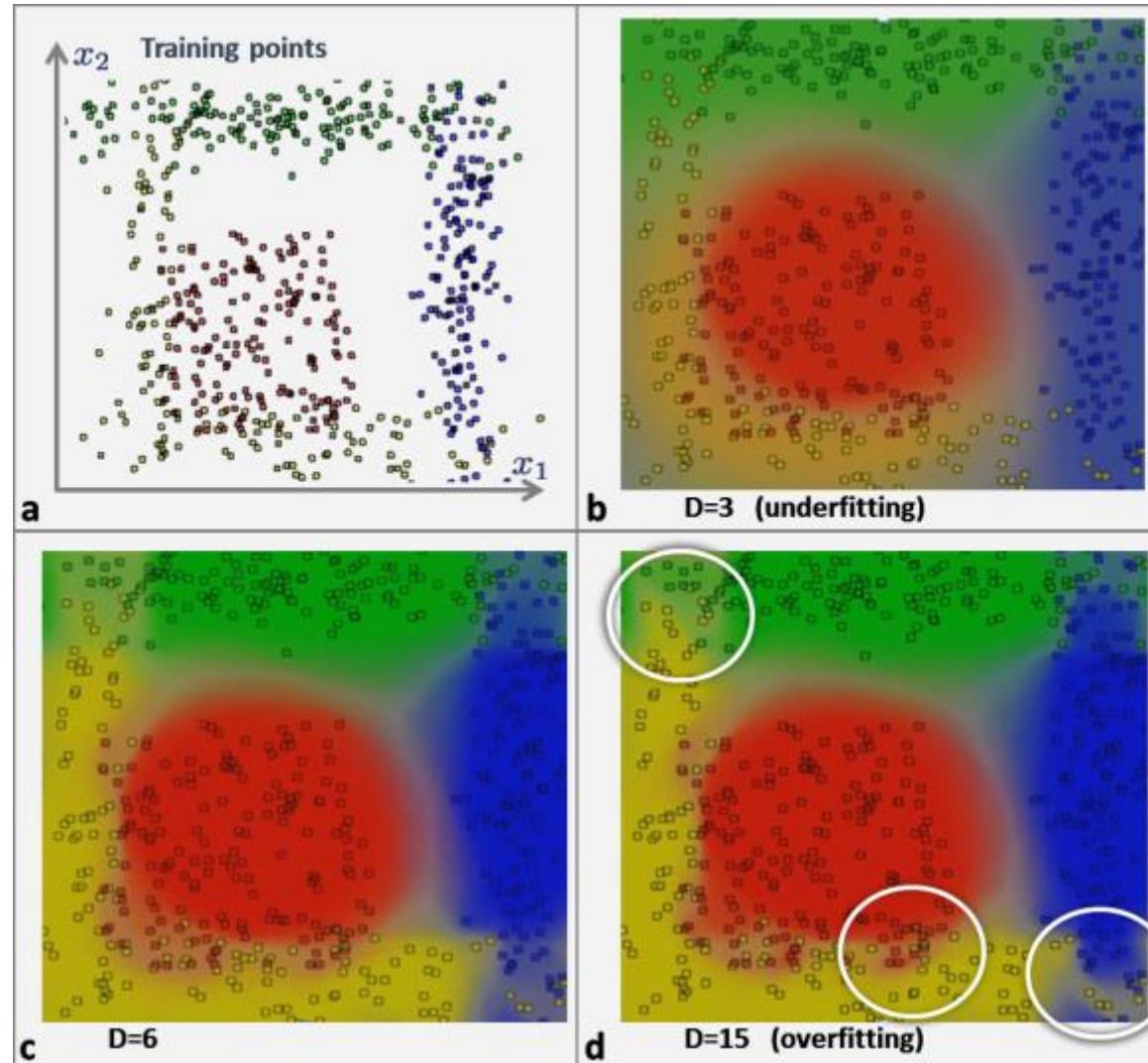
randomness (Zufälligkeit)

- **bagging-size** (z.B. $b = 50\%$)
- **feature-per-node** (z.B. $\rho = \sqrt{|M|}$)

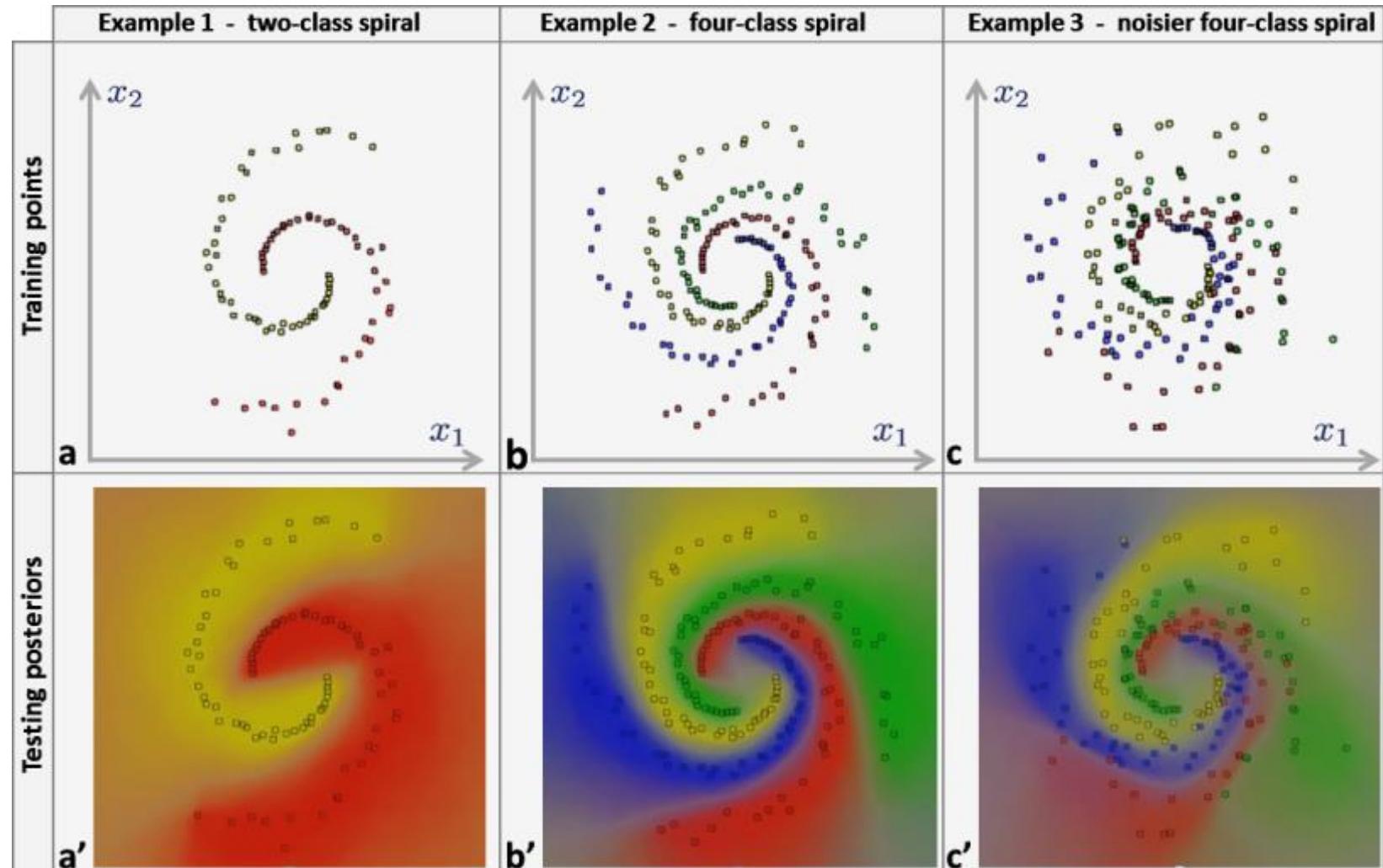
Forest-Parameter: size T

 $T = 1$  $T = 8$  $T = 200$

Forest-Parameter: depth D



RDFs und Rauschen



Random Decision Forests: Zusammenfassung

- Ensemble von schwachen Klassifizierern
- Implizit Mehrklassen-Klassifizierer
- Durch Zufälligkeit robust gegen schlechte Merkmale
- Durch Zufälligkeit robust gegen Ausreißer und Rauschen
- Durch Zufälligkeit gute Generalisierung
- Nicht-lineare Klassifizierung
- In Praxis robust gegen Parameterwahl
- Komplexität des Trainings abhängig von der Anzahl der Bäume und der Merkmale
- Komplexität der Klassifizierung abhängig von der Anzahl der Bäume
- Klassifizierung einfach nachzuvollziehen (keine Black-Box)

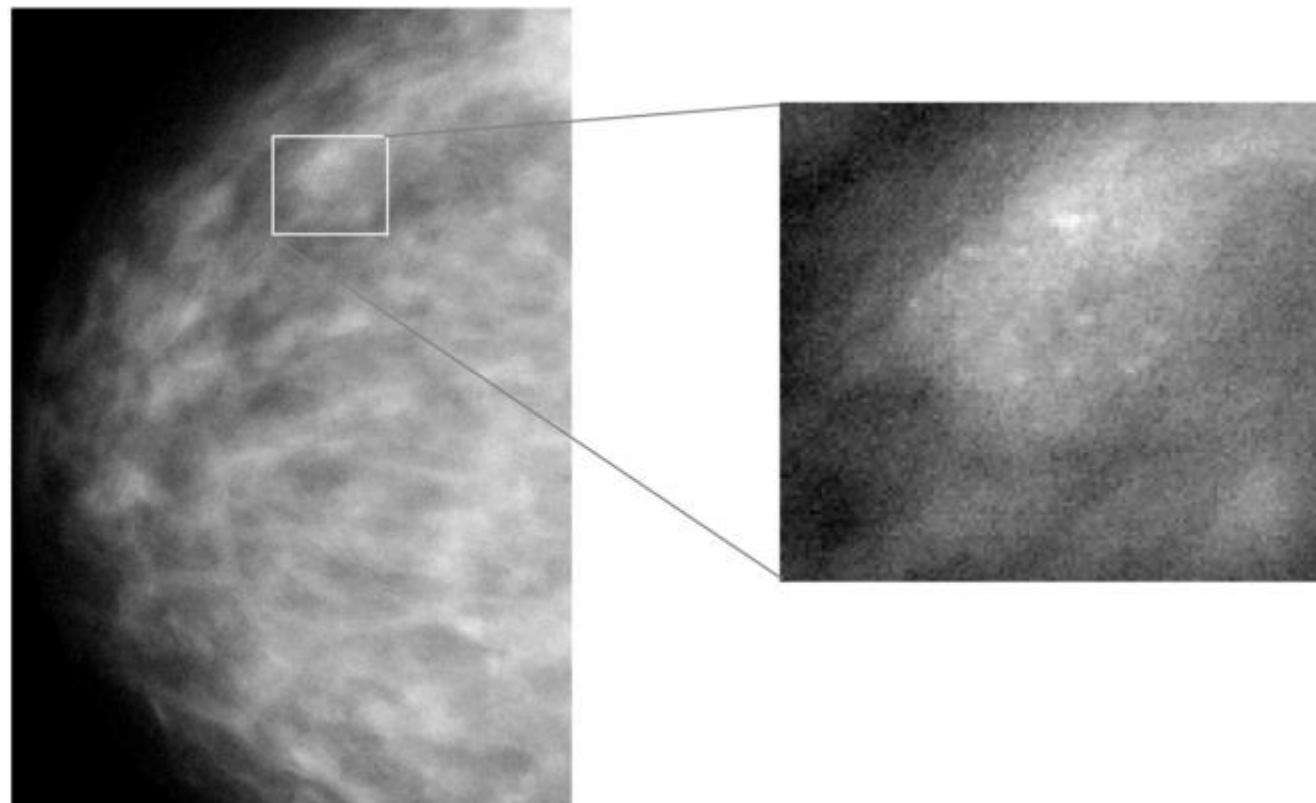


RDF: Q&A

Anwendungsbeispiele: SVM

El-Naqa et al., „A Support Vector Machine Approach
for Detection of Microcalcifications“

IEEE Transactions on Medical Imaging 2002

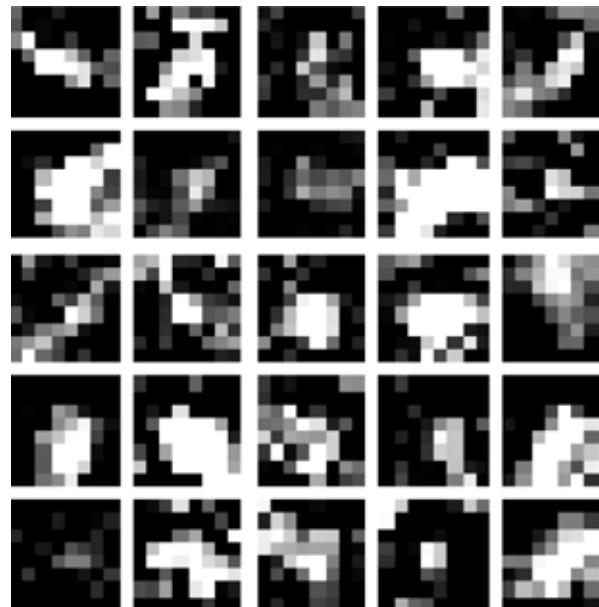


Mammografie & Detailansicht eines MC-Clusters

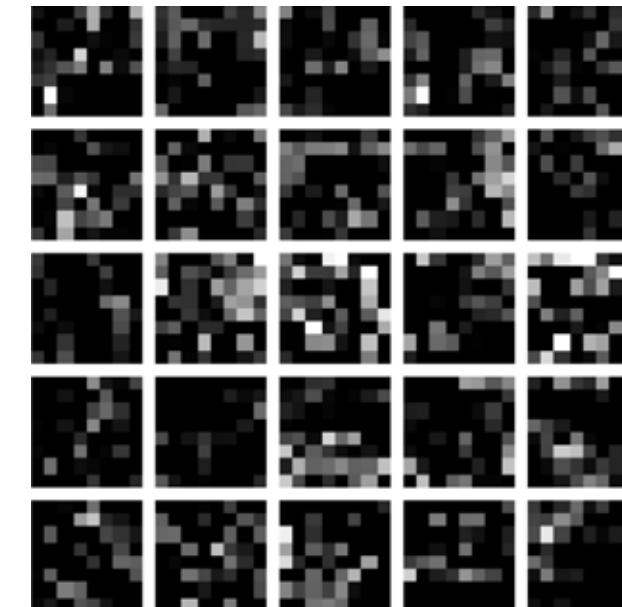
Anwendungsbeispiele: SVM

El-Naqa et al., „A Support Vector Machine Approach for Detection of Microcalcifications“

IEEE Transactions on Medical Imaging 2002



MC-clusters vorhanden

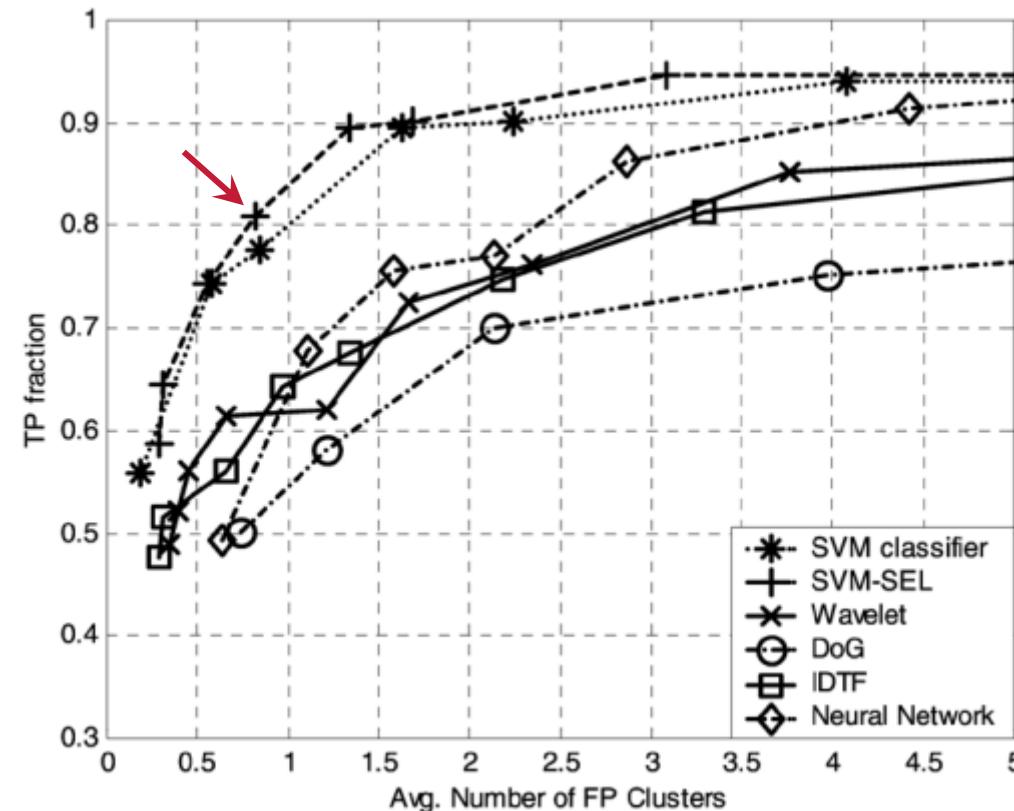


MC-clusters nicht vorhanden

Anwendungsbeispiele: SVM

El-Naqa et al., „A Support Vector Machine Approach for Detection of Microcalcifications“

IEEE Transactions on Medical Imaging 2002

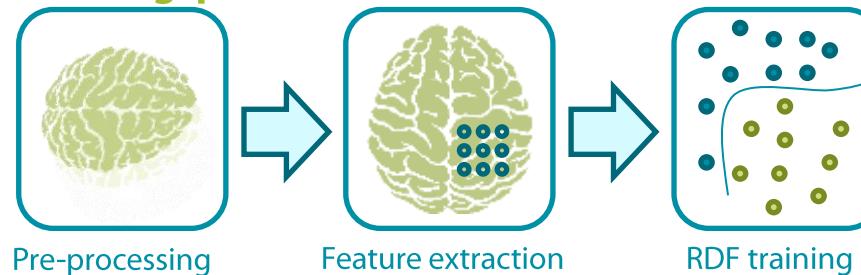


Anwendungsbeispiele: RDF

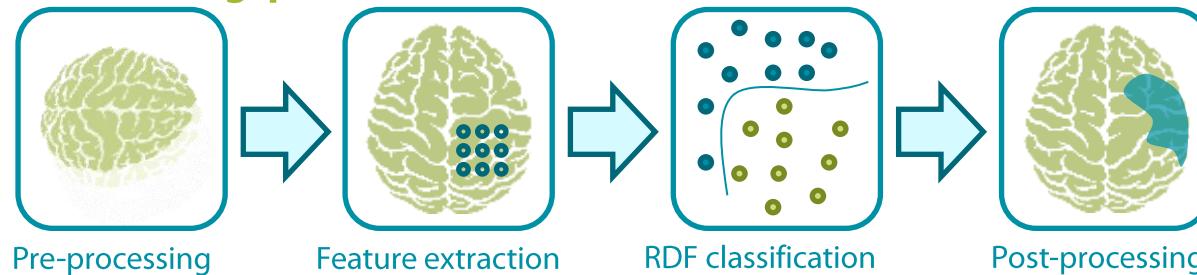
Maier et al., „Segmentierung von ischämischen Schlaganfall-Läsionen in multispektralen MR-Bildern mit RDFs“

Bildverarbeitung für die Medizin 2014

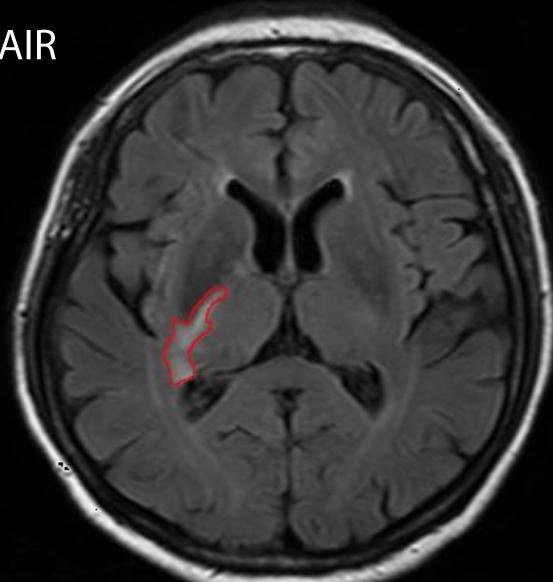
Trainingsphase:



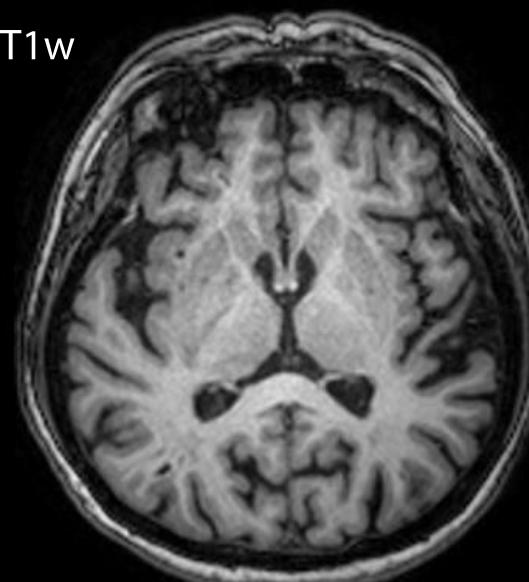
Anwendungsphase:



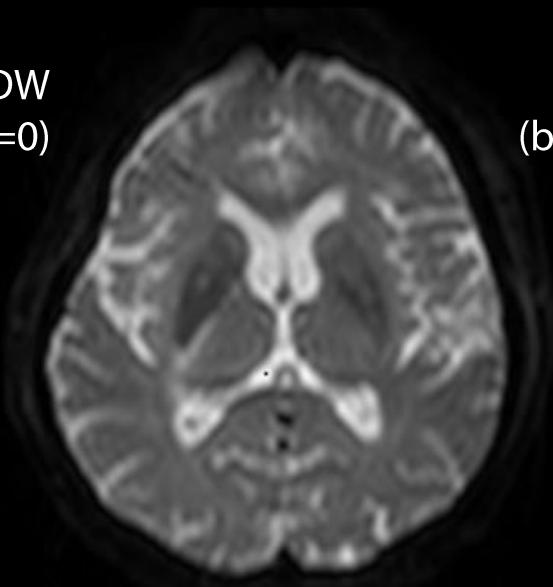
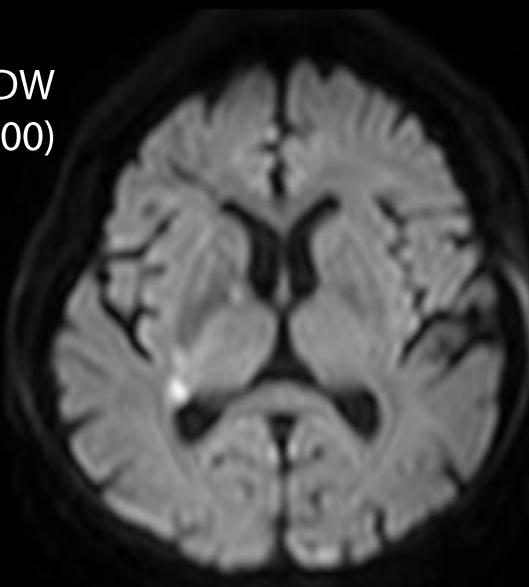
FLAIR



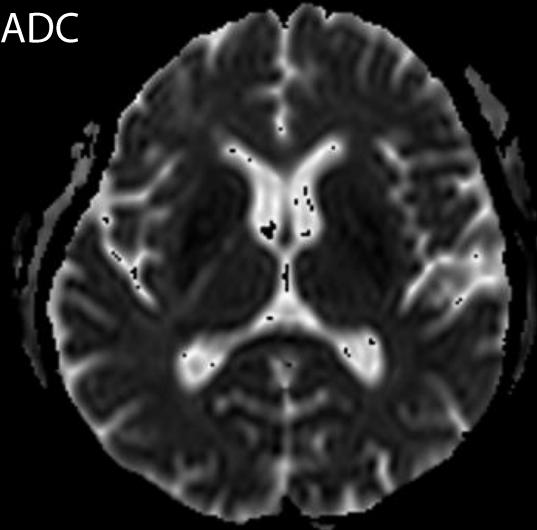
T1w



T2w

DW
($b=0$)DW
($b=1000$)

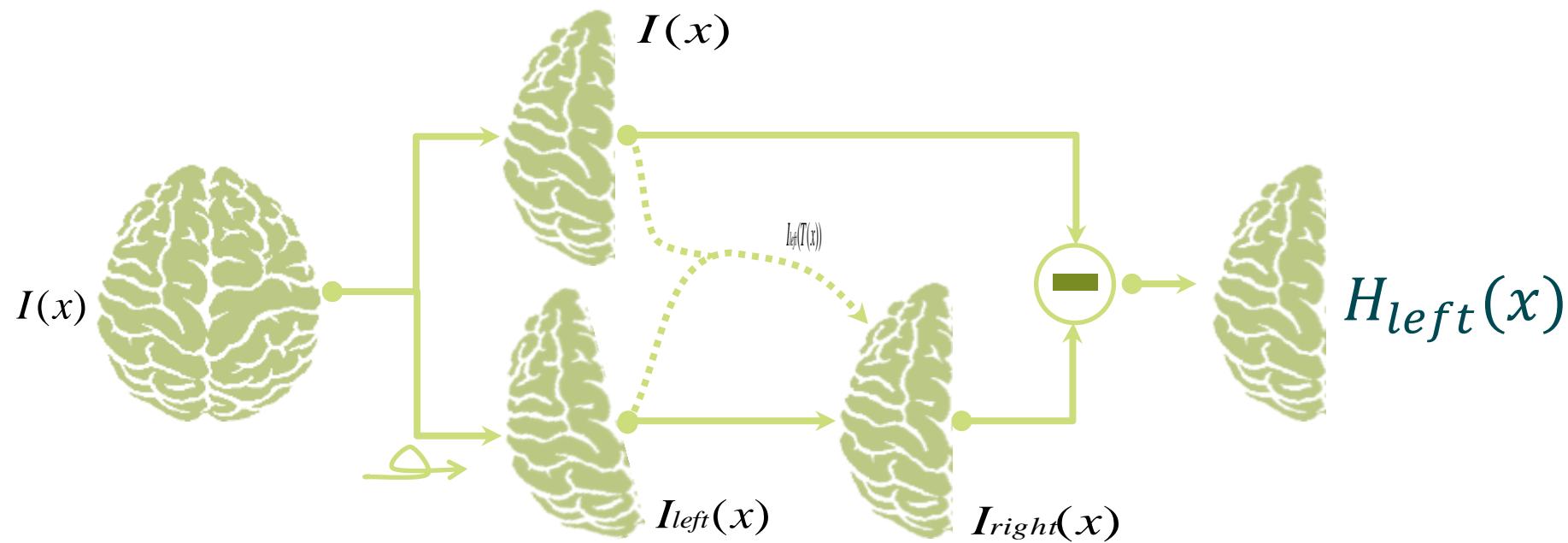
ADC



Anwendungsbeispiele: RDF

Maier et al., „Segmentierung von ischämischen Schlaganfall-Läsionen in multispektralen MR-Bildern mit RDFs“

Bildverarbeitung für die Medizin 2014

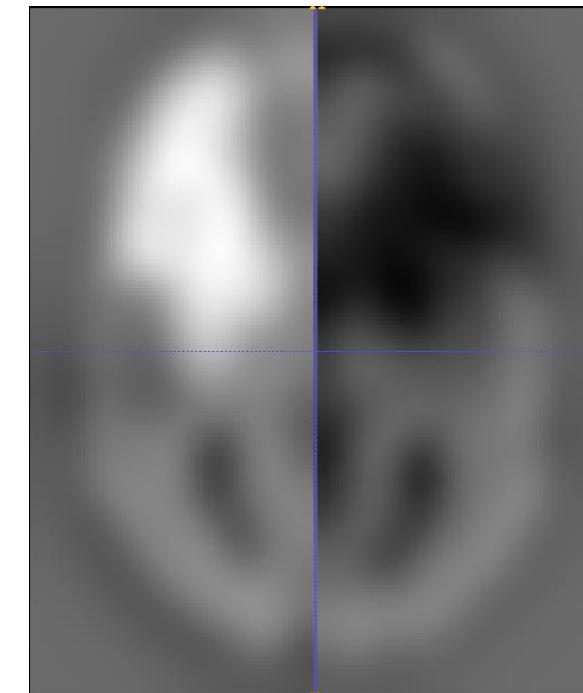
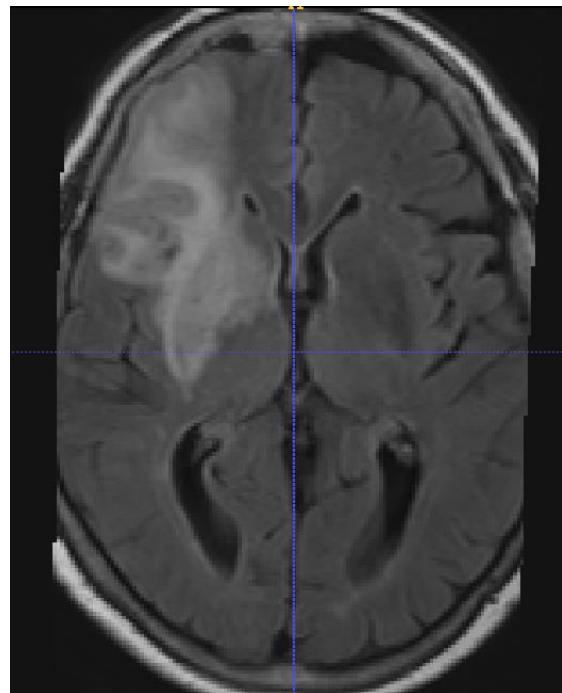


Hemisphärendifferenzmerkmal

Anwendungsbeispiele: RDF

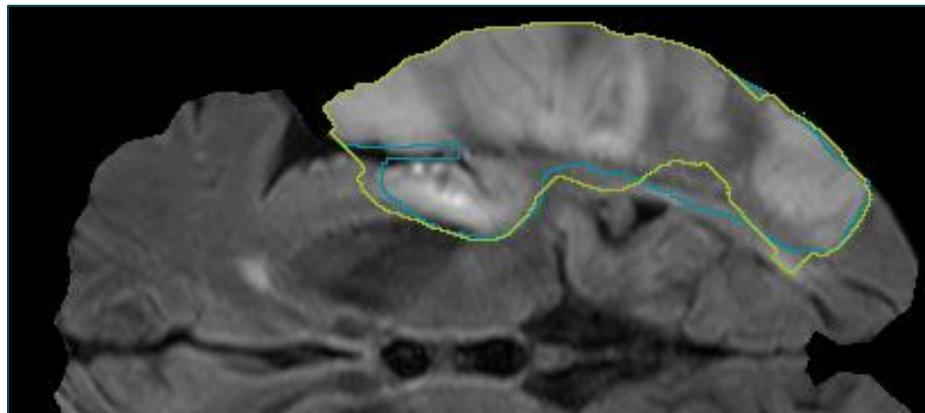
Maier et al., „Segmentierung von ischämischen Schlaganfall-Läsionen in multispektralen MR-Bildern mit RDFs“

Bildverarbeitung für die Medizin 2014



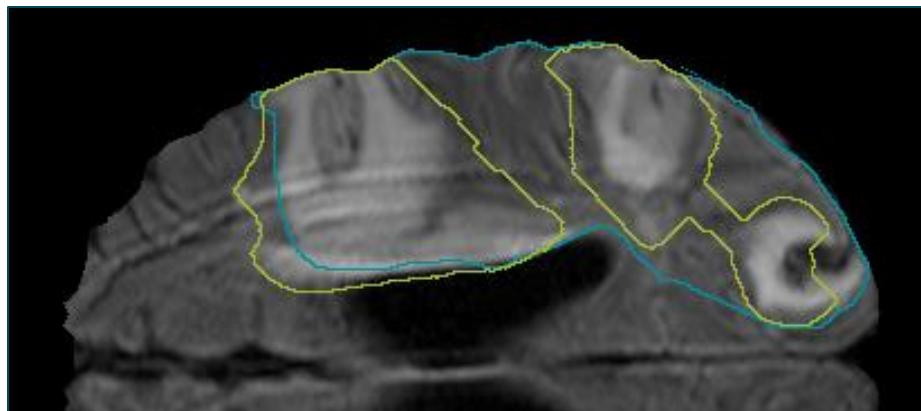
Hemisphärendifferenzmerkmal

Anwendungsbeispiele: RDF (Maier et al.)



Bester Fall

Fall Nr.	02
Dice Coeff.	0.874
Hausdorff Dist.	11.36 mm
Avg. Surface Dist.	1.55 mm

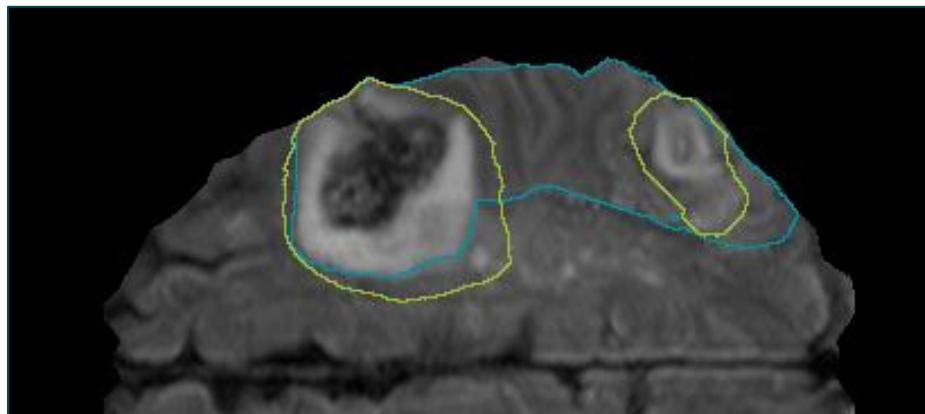


Schlechtester Fall

Fall Nr.	04
Dice Coeff.	0.780
Hausdorff Dist.	41.85 mm
Avg. Surface Dist.	3.57 mm

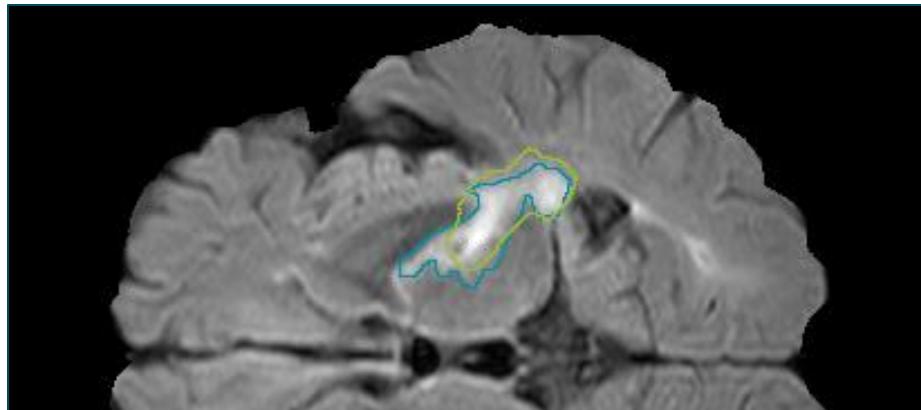
ground truth / segmentation

Anwendungsbeispiele: RDF (Maier et al.)



Haemorrhage

Fall Nr.	04
Dice Coeff.	0.780
Hausdorff Dist.	41.85 mm
Avg. Surface Dist.	3.57 mm



Kleinste Läsion

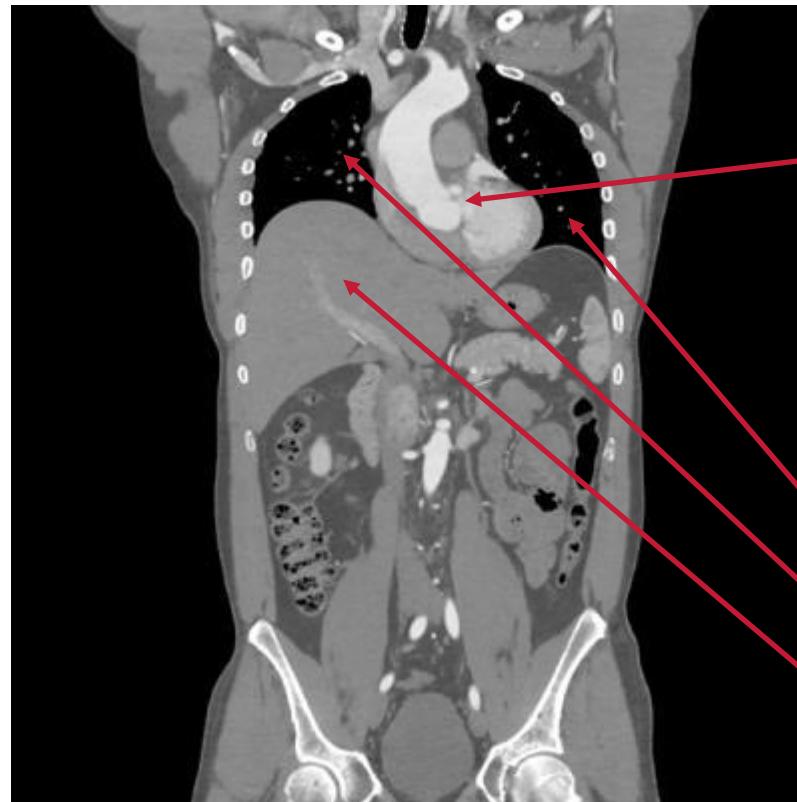
Fall Nr.	03
Dice Coeff.	0.706
Hausdorff Dist.	9.92 mm
Avg. Surface Dist.	1.22 mm

ground truth / segmentation

Anwendungsbeispiele: RDF

Criminisi et al., „Decision Forests with Long-Range Spatial Context for Organ Localization in CT Volumes“

MICCAI Workshop on Probabilistic Models for Medical Image Analysis 2009

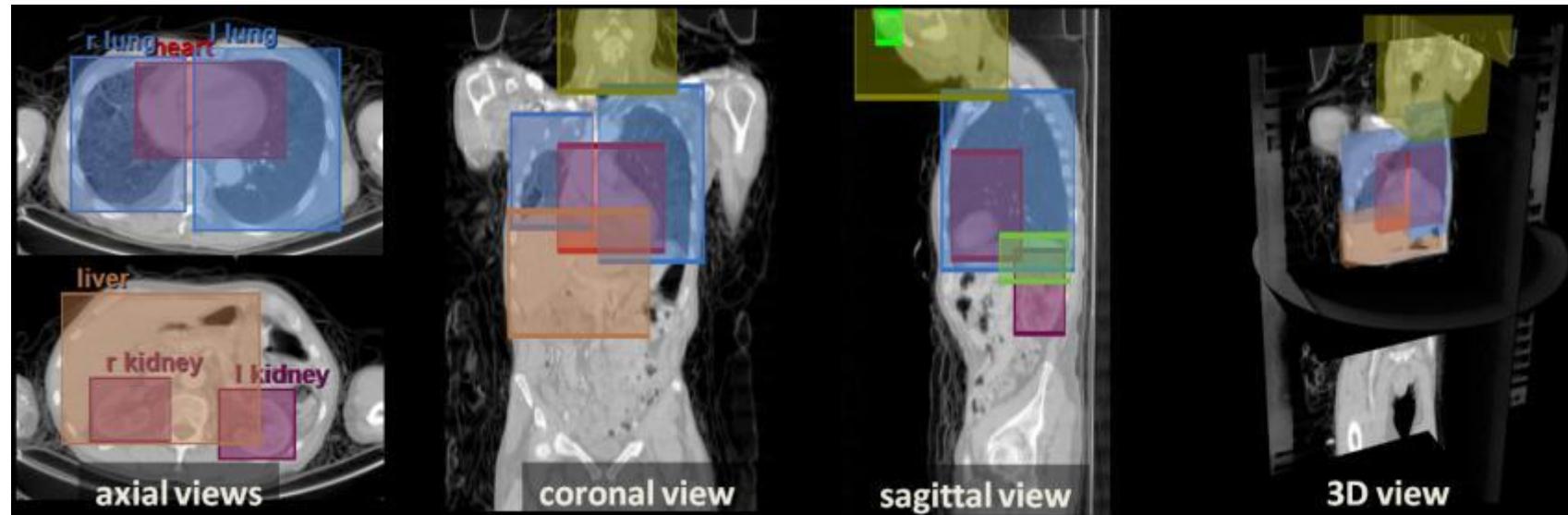


- head
- heart
- left eye
- right eye
- l. kidney
- r. kidney
- l. Lung
- r. Lung
- liver

Anwendungsbeispiele: RDF

Criminisi et al., „Decision Forests with Long-Range Spatial Context for Organ Localization in CT Volumes“

MICCAI Workshop on Probabilistic Models for Medical Image Analysis 2009

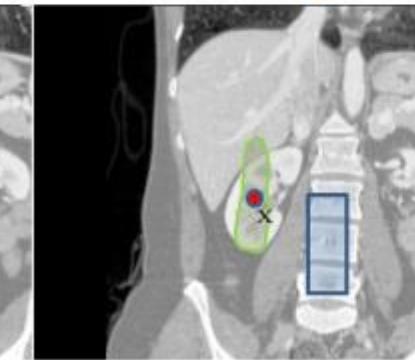
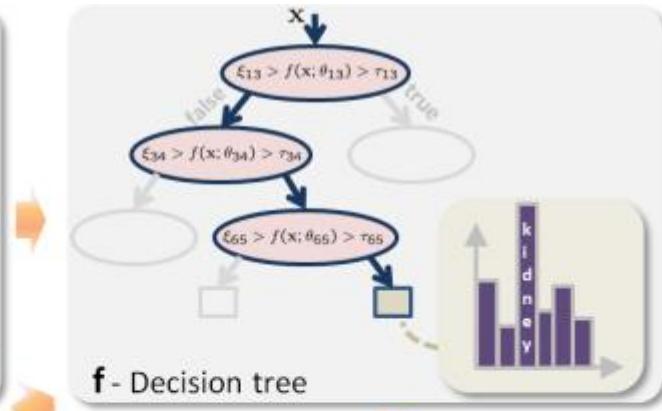
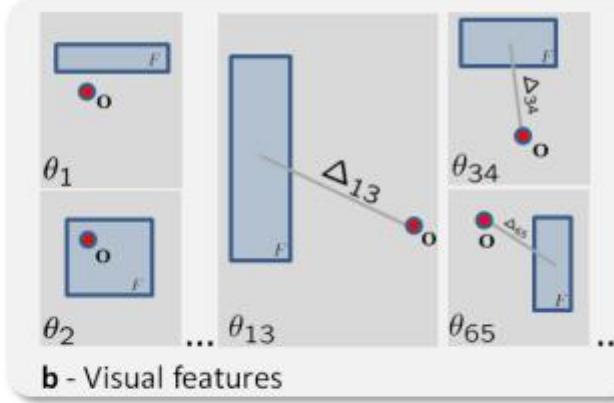


Trainingsdaten

Anwendungsbeispiele: RDF

Criminisi et al., „Decision Forests with Long-Range Spatial Context for Organ Localization in CT Volumes“

MICCAI Workshop on Probabilistic Models for Medical Image Analysis 2009

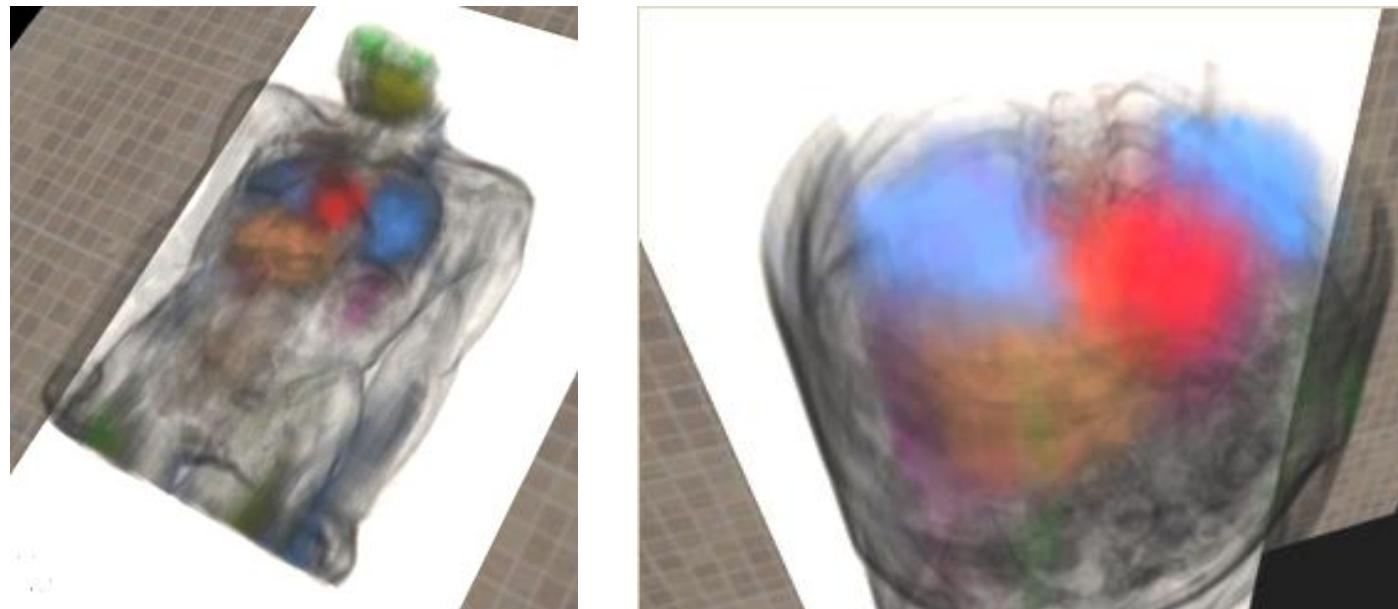


Merkmale

Anwendungsbeispiele: RDF

Criminisi et al., „Decision Forests with Long-Range Spatial Context for Organ Localization in CT Volumes“

MICCAI Workshop on Probabilistic Models for Medical Image Analysis 2009



Ergebnisse

Quellen & Literatur

Vorlesungen von Professor Yaser Abu-Mostafa @ Caltech

- 14 – SVM
 - <https://www.youtube.com/watch?v=eHsErIPJWUU>
- 15 - Kernel Trick & Soft-Margin:
 - <http://www.youtube.com/watch?v=XUj5JbQihIU>

Christopher J.C. Burges SVM Tutorial

- <http://research.microsoft.com/pubs/67119/svmtutorial.pdf>

Criminisis Arbeiten zu RDFs

- http://research.microsoft.com/pubs/155552/decisionForests_MSR_TR_2011_114.pdf

Einfache und machtvolle Bibliothek für Klassifizierer in Python

- <http://scikit-learn.org/>





Q&A

Ende