Karlsruher Institut für Technologie

Maschinelles Lernen II - Fortgeschrittene Verfahren
V02 Semi – überwachtes Lernen
Semi Supervised Learning (SSL)

Sommersemester 2017

Prof. Dr. J.M. Zöllner, Prof. Dr. R. Dillmann

INSTITUT FÜR ANGEWANDTE INFORMATIK UND FORMALE BESCHREIBUNGSVERFAHREN
INSTITUT FÜR ANTHROPOMATIK UND ROBOTIK

Inhalt

- Motivation und Begriffsklärung
  - Das SSL - Problem
  - Grundannahmen
  - Formalisierung
  - Induktiv ←→Transduktiv
- SSL - Methoden:
  - Self-Learning und Co-Training
  - Generative Modelle
  - Low-Densitty Separation
  - Graph basierte Modelle
  - Änderung der Repräsentation
- ....

Grundparadigmen

- Überwachtes Lernen
  - Gelabelte Trainingsdaten: Paare (X,Y)
  - Finde eine Funktion h die X (Merkmalsraum) auf Y abbildet (z.B. Klassen)
- Unüberwachtes Lernen
  - Ungelabelte Daten aus dem Merkmalsraum X
  - Strukturen und Labels der Daten (z.B. durch Cluster-Verfahren) finden
  - Oft auch Dichte-(Träger)-Schätzung (siehe ML I – SVM)

- Semi-Überwachtes Lernen
  - Einige, aber meist wenige gelabelte Lerndaten
  - Viele ungelabelte Daten
  - Finde eine Funktion h die X (Merkmalsraum) auf Y abbildet (z.B. Klassen)

Wieso benötigt man dafür eine Lösung?

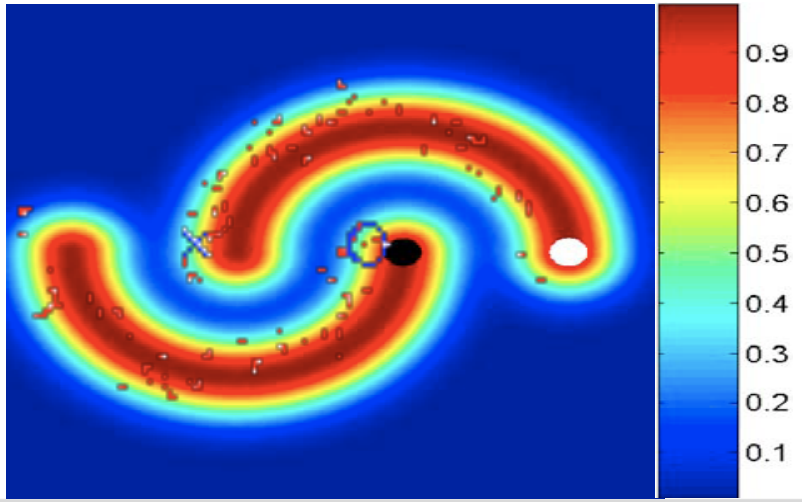
- Weil bessere Performanz mit minimalen Kosten („for free“) angestrebt wird

- Gründe:
  - Ungelabelte Daten sind billig z.B.
    - Sprache,
    - Kognitive Fzg. => Videodatenaufzeichnung rel. billig
    - Bilder/Videos im web , youtube, ...
  - Gelabelte Daten sind relativ teuer und schwer zu erzeugen
    - Annotation von Sprache: Beispiel 400h Annotation für ca. 1h Sprachdaten, ähnlich bei Videodaten etc.....
    - Manuelle Annotation ist teuer und zudem anstrengend
    - „Experten“ sind nötig
    - Annotationen sind auch fehleranfällig

Grundannahmen

- Typische Annahmen für das Lernen
  - Gleichmäßigkeit für überwachtes Lernen (Smoothness Assumption):
    - Wenn zwei Datenpunkte x1, x2 „nahe“ beieinander sind dann sollten auch die Ausgaben y1, y2 „ähnlich“ sein
  - Gleichmäßigkeit für Semi-überwachtes Lernen:
    - Wenn zwei Datenpunkte x1, x2 in einer dichten Region „nahe“ beieinander sind, dann sollten auch die Ausgaben y1, y2 „ähnlich“ sein

➔ wenn zwei Datenpunkte
durch einen Pfad hoher Dichte verbunden
sind (i.A. gehören dem gleichen Cluster an)
dann sind ihre Ausgaben ähnlich

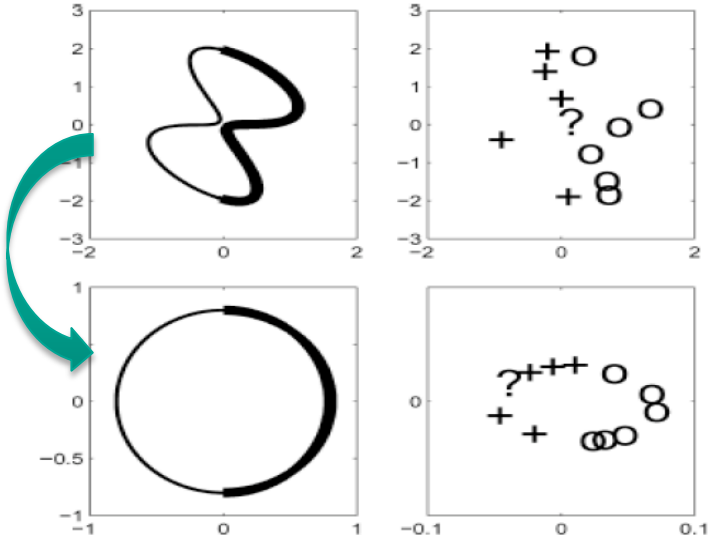


Abgeleitete Grundannahmen

- Cluster oder Dichte Annahme
  - Wenn zwei Datenpunkte im selben („dichten“) Cluster sind, dann sind sie in derselben Klasse
  - ➔ eine Trennung sollte in einer Region niedriger Dichte (zw. den Clustern) liegen

- Manigfaltigkeit-Annahme (Manifold Assumption)
  - Hochdimensionale Daten haben eine Abbildung in einen i.A. anders dimensionalen Raum (Manigfaltigkeitsraum) in dem sich ihre Strukturen abbilden (unterscheiden/ erhalten)

  - ➔ Dieser Raum kann dann für die Berechnung des geodäsischen Abstand benutzt
  - ➔ approximative Implementierung der Gleichmäßigkeitsannahme



Wie könnte SSL realisiert werden

- Möglichkeiten
  - SSL als Erweiterung des überwachten Lernens
  - ➔ SSL ist überwachtes Lernen mit zusätzlichen Informationen über die Verteilung der Daten in X
  - SSL als Erweiterung des unüberwachten Lernens
  - ➔ SSL ist unüberwachtes Lernen mit zusätzlichen Einschränkungen und Informationen über die Cluster/Strukturen

Das SSL - Problem

- SSL-Problem:
  - Verwendung von gelabelten und ungelabelten Daten um bessere Hypothesen zu erzeugen

- Bessere Hypothesen heißt ➔ 2 untere Güte-Schanken:
  - SSL kann und sollte verglichen werden mit (und besser sein als):
    - Ergebnis des überwachten Lernens mit ausschließlich gelabelten Daten
    - Ergebnis des unüberwachten Lernens mit allen Lerndaten (ohne Labels)
  - Vergleich mit diesen beiden unteren Schranken ➔ Informationen darüber, ob die Annahme, die bei dem semi-überwachten Lernen gemacht wird gilt oder ob der verwendete Ansatz (Modell) diese verletzt

- Leider:
  - „as we all know - there is no free lunch...“

  ➔ Vorwissen oder korrekte Annahmen nötig

Formalisierung

- Instanzen (Feature – Vektor): $x \in X$
  label: $y \in Y$
- Hypothese: $h: \; X \rightarrow Y$
- Gelabelte Daten $(X_l, Y_l) = \{(x_{1:l}, y_{1:l})\}$
- Ungelabelte Daten: $X_u = \{x_{l+1:n}\}$
  vorhanden beim Trainieren
- Üblicherweise gilt: $l << n$
- Neue Daten: $X_{test} = \{x_{n+1:....}\}$
  nicht vorhanden beim Trainieren

Induktiv vs. Transduktiv

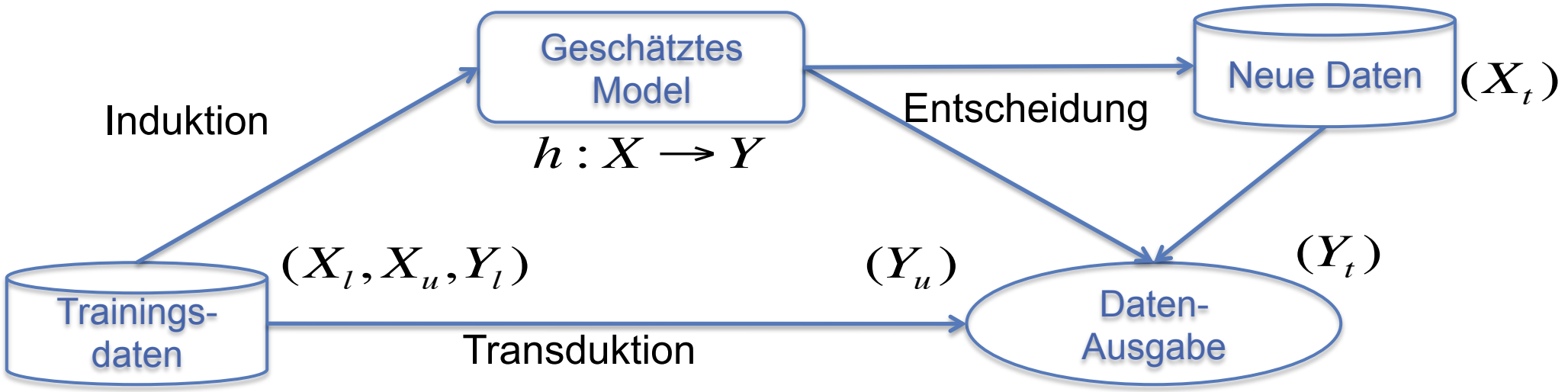
- Gegeben die Daten
  - Gelabelte und ungelabelte Lerndaten
  - Ungelabelte Daten

- Induktives Lernen (d.h. auch semi - überwachtes Lernen):
  - Ziel ist das Schätzen einer Hypothese

$$h: \quad X \rightarrow Y$$

  - Die auch unbekannte Daten „gut“ abbildet
- Transduktives Lernen:
  - Ziel ist das Labeln der ungelabelten Daten
  - (kann! auch das Labeln neuer Daten sein)
  - Das Finden einer Hypothese ist nicht das Ziel (kann aber erfolgen)

Induktion (Deduktion) Transduktion



[Learning from Data: Concepts, Theory and Methods.
V. Cherkassky, F. Mulier. Wiley, 1998.]

- Vorsicht: einige Verfahren heißen zwar „transductive ...“ sind aber eher induktiv (z.B. die ursprüngliche transductive SVM)

SSL – X

- Unterschiedliche Problemstellungen:
  - Überwachtes Lernen (Klassifikation/Regression) $\{(x_{1:n}, y_{1:n})\}$
  - Semi-überwachte Klassifikation/Regression $\{(x_{1:l}, y_{1:l}), x_{l+1:n}, x_{test}\}$
  - Transduktive Klassifikation/Regression $\{(x_{1:l}, y_{1:l}), x_{l+1:n}\}$
  - Semi-überwachtes Clustern $\{x_{1:n}, \text{must-links}, \text{cannot-links}\}$
  - Unüberwachtes Lernen $\{x_{1:n}\}$

- Fokus in MLII auf Algorithmen für die Klassifikation

Verschieden Ansätze

- Erste Algorithmen
  - Self-Training & Co-Training
- Generative probabilistische Modelle (Generative Probabilistic Models)
  - EM for Gaussian Mixtures
- Dichte Trennung (Low-Density Separation)
  - „Transduktive“ SVM
- Graph basierte Modelle / Methoden
  - Methoden bei denen die Daten als Knoten eines Graphs repräsentiert sind und die Kanten die jeweiligen Abstände enthalten
- Änderung der Repräsentation
  - unüberwachtes Lernen (z.B.: Clustern) um neue (i.A. niedrig dimensionale) Repräsentationen der Daten zu erhalten
  - Lernen der Zuordnung der Cluster zu Klassen

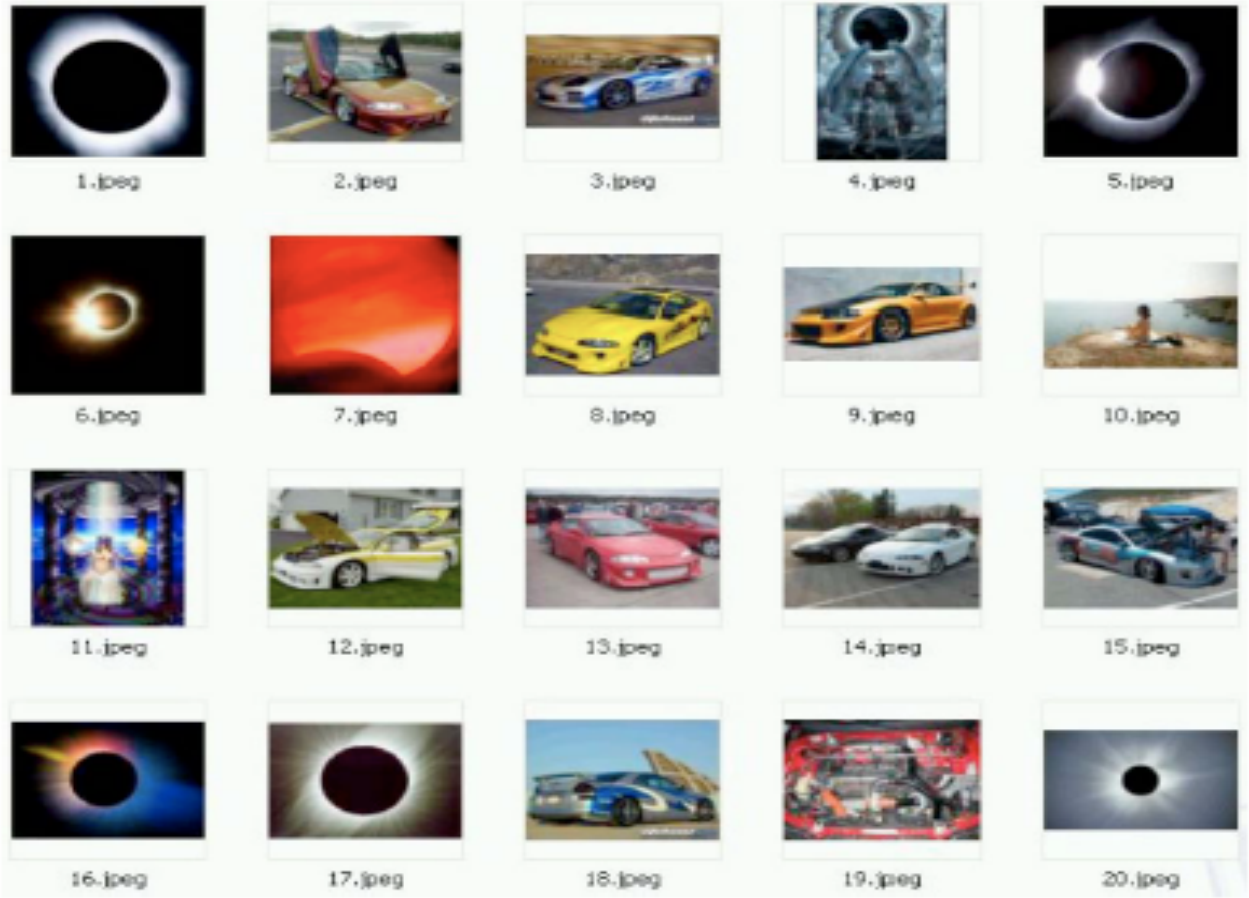
Selbst-Lernen (Self – Training)

- Grundidee: Sukzessive Verwenden der ungelabelten Daten, die durch die gelernte Hypothese eine hohe Konfidenz der Prädiktion erreichen
  (Konfidenz ~= Wahrscheinlichkeit der Klassenzugehörigkeit)
- Grundalgorithmus – Wrapper
- Wiederhole
  - Trainieren mit gelabelten Daten $h \leftarrow (X_l, Y_l)$
  - Auswerten der ungelabelten Daten $h(x), \ \forall x \in X_u$
  - Einfügen konfidenter Daten zu den Trainingsdaten $(X_l, Y_l) \leftarrow (X_l, Y_l) \cup \{(x, \ h(x)) | x \in X_u, h(x) - hohe \ \ Konf.\}$

- Variationen – Hinzunehmen neuer Daten
  - Nur konfidente neue Beispiele
  - Alle Beispiele
  - Mit Gewichtung anhand der Konfidenz (🡐 Lernverfahren muss dies erlauben)

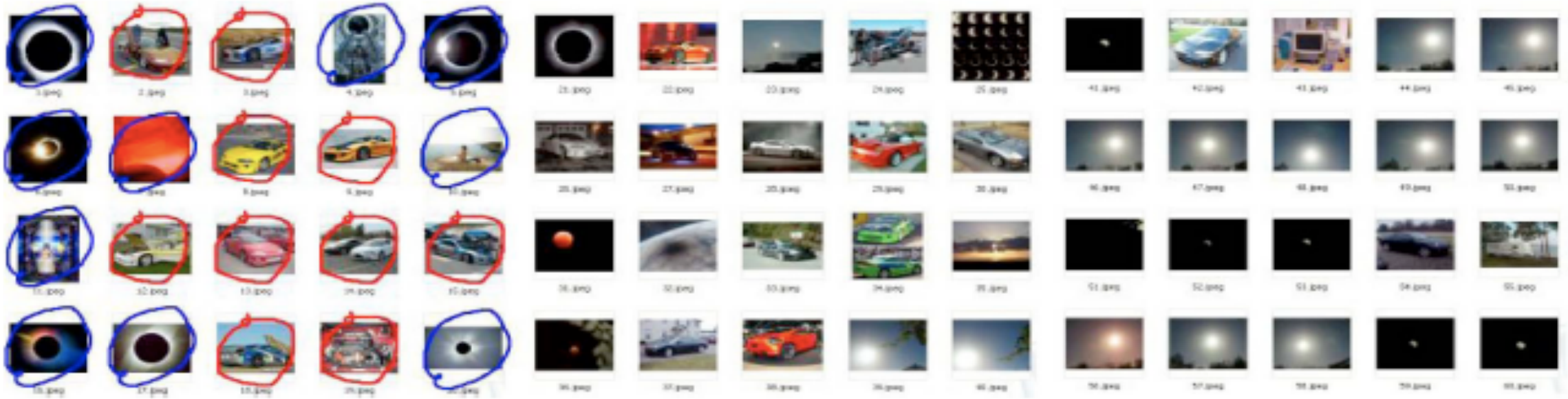
Beispiel

- Ziel : Klassifikation Sonnenfinsternis



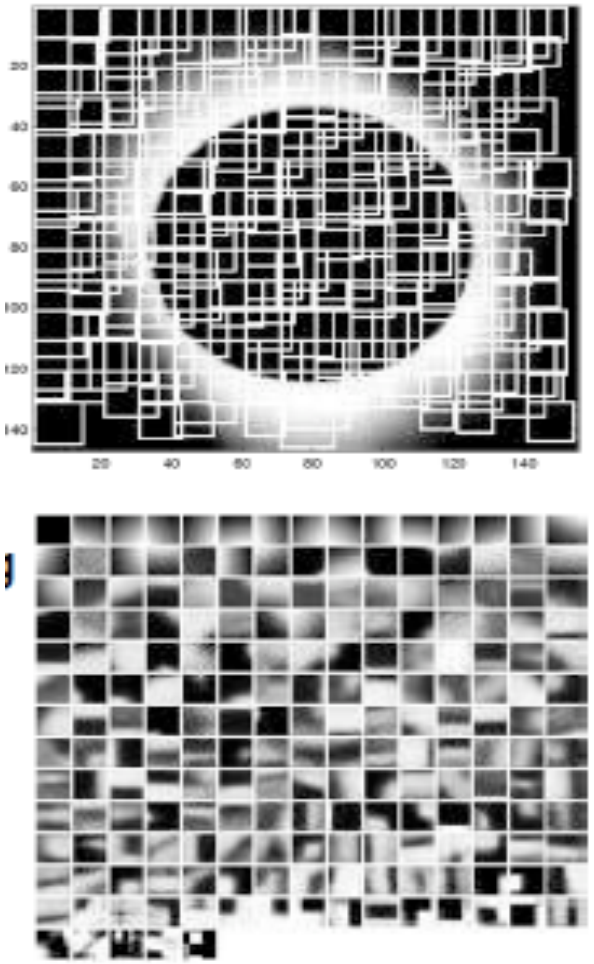
Beispiel

- Gegeben
  - Einige gelabelte Daten: Blau – Sonnenfinsternis, rot – andere
  - Viele ungelabelte Daten



Beispiel – nur zur Verdeutlichung

- Repräsentation (Feature Vektor)
  - Jedes Bild wird in kleine patches aufgeteilt
  - 10x10 Grid, zufällige Größe zw.10-20 pixels
  - (Normalisieren der patches)
- Definiere 200 Datensätze (“visual words”)
  - Clustern z.B. mit k-means clustering
  - Klassifiziere jeden patch durch seinen Clustervertreter (visual word)
- Bilder werden repräsentiert durch einen Satz (Vektor) = Häufigkeiten solcher „Visuellen Wörtern“
- Verwende neue Repräsentation (Vektor)

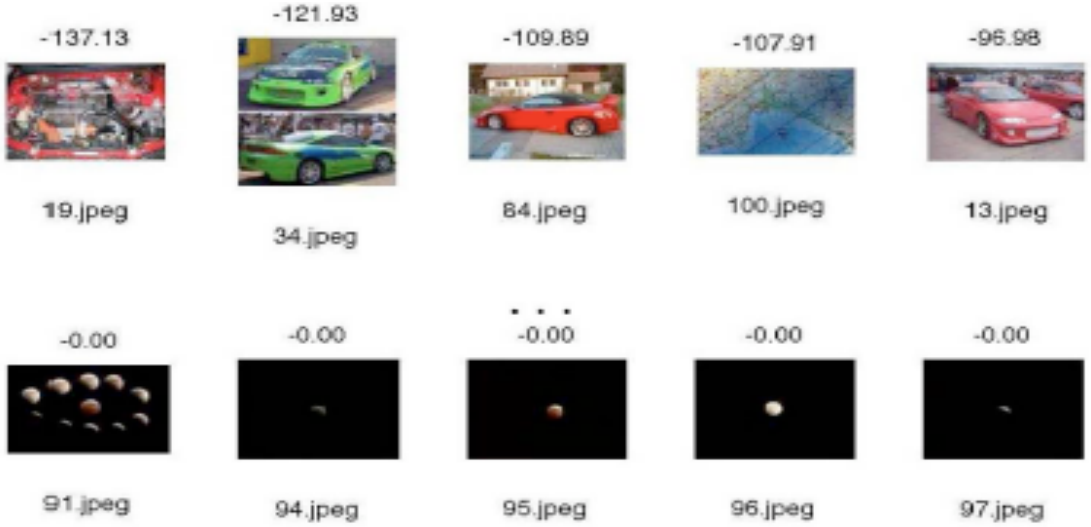


(Lässt sich auch anders realisieren)

Beispiel

- Trainiere naiven Bayes-Klassifikator auf den ursprünglichen 2 Bildern

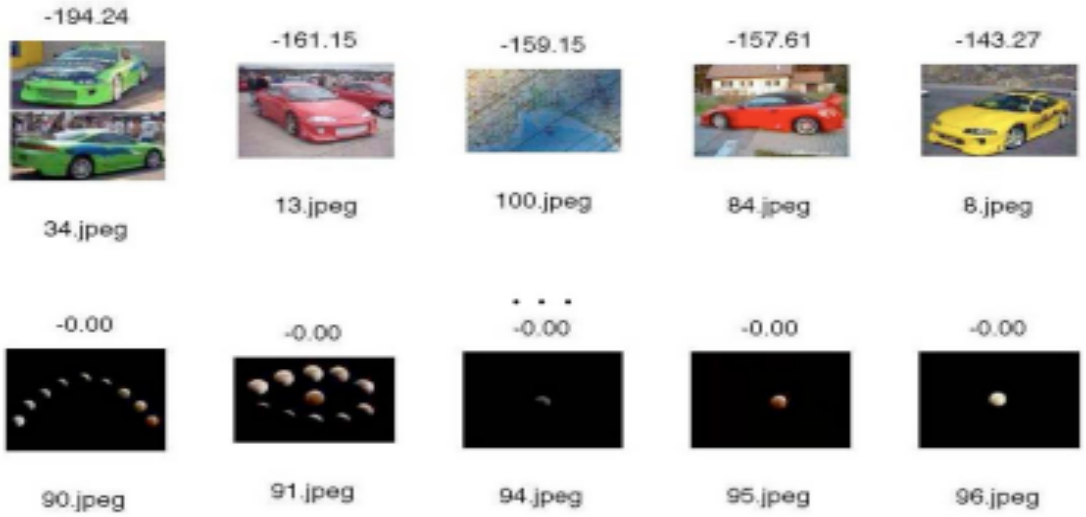
- Klassifiziere ungelabelte Daten,
  sortiert nach Konfidenz ← $log\ p(y=Sonnenfinsternis \mid x)$



Beispiel

- Hinzufügen der Daten zu den Lerndaten (mit Klassifikation)

- Neu-Trainieren des Klassifikators und wiederholen



Selbst – Lernen Diskussion

- Weit verbreitet und vermutlich ältester SSL – Ansatz (1965 -70)
  - Wrapper Algorithmus anwendbar für alle überwachten Lernmethoden
  - Startet auf gelabelten Daten
  - In jeder Iteration werden ungelabelte Daten gelabelt, abh. von der Entscheidungsfunktion
  - Bezeichnungen: self-learning, self-labeling, decision-directed learning

- Diskussion
  - Kann effektiv sein
  - Aber es ist nicht genau bestimmt, wie das Ergebnis ist (und abhängig von der Methode des überwachten Lernens)
  - Nicht methodisch festgelegt, welche Annahmen über das Problem getroffen werden

Selbst – Lernen Diskussion

- Vorteile
  - Sehr einfache semi-überwachte Lernmethode
  - Wrapper – passend zu existenten auch komplexen Klassifikatoren /etc...
  - Oft angewandt in realen Anwendungen wie z.B. Sprachanalyse

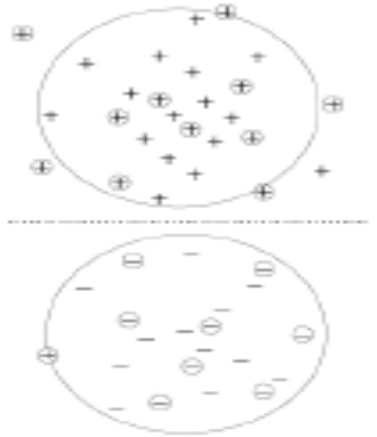
- Nachteile
  - Frühe Fehlentscheidungen können sich verstärken
    - Heuristische Lösung: Daten „un-labeln“ sofern ihre Konfidenz unter einen Schwellwert fällt
  - Generelle Analyse kompliziert
    - Nur für Spezialfälle ist eine geschlossen, formale Analyse möglich
    - In Spezialfällen entspricht Selbst-Lernen dem EM – Ansatz (siehe GMM)

Mit-Lernen (Co – Training)

- Idee (nach Blum & Mitchell ’98)
  - Annahme:
    - Features können in 2 Mengen aufgeteilt werden (Feature-Split) somit gilt für jeden Vektor: $x = \left[x^{(1)}, x^{(2)}\right]$
    - Jede Untermenge ist ausreichend um eine gute Lernmaschine (im weiteren Klassifikator) einzutrainieren

$X^{(1)} - view$

$X^{(2)} - view$



  - Wichtig: die Featurevektoren $x^{(1)}, x^{(2)}$ sind voneinander unabhängig

Mit-Lernen (Co – Training)

- Lern - Ansatz
  - Verwende den Wrapper – Ansatz
  - 2 unabhängige Klassifikatoren werden auf je einer Featuremenge trainiert und auf die ungelabelteten Daten angewendet

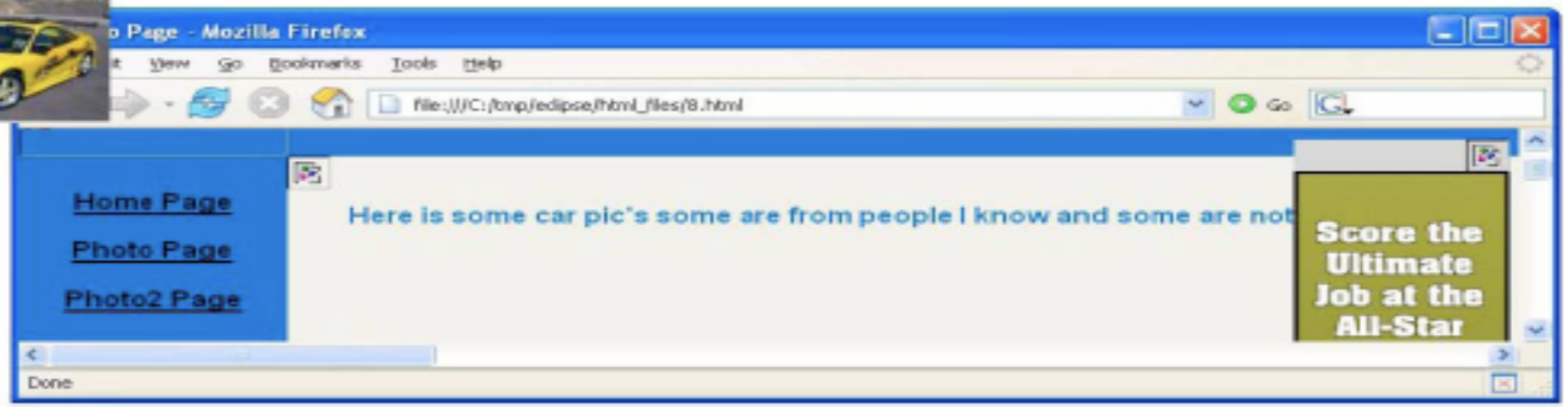
$$\left(X_l^{(1)}, Y_l\right) \rightarrow h^{(1)} \qquad \left(X_l^{(2)}, Y_l\right) \rightarrow h^{(2)}$$

  - Wenige Daten hoher Konfidenz (mit oder ohne Übereinstimmung der Klassifikatoren) werden zu dem jeweilig anderen Lerndatensatz hinzugenommen
  - Neutrainieren der Klassifikatoren mit dem erweiterten Datensatz
  - Wiederholen .....

- Erweiterung
  - Demokratisches Co-Training: Erweiterung zu mehr als 2 Basisklassifikatoren
    ➔ Mehrheitsentscheidung nötig um neue Daten hinzuzufügen

Co-Training Beispiel

- Bild und Textklassifikation von Webseiten



Co-Training Beispiel

- Merkmalsmengen
  - Jede Instanz wird durch zwei Mekmalsvektoren dargestellt

$$x = \left[x^{(1)}, x^{(2)}\right] = \left[Bildvektor,\ Webseiten - Text - Merkmale\right]$$

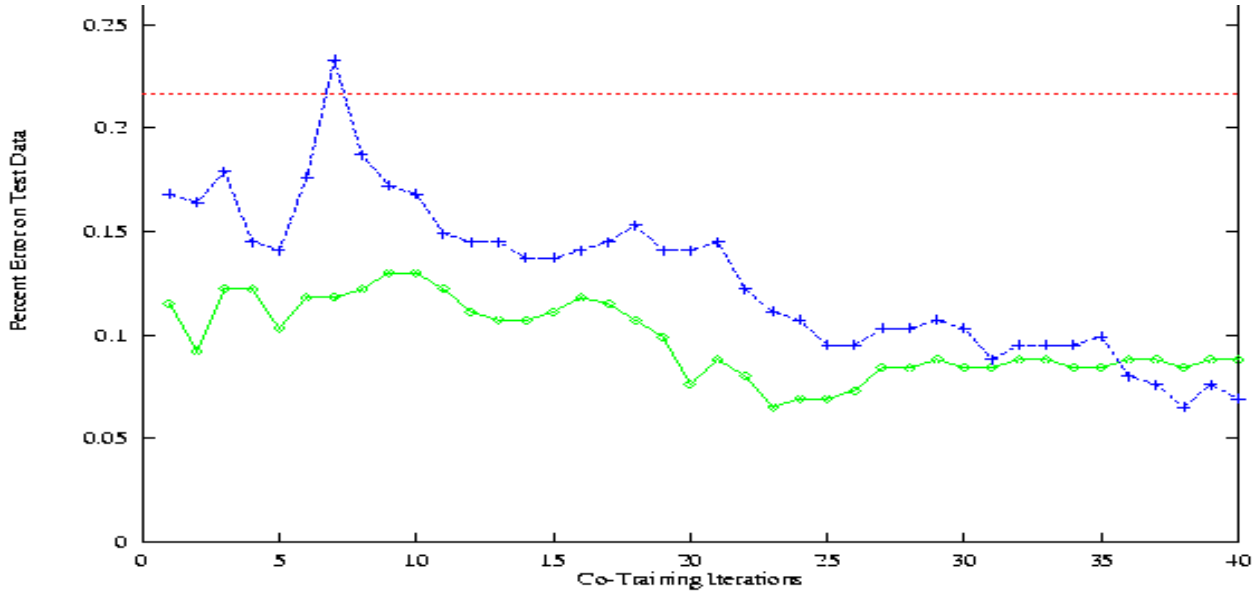
  - Natürliche Untermengen (multiple – Sicht)
- Umsetzten der Co-Training Idee
  - Verwenden: einen Bildklassifikator und einen Textklassifikator
  - Gegenseitig Zulieferung neuer Lerndaten
- Auswertung

$$h^{(1)}(x^{(1)}),\quad h^{(2)}(x^{(2)})$$

oder

$$h^{(1)}(x^{(1)}) = h(x) = h^{(2)}(x^{(2)})$$

$$h(x) = h^{(1)}(x^{(1)}) * h^{(2)}(x^{(2)})$$



Co-Training Diskussion

- Vorteile
  - Wrapper Methode – anwendbar auf alle existierenden Klassifikatoren
  - Weniger anfällig für Missentscheidungen als Selbstlernen

- Nachteile
  - „Natürliche“ Featureaufteilung ggf. nicht vorhanden
  - Modelle die die vollständige Featuremenge benutzten erreichen oft bessere Ergebnisse

Co-Training Varianten

- Fake Feature Split
  - Zufällige, künstliche Aufteilung der Merkmale
  - Co-Training wie bisher
- Multi-View-Ansatz
  - Kein Feature Split
  - Trainiere mehrere Klassifikatoren
  - Klassifizierung der ungelabelten Daten mit allen Klassifikatoren
  - Verwende Mehrheitsentscheidung für neue Labels
- CO-EM
  - Nutzung aller Daten
  - Jeder Klassifikator labelt die Daten X_u probabilistisch
  - Daten (x,y) werden probabilistisch gewichtet genutzt, mit Gewicht $p(y|x)$

Anwendungsbeispiel (Schiele 2008)

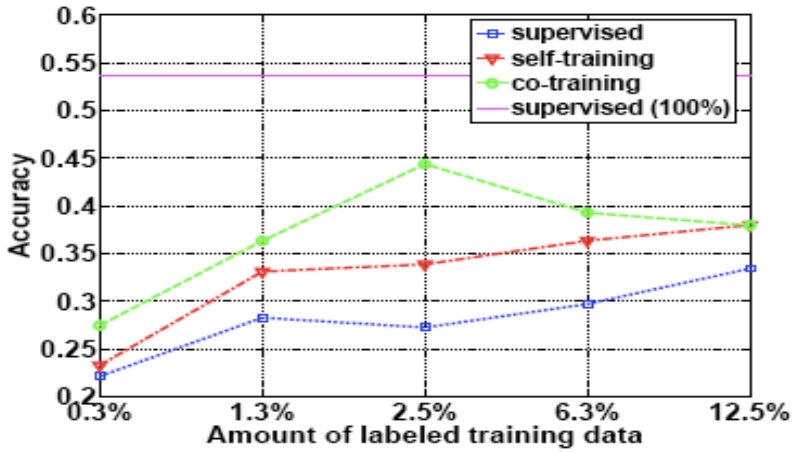
Case Study: Recognition of ADL (Activities of Daily Living)

- Dataset called PLCouple1 from PlaceLab - MIT
  - data from a couple - living 10 weeks in PlaceLab-appartment
  - ‘normal’ daily routine
  - 104 hours are annotated from 15 separate days
  - publicly available only 68 hours from 9 separate days
  - 9 activities studied
    - actively watching TV, dishwashing, eating, grooming, hygiene, meal preparation, reading paper/book/magazine, using computer, using phone
  - two different modalities used
    - room-level movement (infrared sensor)
    - 3 3D-accelerometers (worn by male)
  - experimental protocol: leave-one-day-out cross-validation

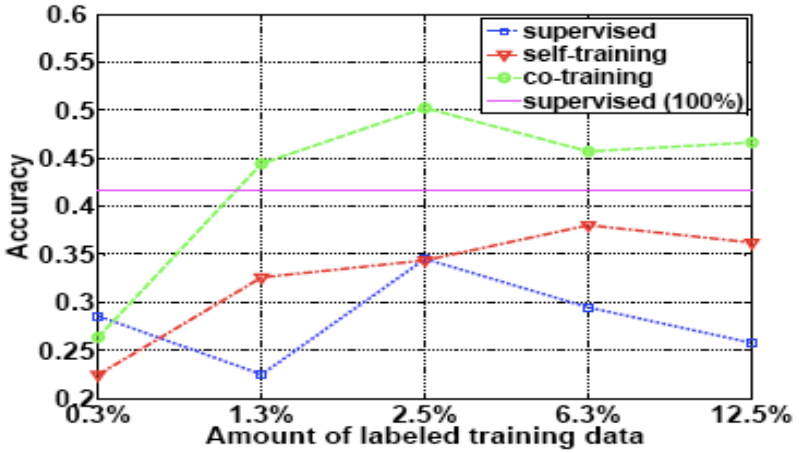
Anwendungsbeispiel

Case Study: Recognition of ADL (Activities of Daily Living)

- Reduce annotation effort to recognize ADLs
  - Semi-Supervised Learning of Activities



(a) Results based on acceleration data



(b) Results based on infra-red data

  - Best Result
    - acceleration: (53.6% supervised) - (44.4% co-training)
    - infrared: (41.6% supervised) - (50.3% co-training)

Anwendungsbeispiel

Case Study: Recognition of ADL (Activities of Daily Living)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Active Learning and Sampling of Labels
  - start with 1.3% labels (124 labels for 9 activities)
  - let user label data if
    - a, both classifiers have low score
    - b, the classifiers contradict each other

Acceleration				Infra-red			Combined		
Labeled	Supervised	Active - low scores	Active - conflicts	Supervised	Active - low scores	Active - conflicts	Supervised	Active - low scores	Active- conflicts
12.5%	33.4%	54.3%	56.0%	25.8%	45.9%	30.2%	34.3%	62.8%	61.0%
6.3%	29.7%	58.7%	56.9%	29.5%	40.9%	30.9%	34.3%	65.4%	59.5%
2.5%	27.3%	52.5%	50.3%	34.5%	33.3%	41.2%	36.0%	57.2%	56.6%
1.3%	28.3%	46.1%	46.5%	22.5%	40.3%	32.3%	27.1%	54.9%	52.5%

  - Best result:
    - acceleration: (53.6% supervised) - (58.7% active learning)
    - infrared: (41.6% supervised) - (45.9% active learning)
    - combined: (65.4% active learning)

Anwendungsbeispiel

Case Study: Recognition of ADL (Activities of Daily Living)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Comparison each sensor modality
  - active learning most profitable
  - co-training only profitable for infrared - as its profits from the better performance of the accelerometers

Acceleration		Infra-red	
Active - low scores	58.7%	Co-training	50.3%
Active - conflicts	56.9%	Active - low scores	45.9%
Supervised	53.6%	Supervised	41.6%
Co-training	44.4%	Active - conflicts	41.2%
Self-training	38.0%	Self-training	38.0%

- Combination of both sensor modalities
  - active learning: 65.4%

Verschieden Ansätze

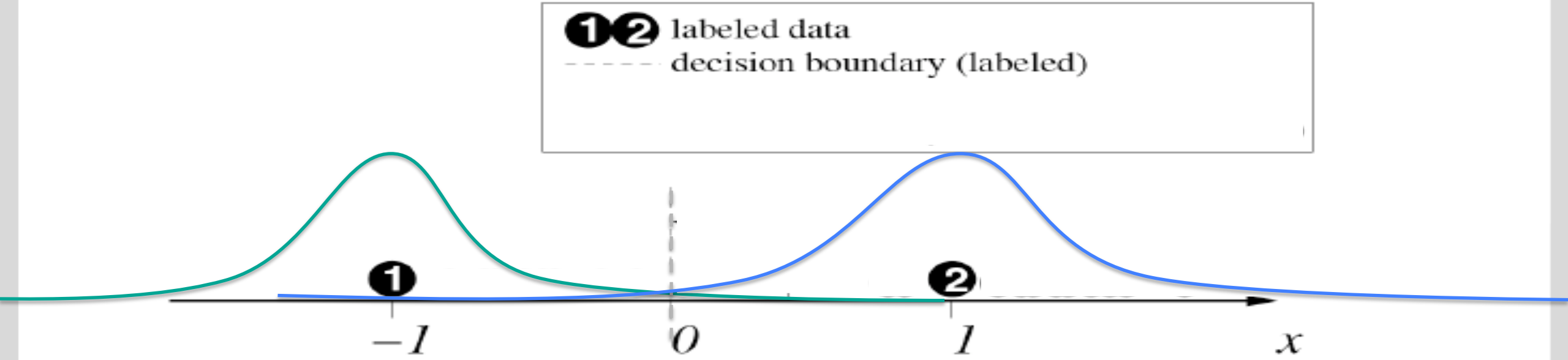
- Erste Algorithmen
  - Self-Training & Co-Training
- Generative probabilistische Modelle (Generative Probabilistic Models)
  - EM for Gaussian Mixtures
- Dichte Trennung (Low-Density Separation)
  - „Transduktive“ SVM
- Graph basierte Modelle / Methoden
  - Methoden bei denen die Daten als Knoten eines Graphs repräsentiert sind und die Kanten die jeweiligen Abstände enthalten
- Änderung der Repräsentation
  - unüberwachtes Lernen (z.B.: Clustern) um neue (i.A. niedrig dimensionale) Repräsentationen der Daten zu erhalten
  - Lernen der Zuordnung der Cluster zu Klassen

Generative Modelle / Methoden

- Generative Algorithmen nutzen eine Schätzung der Verteilung der Daten für die Klassen

➔ Zusätzliche Information der Verteilung der Daten sind sinnvoll !!

- Ausgehend davon, dass Klassen kohärente Daten enthalten z.B. normalverteilt ➔ Geschätzte Entscheidungsgrenze wandert je nach Verteilung der Daten

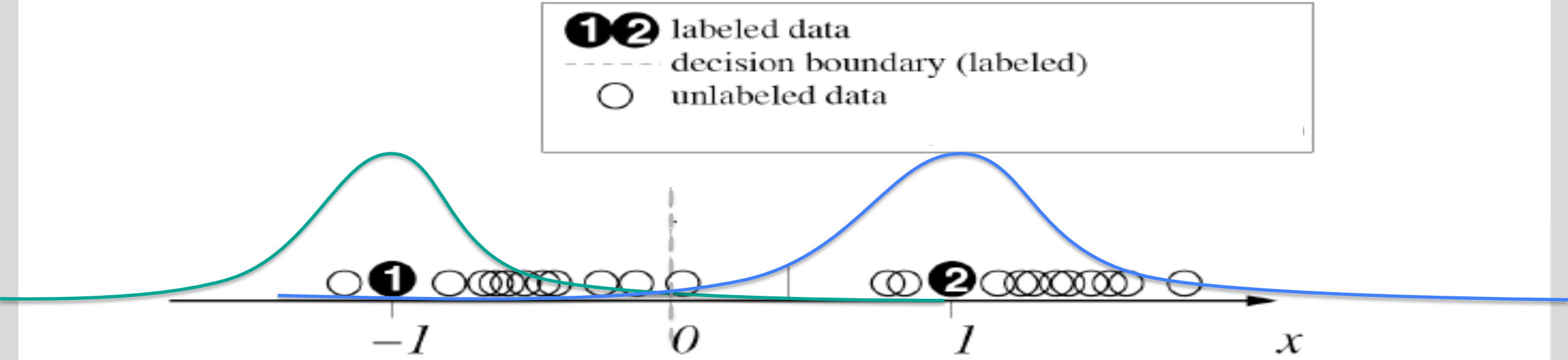


Generative Modelle / Methoden

- Generative Algorithmen nutzen eine Schätzung der Verteilung der Daten für die Klassen

➔ Zusätzliche Information der Verteilung der Daten sind sinnvoll !!

- Ausgehend davon, dass Klassen kohärente Daten enthalten z.B. normalverteilt ➔ Geschätzte Entscheidungsgrenze wandert je nach Verteilung der Daten

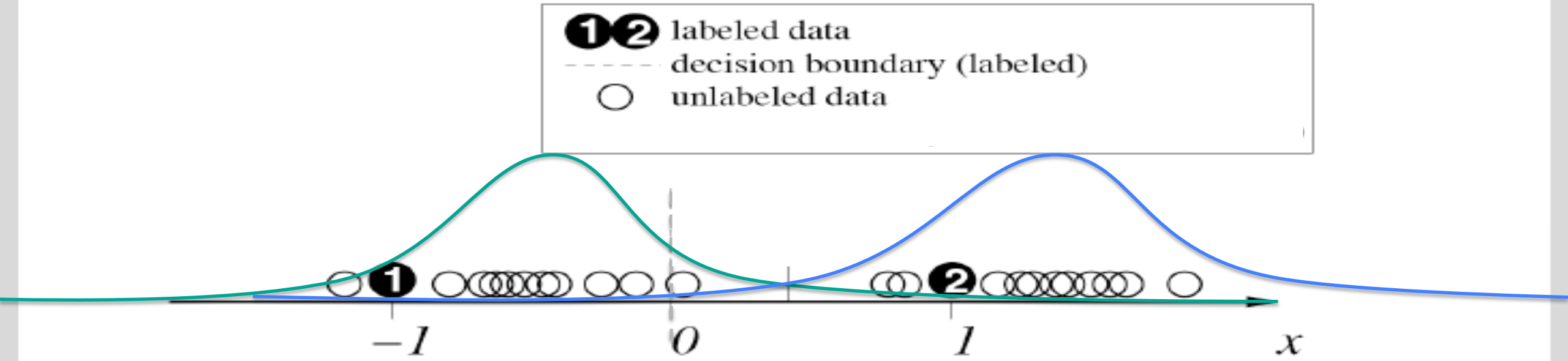


Generative Modelle / Methoden

- Generative Algorithmen nutzen eine Schätzung der Verteilung der Daten für die Klassen

➔ Zusätzliche Information der Verteilung der Daten sind sinnvoll !!

- Ausgehend davon, dass Klassen kohärente Daten enthalten z.B. normalverteilt ➔ Geschätzte Entscheidungsgrenze wandert je nach Verteilung der Daten

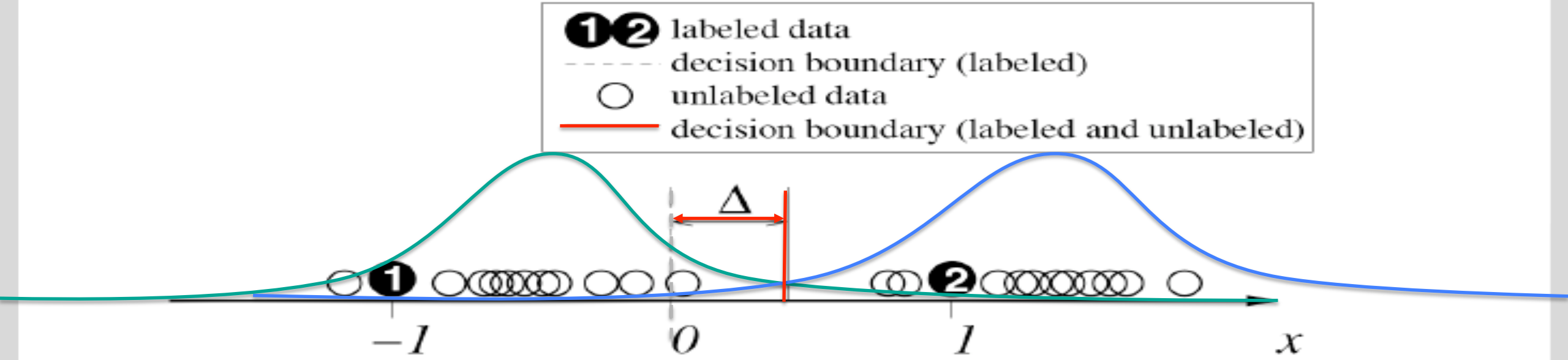


Generative Modelle / Methoden

- Generative Algorithmen nutzen eine Schätzung der Verteilung der Daten für die Klassen

➔ Zusätzliche Information der Verteilung der Daten sind sinnvoll !!

- Ausgehend davon, dass Klassen kohärente Daten enthalten z.B. normalverteilt ➔ Geschätzte Entscheidungsgrenze wandert je nach Verteilung der Daten

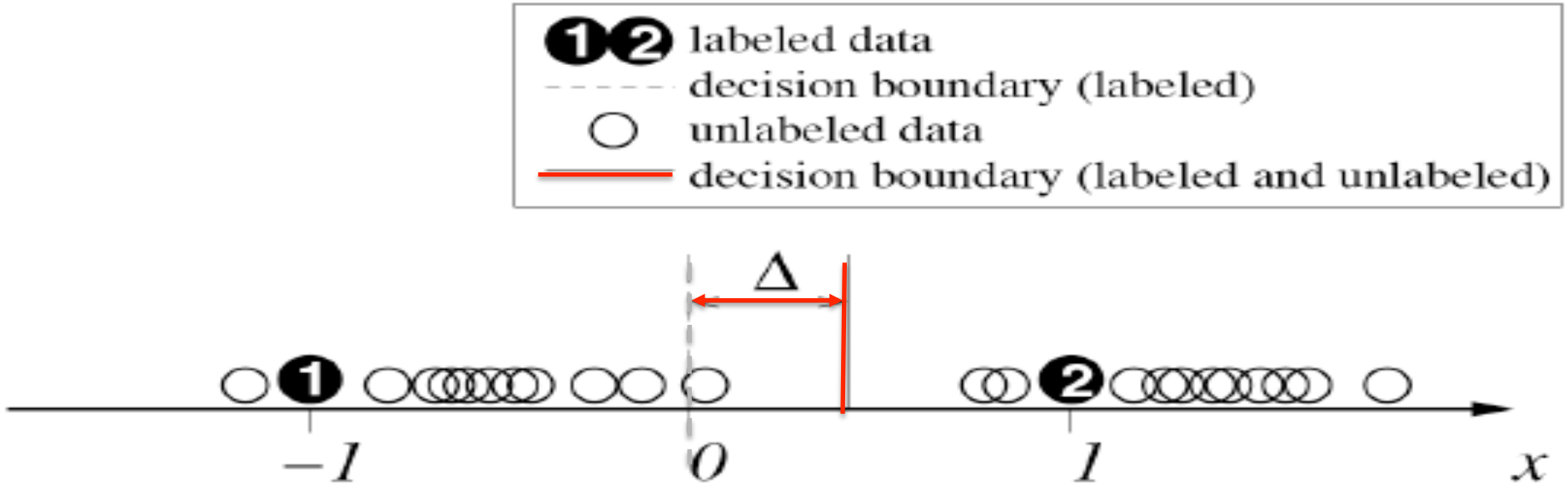


Generative Modelle / Methoden

- Generative Algorithmen nutzen eine Schätzung der Verteilung der Daten für die Klassen

➔ Zusätzliche Information der Verteilung der Daten sind sinnvoll !!

- Ausgehend davon, dass Klassen kohärente Daten enthalten z.B. normalverteilt ➔ Geschätzte Entscheidungsgrenze wandert je nach Verteilung der Daten



Generative Modelle / Methoden

- Zunächst (iteratives) Schätzen eines probabilistischen, parametrisierten Verbundmodells,

$$p(x, y|\theta)$$

  dann Entscheidung unter Verwendung des Modells
  - Modelle
    - Gaussche Mixturen (GMM)
    - Multinomiale Mixturen (Naive Bayes)
    - Hidden Markov Modelle (HMMs)
    - ...

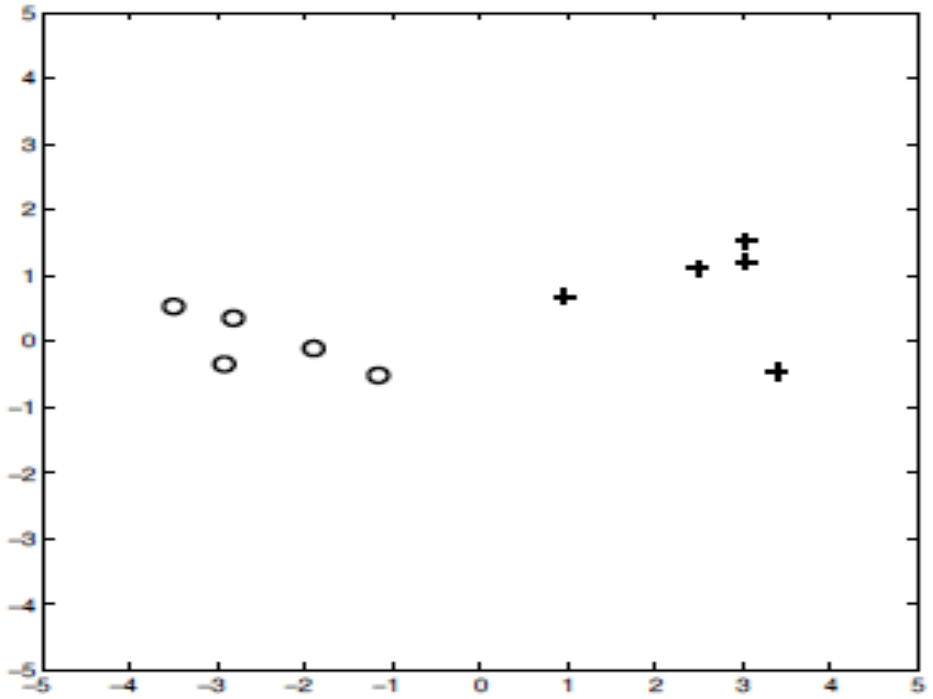
- Im Gegensatz dazu schätzen (nutzen) die diskriminativen Modelle direkt:

$$p(y|x)$$

  - z.B. SVM (support vector machines), CRF(conditional random fields) etc.

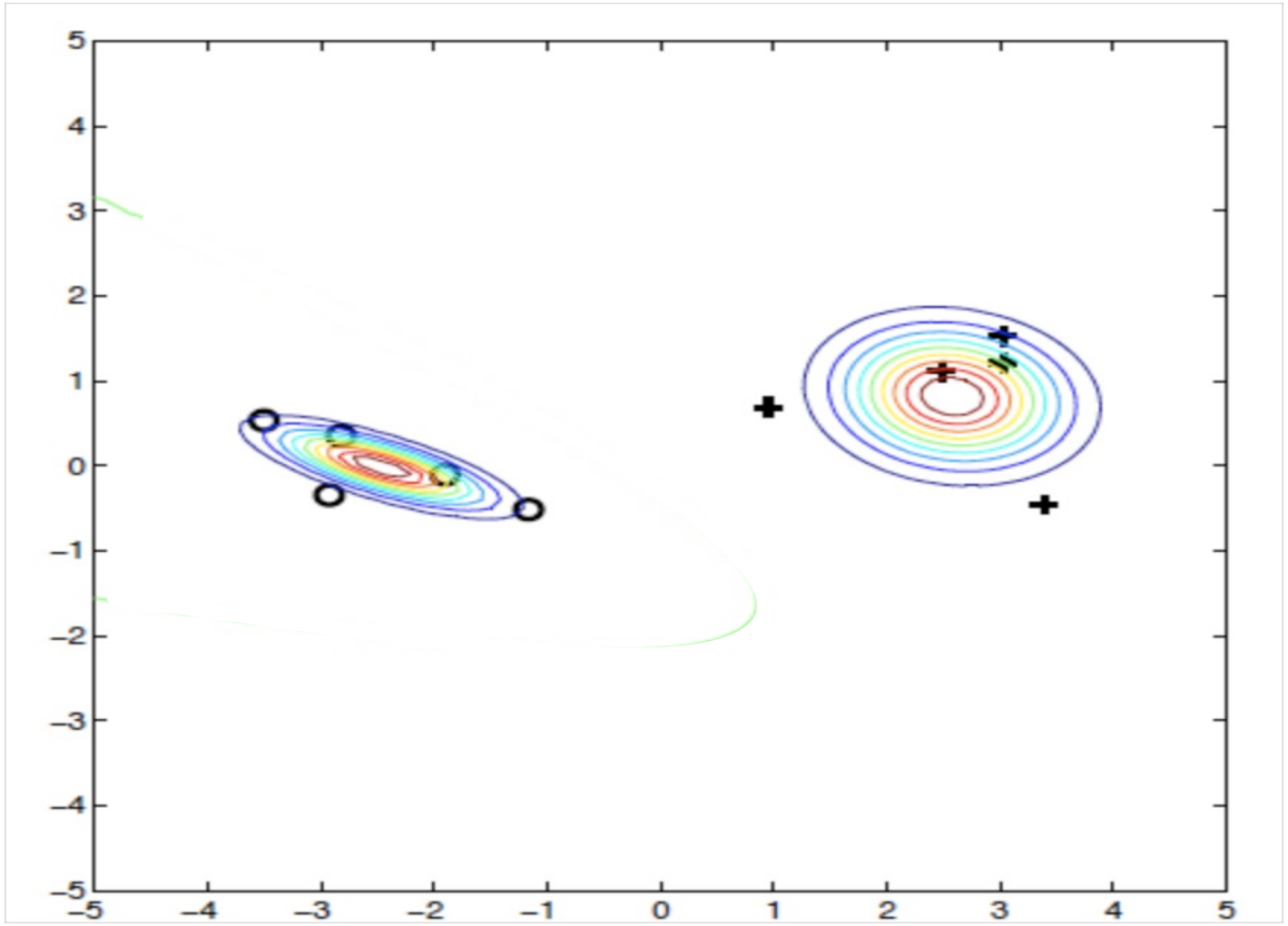
Einfaches Beispiel

- Gegeben folgende gelabelte Daten (X_l, Y_l)

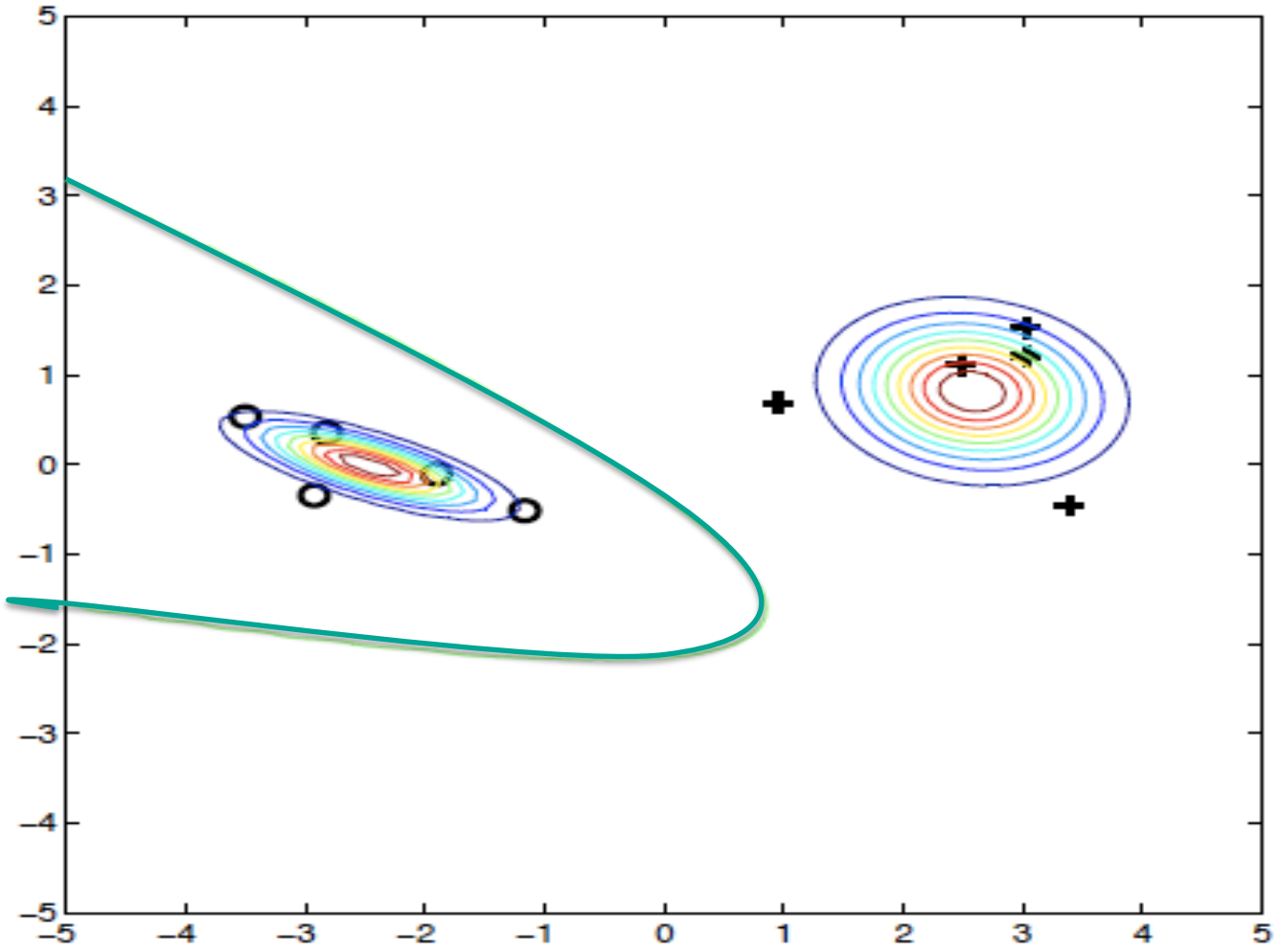


- Unter der Annahme, dass jede Klasse eine gaußsche Verteilung der Daten im Merkmalsraum hat, welches ist die wahrscheinlichste Trennung der Daten?

Einfaches Beispiel – Modell der Verteilung



Einfaches Beispiel - Trennung



Einfaches Beispiel - Trennung

- Gegeben die Modelparameter

$$\theta = \{\pi_1, \pi_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}, \qquad \pi_y = p(y)$$

- Gilt $$p(x|\theta) = \sum_{y=1}^{2} p(x|\theta_y)\pi_y = p(x|\mu_1, \Sigma_1)\pi_1 + p(x|\mu_2, \Sigma_2)\pi_2$$

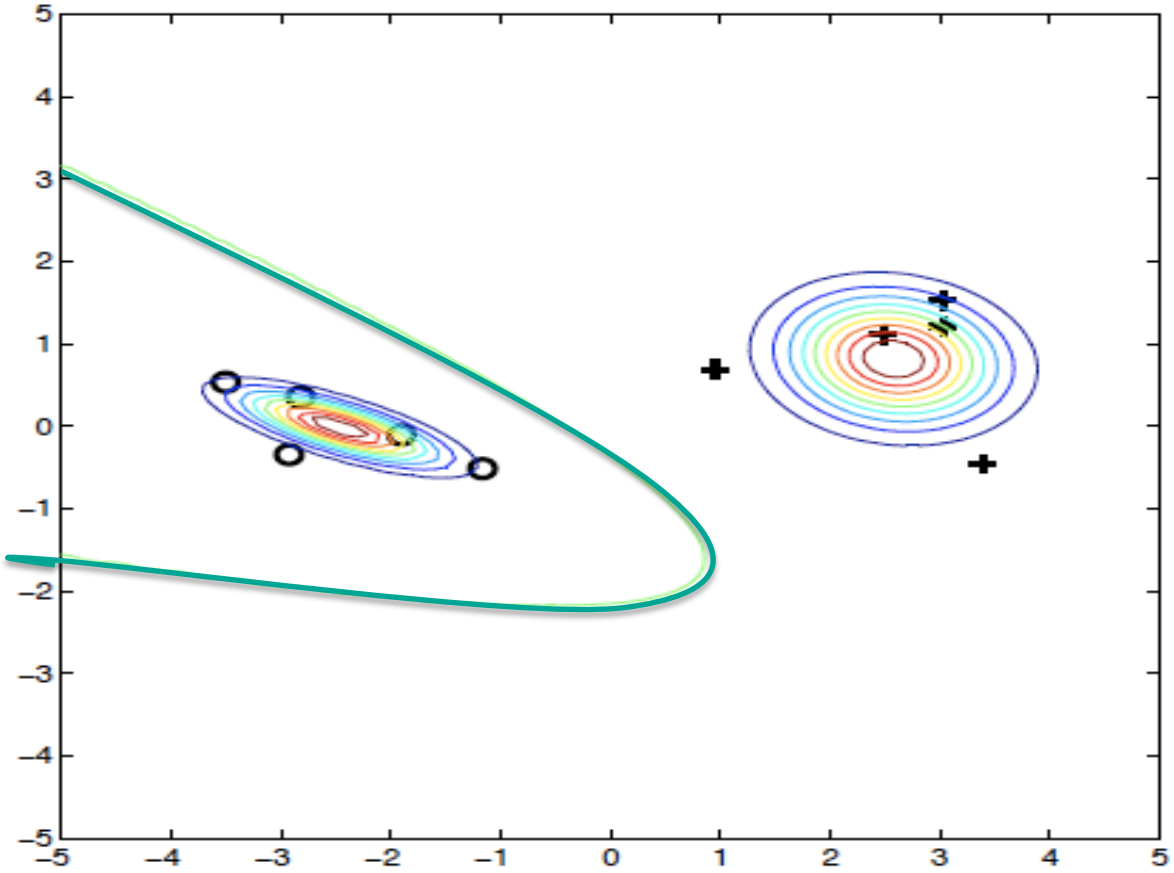
- Und die Klassenwahrscheinlichkeit

$$x \in X_u, \;\; p(y|x, \theta) = \frac{p(x|\theta_y)\pi_y}{p(x|\theta)}$$

es gilt (lt. Bayes)

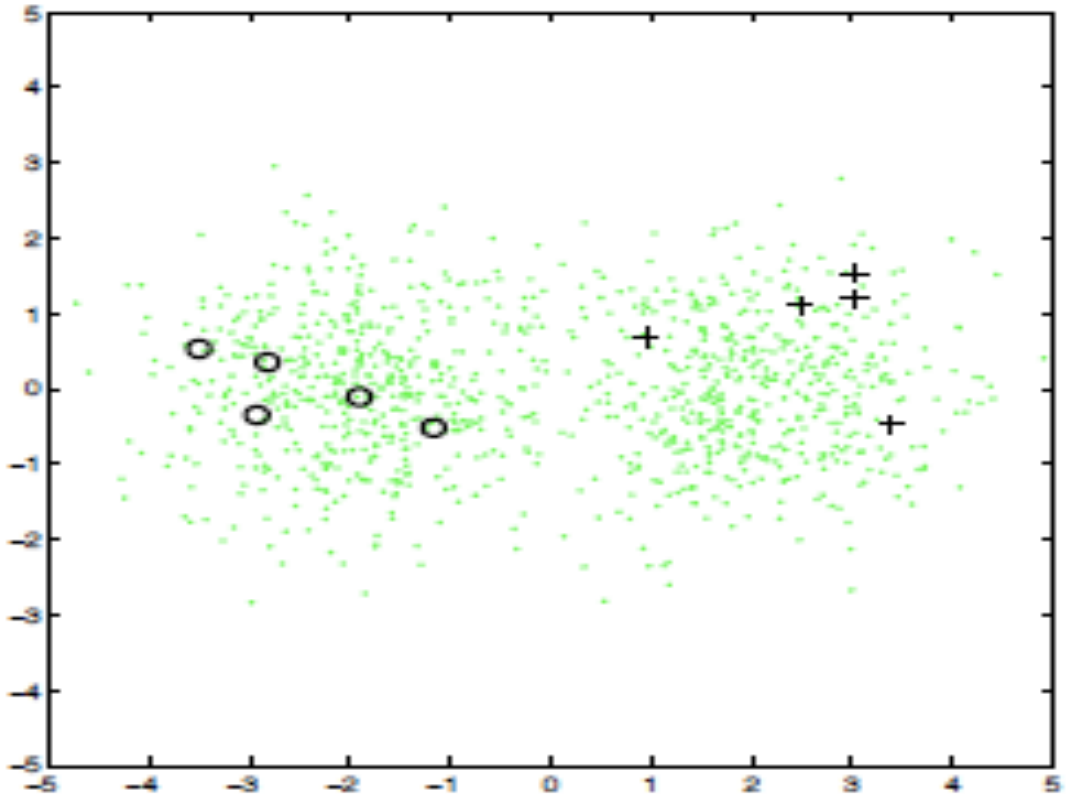
$$p(y|x) = \frac{p(x|y)p(y)}{\int_{y'} p(x|y')p(y')}$$

Einfaches Beispiel – Trennung (auf Basis gelabelter Daten)

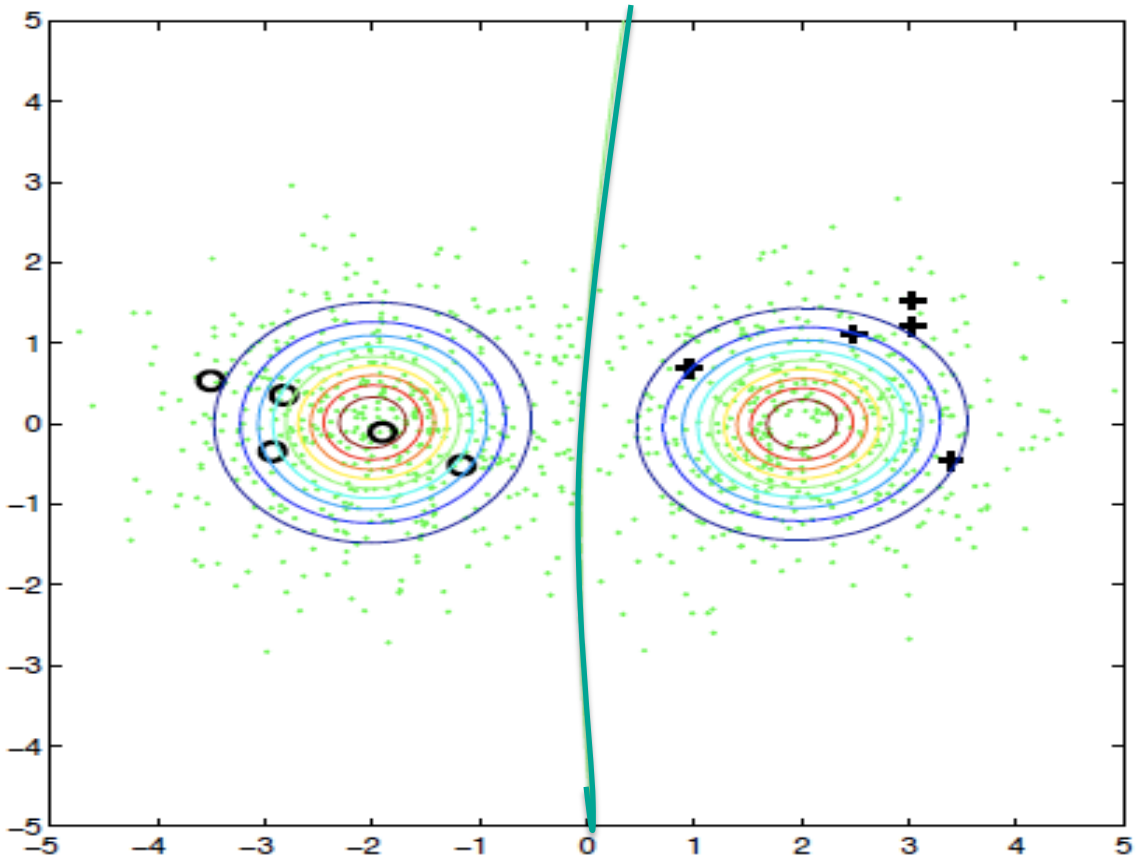


Einfaches Beispiel – Neue Daten

- Wenn zu den gelabelten Daten (X_l, Y_l) noch X_u hinzukommen?



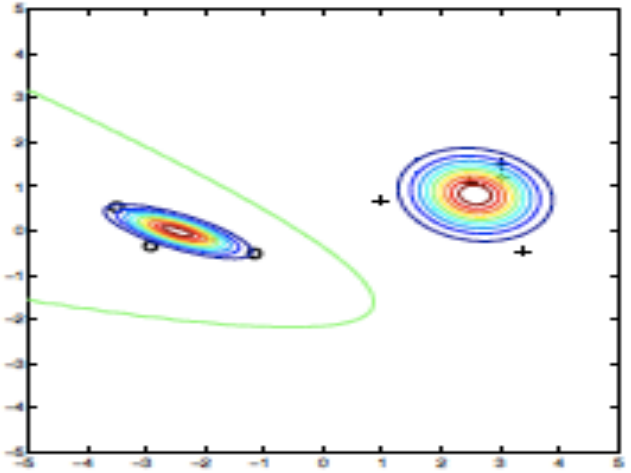
Einfaches Beispiel – Trennung durch das beste Modell



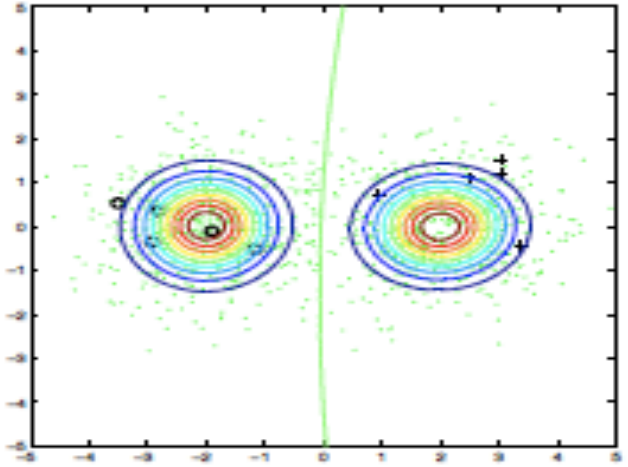
Einfaches Beispiel – Wieso?

- Weil hier GMM (gaussian mixture models) verwendet werden und
- Für die Bestimmung der Parameter θ ergeben sich unterschiedliche Optimierungsaufgaben (Maximierung):

$$p(X_l, Y_l|\theta)$$



$$p(X_l, Y_l, X_u|\theta) = \sum_{Y_u=1}^{2} p\left(X_l, Y_l, X_u, Y_u \middle| \theta\right)$$



Generative probabilistische Modelle – Grundverfahren

- 1. Wähle ein generatives Modell

$$p(x, y|\theta)$$

- 2. Finde Maximum likelihood Schätzung (MLE) auf gelabelte und ungelabelte Daten

$$\theta^* = \arg\max_{\theta} p(X_l, Y_l, X_u|\theta)$$

EM (expectation maximization)

- 3. Bestimme Klassenzugehörigkeit entsprechend der Bayes‘schen Regel

$$p(y|x, \theta^*) = \frac{p(x, y|\theta_y^*)}{\sum_{y'} p(x, y'|\theta^*)} = \frac{p(x|\theta_y^*)\, p(y|X, Y)}{p(x|\theta^*)}$$

EM – Algorithmus für GMM

- Starte mit MLE $\theta = \{\pi_1, \pi_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$ auf (X_l, Y_l)
- E- Schritt
  - Berechne Wahrscheinlichkeit für alle $x \in X_u,\ p(y|x,\theta) = \frac{\pi_y p(x|\theta_y)}{p(x|\theta)}$
  - Setzte Label (je nach Maximum) $y_u = 1 \quad bzw. \quad y_u = 2$
- M – Schritt
  - Update von θ jetzt auch mit (X_u, Y_u) (soft label)
  - A-priori-Klassenwahrscheinlichkeit = Anteil der Daten mit Klasse $i = \pi_i$
  - (gewichteter) Mittelwert der Daten der Klasse $i = \mu_i$
  - (gewichtete) Kovarianz der Daten der Klasse $i = \Sigma_i$
- Iteriere
- Kann als Sonderfall des Selbstlernens gesehen werden

EM – Verallgemeinerung

- Problem
  - Beobachtete Daten $\mathcal{D} = (X_l, Y_l, X_u)$
  - Versteckte Daten $\mathcal{H} = Y_u$

- Ziel: Finde θ so dass $p(\mathcal{D}|\theta) = \sum_{\mathcal{H}} p(\mathcal{D}, \mathcal{H}|\theta)$ maximiert wird

- Ablauf und Eigenschaften
  - EM startet mit θ_0
  - E-Schritt: Bestimme Belegung $q(H) = p(H|D, \theta)$
  - M – Schritt Maximiere $p(D|\theta) \geq \sum_H q(H) \ln p(D|H, \theta)$
    (untere Schranke lt Jensen, i.A. einfacher zu berechnen)
  - EM – verbessert somit iterativ θ
  - EM konvergiert zu einem lokalen Maximum $p(D|\theta)$ in Abh. von θ

Generative Modelle für SSL

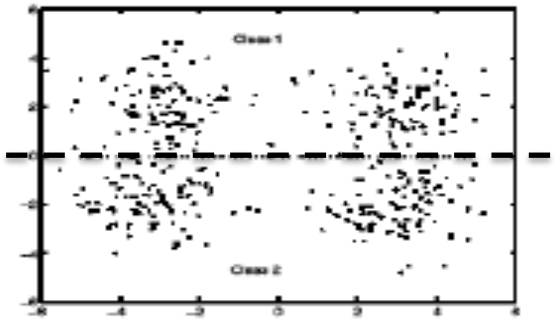
- Grundsatz
  - Maximierung von $p(X_l, Y_l, X_u|\theta)$
  - Verwende optimales Modell für die Trennung
  - EM ist nur eine Variante, andere Methoden existieren

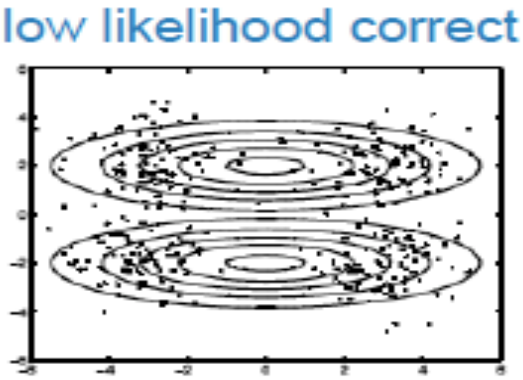
- Vorteile
  - Klares, wohl definiertes Framework
  - Kann sehr effektiv sein WENN das Modell korrekt ist

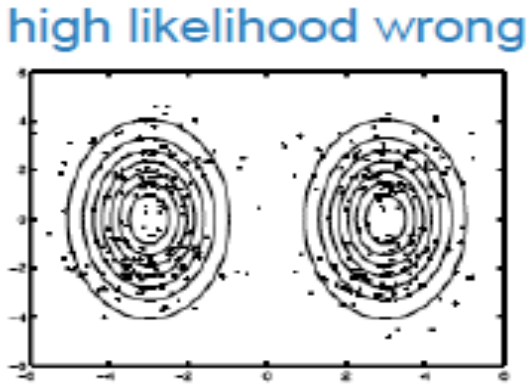
- Nachteile
  - Verifikation der Korrektheit des Modell meist nicht möglich
  - EM – kann zu lokalen Minima führen
  - Ungelabelte Daten können schaden wenn das Modell nicht korrekt ist

Generative Modelle für SSL

- Beispiel
  - Gegeben 2 Klassen mit der Annahme: gaußschen Verteilungen
  - Wenig gelabelte Daten
  - Viele ungelabelte Daten







  - Je mehr ungelabelte Daten umso schlechter ...
    (hoher likelihood der Modellbestimmung führt zu falscher Trennung)
  - Annahme dieser gaußschen Verteilung „killt EM“ aber nicht SSL!
    (z.B. durch Verwendung eines anderen Modells)

Generative Modelle für SSL – Erweiterungen

- Heuristiken
  - Generative Modelle erweitern um der Aufgabenstellung gerecht zu werden z.B. multiple Gausverteilungen pro Klasse (statt nur einer pro Klasse)
  - Ungelabelte Daten gewichten mit $\lambda < 1$

➔ Maximierung des „gewichteten“ Loglikelihood :

$$\ln p(X_l, Y_l, X_u \mid \theta) = \sum_{i=1}^{l} \ln p(x_i, y_i \mid \theta) + \lambda \sum_{i=l+1}^{u} \ln \left(\sum_{y=1}^{2} p(x_i, y \mid \theta) \right)$$

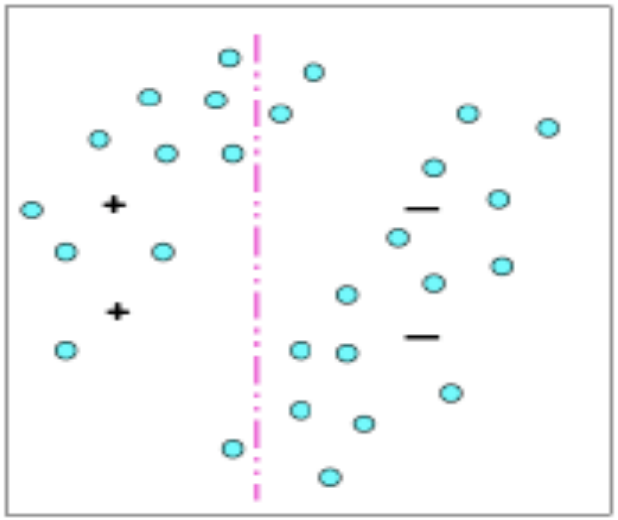
➔ Ungelabelte Daten fließen nur gewichtet in die Bestimmung des Modells ein

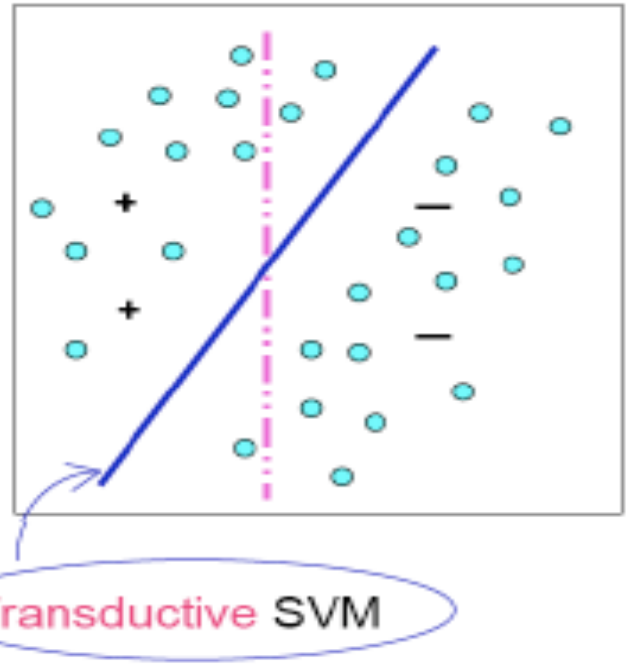
Verschieden Ansätze

- Erste Algorithmen
  - Self-Training & Co-Training
- Generative probabilistische Modelle (Generative Probabilistic Models)
  - EM for Gaussian Mixtures
- Dichte Trennung (Low-Density Separation)
  - „Transduktive“ SVM
- Graph basierte Modelle / Methoden
  - Methoden bei denen die Daten als Knoten eines Graphs repräsentiert sind und die Kanten die jeweiligen Abstände enthalten
- Änderung der Repräsentation
  - unüberwachtes Lernen (z.B.: Clustern) um neue (i.A. niedrig dimensionale) Repräsentationen der Daten zu erhalten
  - Lernen der Zuordnung der Cluster zu Klassen

Dichte Trennung (Low – density separation) mit SVM

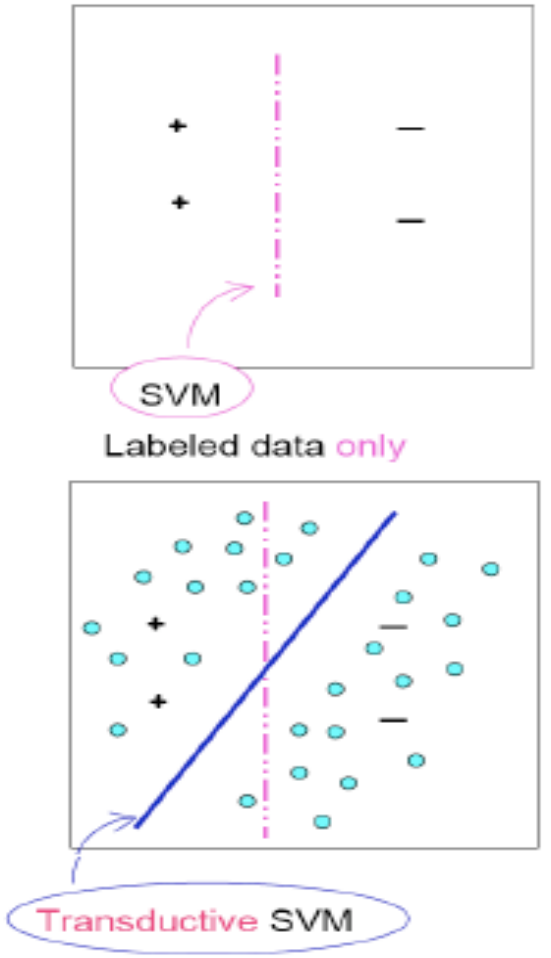
- Ziel





Transduktive SVM

- Annahme
  - Ungelabelte Daten unterschiedlicher Klassen werden mit großem Rand getrennt – aber wie?
- Naiver Ansatz
  - Alle 2^u Möglichkeiten der Labels v. $X_u = \{x_{l+1:l+u}\}$ betrachten
  - Trainiere SVM für alle Möglichkeiten
  - Wähle SVM mit größtem Rand
- Besser
  - Integriere ungelabelte Daten in das Optimierungsproblem



Literatur

Chapelle, Schöllkopf, Zien: „Semi – Supervised Learning“,

- MIT – Press, 2010

X. Zhu: „Semi-Supervised Learning Literature Survey“

- http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf

K-R Müller, A.Zien: „Semi-Supervised Learning“

- Folien: https://ml01.zrz.tu-berlin.de/wiki/Main/SS09_MaschinellesLernen2

B. Schiele: Vorlesung ML

- Folien: http://www.mis.tu-darmstadt.de/ml2