Zusammenfassung: Statistical Learning Theory & Linear Models

Kevin Pech

25.11.2025

0. Einführung / Erinnerung

Sei $\mathcal{H}$ eine Hypothesis Class, $h \in \mathcal{H}$, f eine Labeling Function, $\mathcal{D}$ eine Wahrscheinlichkeitsdichte.

- **Dataset:** $S = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \dots, m\}$
- **Generalization Error:**

$$R_{\mathcal{D}}(h) := \mathbb{E}_{x \sim \mathcal{D}}[l(h(x), f(x))] = \int l(h(x), f(x))\, d\mu(x)$$

- **Empirical Error:**

$$L(h) = \frac{1}{m} \sum_{i=1}^{m} l(h(x_i), y_i)$$

Für ein Klassifikationsproblem gilt: $l(y, y') = \mathbb{1}_{\{y \neq y'\}}$.

1. Statistical Learning Theory

1.1 Learnability

Betrachte binäre Klassifizierung:

- $\mathcal{Y} = \{0, 1\}$, $S = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \dots, m\}$ mit x_i u.i.v. nach $\mathcal{D}$ über $\mathcal{X}$.
- $y_i = f(x_i)$ für zunächst beliebige Labeling Function f.
- Loss function: $\mathbb{1}_{\{h(x) \neq y\}}$

S ist random gesampled, also ist auch der trainierte Predictor h zufallsbehaftet (da h mit S trainiert wurde). $\Rightarrow$ Betrachte $R_{\mathcal{D}}(h)$ als Zufallsvariable. S soll $\mathcal{D}$ gut repräsentieren.

Bsp: 30% rote, 70% grüne Kugeln in einer Urne. Wahrscheinlichkeit für das Ziehen von "GGGGG" $= (0, 7)^5 \approx 0, 16 \gg 0$. $\Rightarrow$ Problem mit kleinen Datensätzen.

Untersuche nun Lernbarkeit von $\mathcal{H}$ aus einem endlichen Datensatz.

1.1.1 Realisierbarkeitsannahme

Def: Gegeben $\mathcal{H}, \mathcal{D}, f$. Die Realisierbarkeitsannahme gilt für $\mathcal{H}, \mathcal{D}, f$, wenn:

$$\exists h \in \mathcal{H} : R_{\mathcal{D}}(h) = 0$$

Das bedeutet, dass f durch Elemente von $\mathcal{H}$ dargestellt werden kann. Insbesondere: $f \in \mathcal{H} \Rightarrow$ Real.ann. gilt.

Bsp: Sei $\mathcal{X} = \mathbb{R}$, $\mathcal{D} = \text{Unif}(-1, 1)$, $f : x \mapsto \mathbb{1}_{\{|x|>1/3\}}$, $\mathcal{H} = \{h_w : x \mapsto \mathbb{1}_{\{x>w\}}\}$.

$$R_{\mathcal{D}}(h_w) = \int_{-1}^{1} \mathbb{1}_{\{h_w(x) \neq f(x)\}} \, dx \neq 0 \Rightarrow \text{Real.ann. gilt nicht.}$$

1.1.2 PAC Learnability (Probably Approximately Correct)

Real.ann. bedeutet, es gibt einen Predictor, der perfekt zu S passt. Das bedeutet nicht, dass wir diesen finden $\rightarrow$ PAC Framework.

Sei A ein Algorithmus, der einen Predictor h_A berechnet, welcher L minimiert, also $h_A \in \underset{h \in \mathcal{H}}{\text{argmin}} \, L(h)$.

Def: ($\epsilon \hat{=}$ Accuracy, $\delta \hat{=}$ Confidence) $\mathcal{H}$ ist PAC learnable :$\Leftrightarrow$

$$\exists (m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}) : \forall \epsilon, \delta \in (0,1); \mathcal{D} \text{ über } \mathcal{X}; (f : \mathcal{X} \to \{0,1\}) :$$

$$\text{Real.ann. gilt für } \mathcal{H}, \mathcal{D}, f \Rightarrow R_{\mathcal{D}}(h_A) \leq \epsilon \text{ mit Wahrscheinlichkeit } 1 - \delta,$$

sofern A auf $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ u.i.v. nach $\mathcal{D}$ gesampleten und f-gelabelten Datenpunkten gelaufen ist.

Bsp: Sei $\mathcal{H} = \text{Abb}(\mathbb{R}, \{0,1\})$. Intuitiv sehr mächtig. Zu Beginn: $m_{\mathcal{H}}(\epsilon, \delta) = m$ fixiert. Konstruiere nun $\mathcal{D}$ als diskrete uniforme Dichte über $2m$ Punkte.

1. Ziehe m u.i.v. Samples $\Rightarrow$ erwartet werden $\approx m$ verschiedene Punkte. $\Rightarrow \approx$ Hälfte des Supports von $\mathcal{D}$ wird nicht gesehen.
2. Offenbar ist $f \in \mathcal{H}$.
3. A findet h_A mit $L(h_A) = 0$.
4. Keine Information über ungesamplete Punkte gelernt $\Rightarrow$ keine Generalisierung.
5. Setze f so, dass $f(x) = 1 - h_A(x)$ auf ungesampleten Punkten ist.

$\Rightarrow R_{\mathcal{D}}(h_A) \approx \frac{1}{2} \Rightarrow$ Für $\epsilon \lesssim \frac{1}{2}$ ist $\mathcal{H}$ nicht PAC learnable.

1.1.3 Agnostic PAC Learnability

Real.ann. i.A. nicht gegeben, da Noise im Datensatz enthalten ist $\Rightarrow$ keine Garantie für $R_{\mathcal{D}}(h) = 0$. Idee: Korrekturterm $\min_{h^* \in \mathcal{H}} R_{\mathcal{D}}(h^*)$.

Def: $\mathcal{H}$ agnostic PAC learnable :$\Leftrightarrow$

$$\exists (m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}) : \forall \cdots : \cdots \Rightarrow R_{\mathcal{D}}(h_A) \leq \epsilon + \min_{h^* \in \mathcal{H}} R_{\mathcal{D}}(h^*) \text{ mit W-keit } 1 - \delta, \ldots$$

1.2 Finite sized hypothesis classes

PAC Framework → Ist $\mathcal{H}$ zu komplex, finden wir keine guten h. Sei im Folgenden $\mathcal{H}$ endlich.

1.2.1 PAC Learnability

Jede endliche Hypothesis Class ist PAC Learnable.

Satz: Es gelte Real.ann., $\mathcal{H}$ endlich $\Rightarrow$ $\mathcal{H}$ PAC learnable. Zudem kann $m(\epsilon, \delta) = \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil$ gewählt werden.

Beweis: Sei h_A der unter A trainierte Predictor. Wähle $\epsilon, \delta \in (0,1)$ und sei $m(\cdot,\cdot)$ wie oben. Es genügt zu zeigen, dass $\mathcal{D}^m(\{S : R_\mathcal{D}(h_A) > \epsilon\}) < \delta$.

Sei $\mathcal{H}_b := \{h \in \mathcal{H} \mid R_\mathcal{D}(h) > \epsilon\}$, $M_m := \{S \in (\mathcal{X} \times \mathcal{Y})^m \mid \exists h \in \mathcal{H}_b : L(h) = 0\}$. $M_m(h) := \{S \in (\mathcal{X} \times \mathcal{Y})^m \mid L(h) = 0\}$. $\Rightarrow M_m = \bigcup_{h \in \mathcal{H}_b} M_m(h)$.

$$\Rightarrow \mathcal{D}^m(M_m) \leq \sum_{h \in \mathcal{H}_b} \mathcal{D}^m(M_m(h))$$

Da u.i.v.: $\forall h \in \mathcal{H}_b : \mathcal{D}^m(M_m(h)) = \prod_{i=1}^m \mathcal{D}(x_i : h(x_i) = y_i)$.
Zudem ist $R_\mathcal{D}(h) = \mathbb{E}_{x \sim \mathcal{D}}(\mathbb{1}_{\{h(x) \neq y\}}) = \mathcal{D}(x_i : h(x_i) \neq y_i)$.

$$\Rightarrow \mathcal{D}(x_i : h(x_i) = y_i) = 1 - R_\mathcal{D}(h) \leq 1 - \epsilon$$

$$\Rightarrow \mathcal{D}^m(M_m(h)) \leq (1-\epsilon)^m \leq e^{-\epsilon m} \quad (\text{denn } 1 + x \leq e^x \; \forall x \in \mathbb{R})$$

$$\Rightarrow \mathcal{D}^m(M_m) \leq \sum_{h \in \mathcal{H}_b} e^{-\epsilon m} \leq |\mathcal{H}| e^{-\epsilon m}$$

Man setze dies $< \delta$ und löse nach m auf. □

1.2.2 Agnostic PAC Learnability

Anpassung analog zum nicht-endlichen Fall.
$\mathcal{H}$ endl. $\Rightarrow$ $\mathcal{H}$ Agnostic PAC Learnable mit

$$m_\mathcal{H}(\epsilon, \delta) := \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil$$

2. Linear Models

Die in diesem Kapitel vorkommenden Grafiken, sind allesamt unverändert aus [1] übernommen worden.

Sei $\mathcal{X} = \mathbb{R}^n, \mathcal{Y} = \mathbb{R}$, $\mathcal{H} = \{f_w : x \mapsto w^T x + b \mid w \in \mathbb{R}^n, b \in \mathbb{R}\}$.

2.1 Lineare Regression

Def (homogene Darstellung): $w = (w_1, \ldots, w_n, b)$, $x = (x_1, \ldots, x_n, 1)$.

Def (Cost function): $C(w) := \frac{1}{m} \sum_{i=1}^m l(f_w(x_i), y_i) \rightarrow$ analog zum Empirical Error.

Erinnerung: l konvex im 1. Argument $\Rightarrow$ Training auf $\mathcal{H}$ mit Gradientenabstieg auf den Parametern konvergiert garantiert zu einem globalen Minimum der Cost function.

Häufige Loss funcs:

- L_2 loss: $l(y, y') = \frac{1}{2}(y - y')^2$
- L_1 loss: $l(y, y') = |y - y'|$
- Huber loss: $l(y, y', \delta) = \begin{cases} \frac{1}{2}(y - y')^2 & , |y - y'| < \delta \\ \delta(|y - y'| - \frac{1}{2}\delta) & , \text{sonst} \end{cases}$

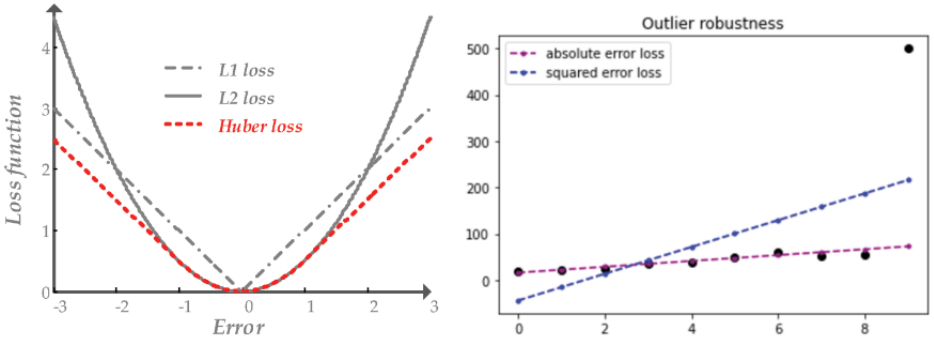

Figure 5.1: *Left*: Different shapes of loss functions. *Right*: Example of an optimal linear predictor for the squared and absolute error losses with the presence of outlier.

Explizite Lösung (Least Squares):

$$\text{argmin}_w C(w) = \text{argmin}_w L(f_w) = \text{argmin}_w \frac{1}{2m} \sum_{i=1}^{m} (w^T x_i - y_i)^2 = \frac{1}{2m} \text{argmin}_w \|X^T w - Y\|^2 \quad (*)$$

Def: (Moore-Penrose Pseudoinverse) Sei $A \in \mathbb{R}^{m \times n}$, $A^+ \in \mathbb{R}^{n \times m}$, dann ist A^+ MPPI von $A :\Leftrightarrow$

1. $AA^+A = A$
2. $A^+AA^+ = A^+$
3. $(AA^+)^T = AA^+$
4. $(A^+A)^T = A^+A$

$\rightarrow$ eindeutige MPPI durch Singulärwertzerlegung.

Satz: Sei $X' \in \mathbb{R}^{m \times (n+1)}$, $Y \in \mathbb{R}^m$, X^+ MPPI von X'. Dann:
w löst $(*) \Leftrightarrow \exists z \in \mathbb{R}^{n+1} : w = X^+Y + (I_{n+1} - X^+X)z$.

Korollar:

1. $n + 1 = m \Rightarrow X^+ = X^{-1}$
2. $m > n + 1 \Rightarrow X^TX$ invertierbar $\rightarrow$ eindeutiges $w = (X^TX)^{-1}X^TY$
3. $m < n + 1 \Rightarrow X^TX$ nicht invertierbar $\rightarrow$ unendlich viele Lösungen w.
   $\rightarrow w^* = \text{argmin}_w \|w\|^2$ s.t. $\min \|X^Tw - Y\|^2 \Rightarrow w^* = X^+Y$.

Regularisierung: Sei $R : \mathbb{R}^{n+1} \to \mathbb{R}_+$. Betrachte $\text{argmin}_w(L(w) + R(w))$. ($\to$ gegen Overfitting / hochgradiges Polynom).

Häufig: $\min_w \frac{1}{2m} \sum (y_i - w^T x_i)^2 + \begin{cases} \lambda \|w\|_2^2 & \text{Ridge regression} \\ \lambda \|w\|_1 & \text{Lasso regression} \end{cases}, \quad \lambda > 0.$

Bsp: $w^* = (X^T X + \lambda m I_m)^{-1} X^T Y$ löst Ridge Reg.

Nichtlineare Transformationen: Was wenn Relation zwischen input und output space nicht linear ist? $\to f_w(x) = w_2 x^2 + w_1 x + b$. $w = (b, w_1, w_2)$, $X = (1, x, x^2)$ features. Allgemeiner: $f_w(x) = \sum_{i=0}^{n} w_i \phi_i(x)$, $\phi_i(x) \hat{=}$ nichtlineare Transformation.

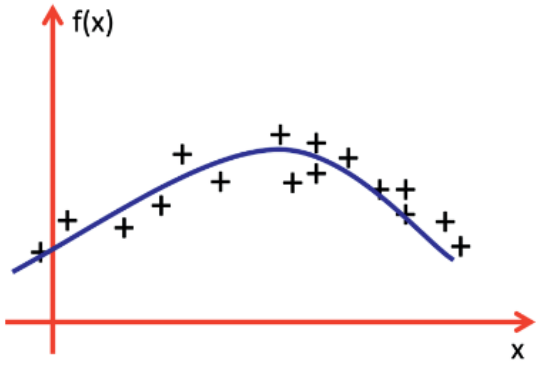

Figure 5.2: Quadratic relation.

2.2 (Binäre) Klassifizierung

Sei $S = \{(x_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} \mid i = 1, \ldots, m\}$.
Annahme: Daten linear separierbar von $P = \{x \in \mathbb{R}^n \mid w^T x + b = 0\}$ für festes w, b. D.h. $\exists P : \forall i = 1, \ldots, m : (y_i = 1 \Leftrightarrow w^T x_i + b > 0)$ und $(y_i = -1 \Leftrightarrow w^T x_i + b < 0)$.

$\Rightarrow \mathcal{H} = \{f_w : x \mapsto \text{sign}(w^T x + b) \mid w \in \mathbb{R}^n, b \in \mathbb{R}\}$. Wie findet man ein gutes w?

Betrachte $\text{argmin}_w C(w) = \text{argmin}_{w,b} \frac{1}{m} \sum \mathbb{1}_{\{y_i \neq \text{sign}(w^T x_i + b)\}}$. $\to$ nicht konvex oder stetig. Besser: $l_{perc}(y, y') = \max(0, -yy')$. $\Rightarrow C(w) = \frac{1}{m} \sum \max(0, -y_i(w^T x_i + b))$.

Perceptron Algorithmus: Gradient: $\nabla C(w) = \frac{1}{m} \sum \begin{cases} 0 & \text{korrekt klassifiziert} \\ -y_i x_i & \text{falsch klassifiziert} \end{cases}$

Gradient descent: $w_{t+1} = w_t - \eta_t \frac{1}{m} \sum_{i: y_i(w^T x_i + b) < 0} -y_i x_i$.

Support Vector Machine (SVM)

Idee: Hyperebene P mit maximalem Abstand zu den nächsten Datenpunkten finden.

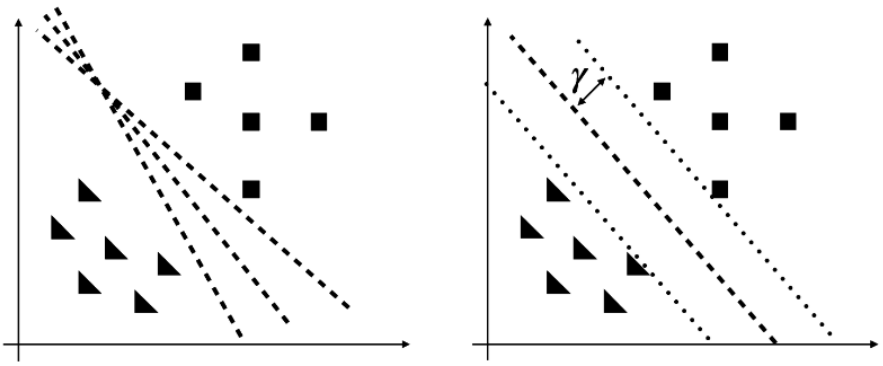

Figure 5.3: Two different separating hyperplanes for the same data set. Right: The maximum margin hyperplane. The margin γ is the distance from the hyperplane (solid line) to the closest points in either class (which touch the parallel dotted lines).

Def (Margin):
Das SVM objective lautet $\max_{w,b} \gamma(w,b)$ s.t. $\forall i = 1,\dots,m : y_i(w^T x_i + b) > 0$.

Lemma: $\gamma(w,b) = \min_{x \in S} \frac{|w^T x + b|}{\|w\|}$.
Beweis: $w^T(x-d)+b = 0 \Leftrightarrow w^T(x-\alpha w)+b = 0 \Rightarrow \alpha = \frac{w^T x + b}{\|w\|^2}$.
$\|d\| = \sqrt{\alpha^2 w^T w} = \frac{|w^T x + b|}{\|w\|}$. □

Margin ist symmetrisch und $\gamma(\beta w, \beta b) = \gamma(w,b) \quad \forall \beta \neq 0 \in \mathbb{R}$.

Satz: Die Lösung vom SVM objective ist gleich der Lösung (w_*, b_*) von:

$$\min_w w^T w \quad \text{s.t. } \forall i = 1,\dots,m : y_i(w^T x_i + b) \geq 1$$

Beweis: Betrachte $\max_{w,b} \left[\min_{x \in S} \frac{|w^T x + b|}{\|w\|}\right]$ s.t. $\forall i = 1 \dots m, y_i(w^T x_i + b) > 0$. Skaliere w, b so, dass $\min_{x \in S} |w^T x + b| = 1 \Rightarrow \max_{w,b}[\dots] = \max_{w,b} \frac{1}{\|w\|} \cdot 1 = \min_w \|w\|$.

Neues Problem: $\min_w w^T w$ s.t. $\forall i : y_i(w^T x_i + b) > 0; \min_i |w^T x_i + b| = 1$. Äquivalent zu: $\min_w w^T w$ s.t. $\forall i : y_i(w^T x_i + b) \geq 1$. □

Lösung mittels Lagrange Multiplier:

$$\mathcal{L}(w,b,\beta) = w^T w - \sum_{i=1}^{m} \beta_i (y_i(w^T x_i + b) - 1), \quad \beta \in [0,\infty)^m$$

Durch lösen von $\min_{w,b} \max_\beta \mathcal{L}(w,b,\beta)$ erhält man (w_*, b_*, β_*).

$\rightarrow$ Lösen unter Berücksichtigung der Karush-Kuhn-Tucker (KKT) Bedingungen / des komplementären Schlupf. z.B.: $\nabla_w \mathcal{L} = 2w_* - \sum \beta_{*,i} y_i x_i \stackrel{!}{=} 0 \Rightarrow w_* = \frac{1}{2} \sum_{i=1}^m \beta_{*,i} y_i x_i$.
$\forall i = 1,\dots,m : \beta_{*,i}(y_i(w^T x_i + b) - 1) \stackrel{!}{=} 0 \Rightarrow \beta_{*,i} = 0$ oder $y_i(w^T x_i + b) = 1$.

Bemerkung: Support vector nicht eindeutig, z.B. verschiebe man die nicht supportenden Datenpunkte im Rahmen des Optimierungsproblems.

Der optimale SVM classifier f_* gegeben durch:

$$f_*(x) = \text{sign}(w_*^T x + b_*) = \text{sign}\left(\frac{1}{2}(\sum \beta_{*,i} y_i x_i)^T x + b_*\right)$$

Nicht-separierbarer Fall (Soft Margin):

$$\min_{w,b,\xi} w^T w + \lambda \sum_{i=1}^{m} \xi_i \quad \text{s.t. } \forall i = 1, \ldots, m : y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \lambda \geq 0, \xi_i \geq 0$$

- λ groß: SVM strict
- λ klein: 'opfere' Punkte für simplere P.

$$\min_{w,b} w^T w + \lambda \sum_{i=1}^{m} \max(1 - y_i(w^T x_i + b), 0)$$

Literatur

[1] Maria Han Veiga und François Gaston Ged. *The Mathematics of Machine Learning.* DE GRUYTER, 2024.