Statistical Learning

M. Gruber

KW 41, Rev.1

In Anlehnung an [1], Lec 3, skizzieren wir den Rahmen, in dem statistisches Lernen stattfindet.

1. Man hat eine *Lernmenge* (üblich ist auch die Bezeichnung *Trainingsdaten*) von der Form

$$D = \left\{ \begin{bmatrix} x^{(i)} \\ y^{(i)} \end{bmatrix} \middle| \, x^{(i)} \in \mathbf{R}^d, \, y^{(i)} \in \mathbf{R}, \, i = 1, \ldots, N \right\}.$$

   Man nimmt an, dass es Wertebereiche $\mathcal{X}$ (*input-Menge*) und $\mathcal{Y}$ (*output-Menge*) gibt, sodass sowohl die Trainingsdaten als auch alle weiteren Daten, die darüber hinaus betrachtet werden, in $\mathcal{X} \times \mathcal{Y}$ liegen.

2. Man nimmt an, dass es eine *Zielfunktion (target function)* f gibt, die den Zusammenhang zwischen x-Werten und y-Werten auf der Lernmenge hergestellt hat und für Daten außerhalb der Lernmenge herstellen wird. In den meisten Fällen geht man davon aus, dass die Zielfunktion "verrauscht" ist, d.h. dass man nicht das Ergebnis von f in den Daten wiederfindet, sondern das Ergebnis von $f +$ "Rauschen".

3. Man wählt ein problemadäquates *Lernmodell*, bestehend aus einer *Hypothesenmenge* und einem *Lernalgorithmus*. Die Hypothesenmenge besteht aus Kandidatenfunktionen für f. Es gibt ein Maß $\mathrm{Err}_{\mathrm{in}}(h)$ für den Fehler, den man auf der Trainingsmenge macht, wenn man die Hypothese h als Approximation für f wählt, den sogenannten *in-sample error*. Der Lernalgorithmus minimiert den *in-sample error* und gibt die Siegerfunktion als *finale Hypothese* g aus.

4. Die finale Hypothese g wird auf Daten außerhalb der Lernmenge angewandt. Man möchte, dass der Fehler, den man dabei machen wird (der *out-of-sample error* $\mathrm{Err}_{\mathrm{out}}(g)$), nicht wesentlich größer ist als der Fehler, den man auf der Lernmenge klein gehalten hat (den *in-sample error*). Die Differenz zwischen *out-of-sample error* und *in-sample error* nennt man *Generalisierungsfehler*. Die *Generalisierungstheorie* beschäftigt sich mit der Frage, wann und wie der Generalisierungsfehler klein gehalten werden kann.

5. Statistisches Lernen unterliegt dem Zufall. Es gibt eine Wahrscheinlichkeit $\mathbf{P}$, die beim Zustandekommen der Lernmenge gewirkt hat und die beim Zustandekommen der Daten außer-

halb der Lernmenge wirkt. Man begegnet der Wahrscheinlichkeit $\mathbf{P}$ wieder bei der Abschätzung des Generalisierungsfehlers. Man rechnet mit $\mathbf{P}$, man trifft aber keine Verteilungsannahmen über $\mathbf{P}$.

Beispiel 1 (Neuronales Netz) $\mathcal{X} = \{1\} \times \mathbf{R}^d$, $\mathcal{Y} = \mathbf{R}$. *Die Hypothesenmenge besteht aus neuronalen Netzen $x \mapsto F(x, w)$ mit einer bestimmten Anzahl $L - 1$ verborgener Schichten, bestimmten Anzahlen $d^{(l)}$ von Neuronen in der l-ten Schicht und einer bestimmten Aktivierungsfunktion (z.B.* $\tanh$*). Der Vektor w repräsentiert alle linearen Operationen innerhalb $F(x, w)$. Der Hypothesenraum ist die Menge aller Kandidaten-Vektoren w. Der in-sample error ist* $\mathrm{Err}_{\mathrm{in}}(w) = \frac{1}{N} \sum_{1 \leq i \leq N} (F(x^{(i)}, w) - y^{(i)})^2$. *Der out-of-sample error ist* $\mathrm{Err}_{\mathrm{out}}(w) = \mathbf{E}(F(X, w) - Y)^2$ *mit Zufallsvariablen X und Y.*

Beispiel 2 (Lineare Regression) $\mathcal{X} = \{1\} \times \mathbf{R}^d$, $\mathcal{Y} = \mathbf{R}$. *Hypothesenraum des Lernproblems ist der Raum $\mathbf{R}^{d+1}$ der Hypothesen w, die als Abbildungen $x \mapsto x^T w$ aufgefasst werden. Jedes w repräsentiert eine Abbildung von $\mathbf{R}^d$ nach $\mathbf{R}$ der Form*

$$\begin{bmatrix} x_1 \dots x_d \end{bmatrix}^T \mapsto w_1 x_1 + \dots + w_d x_d + w_0.$$

Von dieen Abbildungen ist jene gesucht, die von den Punkten der Trainingsmenge am besten erfüllt wird. Siegerhypothese wird also jenes w sein, das alle Gleichungen

$$x^{(i)^T} w = y^{(i)} \tag{1}$$

mit dem kleinsten Gesamtfehler löst, d.h. das den in-sample error

$$\mathrm{Err}_{\mathrm{in}}(w) = \frac{1}{N} \sum_{1 \leq i \leq N} (x^{(i)^T} w - y^{(i)})^2$$

minimiert.

Die Lineare Algebra liefert die perfekte Lösung[1]*. Die Elemente $[x^{(i)}, y^{(i)}]$ der Trainingsmenge fassen wir zur Datenmatrix $M = \left[x^{(i)^T}\right]_{1 \leq i \leq N}$ und zum Labelvektor $y = \left[y^{(i)}\right]_{1 \leq i \leq N}$ zusammen. Die Gleichung*

$$Mw = y \tag{2}$$

beinhaltet alle N Gleichungen (1). Im allgemeinen ist (2) nicht lösbar, aber man kann immer ein w^* angeben, das die Größe

$$||Mw - y||^2$$

minimiert, nämlich

$$w^* = (M^T M)^{-1} M^T y.$$

Damit ist auch $\mathrm{Err}_{\mathrm{in}}(w^*) = \min_w \mathrm{Err}_{\mathrm{in}}(w)$, *denn* $\mathrm{Err}_{\mathrm{in}}(w) = \frac{1}{N} ||Mw - y||^2$.

[1] Ich empfehle die Videovorlesung Projection matrices and least squares von Gilbert Strang.

Der Lernalgorithmus unseres Lernproblems ist die Berechnung der Pseudoinversen $M^+ = M^T M)^{-1} M^T$ *und deren Anwendung auf* y.

Die Siegerhypothese w^* *ist die Näherungslösung von* $Mw = y$ *nach der "Methode der kleinsten Quadrate". Sie ist der Normalenvektor der sogenannten "Ausgleichsebene" in* $\mathbf{R}^{d+1}$, *wobei hier* d *für die Komponenten* $x_1, \ldots, x_d$ *steht und* 1 *für die* y*-Komponente.

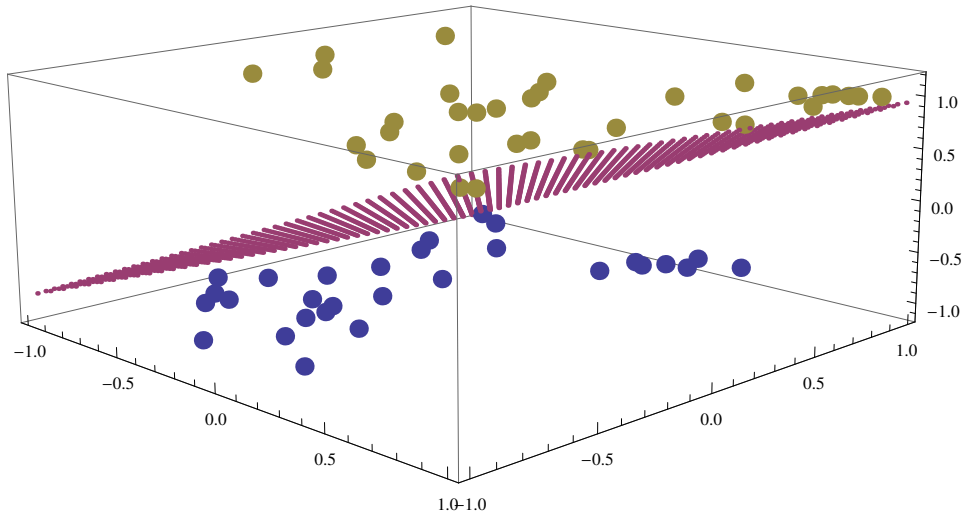

Abbildung 1: Punkte der (x_1, x_2)-Ebene sind mit y-Werten 1 (grün) bzw. -1 (blau) gelabelt. Die Ausgleichsebene ist rot.

Literatur

[1] Y. S. Abu-Mostafa. Learning from Data. Caltech CS 156.