Vorlesungsnotizen

Optimierung

Wintersemester 24/25

Emma Bach

Vorlesung gehalten von
Prof. Rolf Backofen
und
Prof. Moritz Diehl

March 11, 2025

Inhalt

Chapter 1

Einführung

Ein Optimierungsproblem besteht aus einer **zulässigen Menge** G und einer **Zielfunktion** $f : G \to H$, wobei wir im Allgemeinen von $H = \mathbb{R}$ ausgehen werden. Wir schreiben dann zum Beispiel

$$\min_{x \in \mathbb{R}} f(x),$$

um das Problem "finde den kleinsten Wert, den $f(x)$ bei reellem x annimt" zu notieren, und

$$\operatorname*{argmin}_{x \in \mathbb{R}} f(x) = 4$$

für das Problem "finde die kleinste reelle Zahl x, sodass $f(x) = 4$ ist.

Historisch ist das Feld der mathematischen Optimierung unter Nebenbedingungen stark verankert im Feld der **Operations Research**, welches sich mit der Optimierung von Produktionskosten unter gegebenen Bedingungen beschäftigt. Heutzutage hat die Optimierung zahlreiche Anwendungen, z.B. in der Pfadplanung, in Computer Vision, in der Bioinformatik, im maschinellen Lernen oder im Hardwaredesign.

1.1 Anwendungsbeispiel: Support Vector Machines

Ein klassisches Problem des maschinellen Lernes ist die Klassifizierung von Daten durch eine lineare Entscheidungsgrenze - alle Punkte (x_i, y_i) über einer Ebene werden einer Klasse zugeordnet, und alle Punkte unter der Ebene einer anderen Klasse. Das Problem besteht daraus, eine Optimale Trennungsebene zu finden. Diese wird beschrieben durch eine Gleichung der Form

$$\hat{y}(x) = w^T x + w_0.$$

Sei w^* der Vektor $(w_0, w_1, \ldots, w_{n-1})$. Es stellt sich heraus, dass das zu lösende Optimierungsproblem gegeben ist durch:

$$\begin{aligned} \operatorname*{argmin}_{w^* \in \mathbb{R}^n} \quad & \tfrac{1}{2}||w^*||^2 \\ \text{s.t.} \quad & \forall i \; y_i \hat{y}(x_i) \geq 1 \end{aligned}$$

Wir wollen die Länge $||w^*||$ des Vektors w^* minimieren, da dies zu einem größeren Abstand zwischen unserer Ebene und den Datenpunkten führt, wodurch unser Modell besser generalisiert. Die Nebenbedingung entspricht der Anforderung, dass alle Punkte korrekt klassifiziert werden sollen.

1.2 Typische Problemklassen

Optimierungsprobleme können gemäß diverser Kriterien klassifiziert werden:

- Probleme mit Nebenbedingungen vs Probleme ohne Nebenbedingungen

- Optimierung mit Variablen aus verschiedenen Mengen, insbesondere kontinuierliche Variablen vs diskrete Variablen
- Lineare vs nichtlineare Funktionen
- Eindimensionale vs mehrdimensionale Funktionen
- Konvexe Funktionen vs nicht konvexe Funktionen
- Konvexe Mengen vs nicht konvexe Mengen

Diese verschiedenen Problemklassen führen zu unterschiedlich schwierigen Problemen. Insbesondere sind konvexe Probleme einfacher zu lösen als nicht konvexe Probleme, kontinuierliche Probleme sind in der Regel einfacher zu lösen als diskrete Probleme, und lineare Probleme sind einfacher als nichtlineare Probleme. Relevante Fragen sind dann:

- Wie schnell kovergiert das Verfahren zu einer Lösung? Wie viele Iterationen sind nötig? Was ist die Komplexität (in O-Notation) einer einzelnen Iteration?
- Konvergiert das Verfahren immer gegen ein Globales Optimum? Falls nein, gibt es garantierte obere/untere Schranken für die maximale Abweichung vom globalen Optimum?

1.3 Konvexität

Eine Menge G ist **konvex**, wenn für beliebige Punkte $x, y \in G$ auch beliebige lineare Interpolationen zwischen den Punkten in der Menge enthalten sind:

$$x, y \in G \implies \{(1-\alpha)x + \alpha y \mid \alpha \in [0,1]\} \subset G$$

Intuitiv entspricht das der Forderung, dass zwischen für alle Paare von Punkten eine Verbindungsstrecke zwischen den Punkten in der Menge enthalten sein muss.

Die **konvexe Hülle** einer Menge G ist die kleinste konvexe Menge H sodass $G \subset H$.

Analog zur Definition einer konvexen Menge ist eine **konvexe Funktion** definiert als eine Funktion, für die gilt:

$$x, y \in G \implies f((1-\alpha)x + \alpha y) \leq (1-\alpha)f(x) + \alpha f(y)\ \forall \alpha \in [0,1]$$

Dies entspricht der Forderung, dass jede Verbindungsstrecke zwischen zwei Punkten auf dem Graphen unterhalb des Graphen liegen muss.

Eine nicht konvexe Funktion, in der jedes lokale Minimum auch ein globales Minimum ist, wird als **quasikonvex** bezeichnet. Da dies dem Hauptvorteil von konvexen Funktionen in der Optimierung entspricht, ist die Optimierung von quasikonvexe Funktionen ebenfalls einfacher als die Optimierung allgemeiner nichtlinearer Funktionen.

Chapter 2

Gradientenverfahren

2.1 Wiederholung: Mehrdimensionale Differentiation

Zur Erinnerung: der **Gradient** ∇f ist ein Vektor, der alle partiellen Ableitungen von f enthält, also die Ableitung von f in alle Koordinatenrichtungen:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^T$$

Im eindimensionalen Fall entspricht das genau der gewöhnlichen Ableitung, da partielle Ableitungen letztendlich genau wie ordinäre Ableitungen berechnet werden. Sei $f_1(x_1, x_2, x_3)$ eine beliebige Funktion $\mathbb{R}^3 \to \mathbb{R}$ und sei $f_2(x_1) = f_2(x_1, x_2, x_3)$ für beliebige Konstante x_2, x_3. Dann ist

$$\frac{\partial}{\partial x_1} f_1(x_1, x_2, x_3) = \frac{d}{dx_1} f_2(x_1)$$

Zum Beispiel:

$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1^2 x_2 + x_1 x_3 + 2x_2 \\ \frac{\partial}{\partial x_1} f_1(x_1, x_2, x_3) &= 2x_1 x_2 + x_3 \end{aligned}$$

entspricht der Ordinären Ableitung

$$\begin{aligned} f_2(x_1) &= x_1^2 x_2 + x_1 x_3 + 2x_2 \\ \frac{d}{dx_1} f_2(x_1) &= 2x_1 x_2 + x_3 \end{aligned}$$

Eine mehrdimensionale Funktion $f : \mathbb{R}^n \to \mathbb{R}$ kann nicht nur in die Koordinatenrichtungen abgeleitet werden, sondern in beliebige Richtungen $\boldsymbol{d} \in \mathbb{R}^n$ Um Formeln verständlicher zu notieren, werde ich so gut es geht Variablen $\boldsymbol{x} \in \mathbb{R}^n$ im Fall $n > 1$ immer durch fette Buchstaben notieren.. Diese **Richtungsableitung** wird geschrieben als $\nabla_d f(x)$. Analog zur ordinären Definition der Ableitung ist die Richtungsableitung definiert als:

$$\nabla_d f(x) = \lim_{h \to 0} \frac{f(x + h\boldsymbol{d}) - f(x)}{h}$$

f wird als differenzierbar bezeichnet, falls die Richtungsableitung in allen Richtungen existiert. Für ein differenzierbares f kann die Richtungsableitung mit Hilfe des Gradienten sehr einfach als Skalarprodukt berechnet werden:

$$\nabla_d f(x) = \nabla f(x)^T \boldsymbol{d}$$

Im Rahmen dieser Vorlesung gehen wir davon aus, dass $f \in C^1$, also dass die zu minimierende Funktion mindestens einmal differenzierbar ist. Im Allgemeinen ist es jedoch wichtig, anzumerken, dass f nicht umbedingt differenzierbar ist, nur weil partielle Ableitungen in alle Richtungen existieren.

2.2 Iterative Optimierung

Eine Funktion $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ kann numerisch durch iteratives "Ausprobieren" verschiedener Werte von $\boldsymbol{x}$ optimiert werden. Wir starten mit einem beliebigen Startwert $\boldsymbol{x}^{(0)}$ [1] , und bewegen uns dann bei jedem Schritt k in eine Richtung $\boldsymbol{d}^{(k)} \in \mathbb{R}^n$, welche uns hoffentlich näher zum Optimum bringt. Wir nutzen zusätzlich einen Parameter $\tau^{(k)} \in \mathbb{R}$, welcher die Schrittweite beschreibt.

$$\begin{aligned} \boldsymbol{x}^{(0)} &\in \mathbb{R}^n \\ \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \tau^{(k)} \boldsymbol{d}^{(k)} \end{aligned}$$

In der Praxis wird oft einfach $\boldsymbol{x}^{(0)} = \mathbf{0}$ gewählt. Ist jedoch f nicht quasikonvex, so kann eine schlechte Wahl von $\boldsymbol{x}^{(0)}$ dazu führen, dass das Verfahren nur gegen ein lokales Minimum kovergiert, welches oft kein globales Minimum ist und somit in vielen Fällen ein schlechtes Ergebnis.

2.2.1 Welche Richtung ist optimal?

Das Optimale $\boldsymbol{x}^{(k+1)}$ ist das, das $f(\boldsymbol{x}^{(k+1)}$ minimiert. Wir können $f(\boldsymbol{x}^{(k+1)})$ für kleine Schrittweiten $\tau^{(k)}$ folgendermaßen approximieren (**Taylor-Approximation**):

$$f(\boldsymbol{x}^{(k+1)}) = f(\boldsymbol{x}^{(k)} + \tau^{(k)} \boldsymbol{d}^{(k)}) \approx f(\boldsymbol{x}^{k}) + \tau^{(k)} \nabla f(\boldsymbol{x}^{(k)}) \boldsymbol{d}^{(k)}$$

Da wir den Wert von f möglichst stark verringern wollen, sollte $\tau^{(k)} \nabla f(\boldsymbol{x}^{(k)}) \boldsymbol{d}^{(k)}$ ein negativer Term mit möglichst großem Betrag sein. Dabei können wir $\tau^{(k)}$ aber nicht zu hoch wählen, da sonst die Approximation ungenau wird.

Nach der Definition des Skalarprodukts gilt:

$$\nabla f(\boldsymbol{x}^{(k)}) \boldsymbol{d}^{(k)} = ||\nabla f(\boldsymbol{x}^{(k)})||\ ||\boldsymbol{d}^{(k)}|| \cos(\theta) \tag{2.1}$$

Wobei θ der Winkel zwischen $\nabla f(\boldsymbol{x}^{(k)})$ und $\boldsymbol{d}^{(k)}$ ist. Diese Gleichung ist minimal, wenn $\cos(\theta) = -1$, also $\theta = 180$. Dementsprechend muss $\boldsymbol{d}^{(k)}$ in die umgekehrte Richtung von $\nabla f(\boldsymbol{x}^{(k)})$ zeigen, also $\boldsymbol{d}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$. [2]

Mit dieser optimalen Wahl der Abstiegsrichtung $\boldsymbol{d}^{(k)}$ erhalten wir das **Gradientenverfahren** (auch bekannt als "Gradientenabstieg", auf Englisch "**Gradient Descent**"):

$$\begin{aligned} \boldsymbol{x}^{(0)} &\in \mathbb{R}^n \\ \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} - \tau^{(k)} \nabla f(\boldsymbol{x}^{(k)}) \end{aligned}$$

2.3 Bestimmung der Schrittweite

Die Wahl der Schrittweite $\tau^{(k)}$ ist für das Gradientenverfahren von entscheidender Bedeutung. Ist $\tau^{(k)}$ zu klein, wird die Konvergenz des Algorithmus oft deutlich verlangsamt. Ist $\tau^{(k)}$ zu groß, wird die Approximation ungenau. Dies kann dazu führen, dass das Verfahren über das Ziel hinausschießt, und im schlimmsten Fall eventuell gar nicht konvergiert.

Die Bestimmung eines optimalen Werts für $\tau^{(k)} \in \mathbb{R}$ ist ein eigenes Optimierungsproblem. In einigen Fällen ist ein dauerhaft konstantes τ genug, um die Konvergenz zu garantieren. In diesen Fällen ist eine konstante Schrittweite oft effizienter.

[1] Ich schreibe hier $\boldsymbol{x}^{(k)}$ mit der k in Klammern, um klarzustellen, dass $\boldsymbol{x}^{(k)}$ nicht "$\boldsymbol{x}$ hoch k" ist, sondern "das k-te $\boldsymbol{x}$". Eigentlich würde ich das am liebsten als $\boldsymbol{x}_k$ notieren, aber da $\boldsymbol{x} \in \mathbb{R}^n$ ist, ist diese Notation viel zu leicht mit der Notation x_k für die verschiedenen Komponenten von $\boldsymbol{x}$ zu verwechseln ($\boldsymbol{x} = (x_1, \ldots, x_n)^T$). Vermutlich werde ich an einigen Stellen aus Versehen trotzdem $\boldsymbol{x}^k$ schreiben, ich entschuldige mich im Voraus. (´-w-‘)

[2] Technisch gesehen wird die Gleichung durch $\boldsymbol{d}^{(k)} = -\alpha \cdot \nabla f(\boldsymbol{x}^{(k)})$ mit möglichst großen $\alpha \in \mathbb{R}$ minimiert. Die "Rolle" dieses Parameters α wir jedoch bereits durch unseren Schrittweiten-Parameter $\tau^{(k)}$ abgedeckt, und es gilt auch für dieses theoretische α, dass die Approximation sehr ungenau wird, wenn wir ein hohes α wählen.

Um ein optimales $\tau^{(k)}$ für eine gegebene Iteration k zu finden, definieren wir eine neue Funktion $h : \mathbb{R} \to \mathbb{R}$:

$$h(\tau) = f(\boldsymbol{x}^{(k)} + \tau \boldsymbol{d}^{(k)})$$

$$\tau^{(k)} = \underset{\tau \in \mathbb{R}}{\operatorname{argmin}} \; h(\tau)$$

Um die Ableitung von h zu finden, beschreiben wir den Term $\boldsymbol{x}^{(k)} + \tau \boldsymbol{d}^{(k)}$ als eine eigene eindimensionale Funktion $g(\tau)$. Dann gilt $h(\tau) = f(\boldsymbol{g(\tau)})$, gemäß der mehrdimensionalen Kettenregel gilt dann:

$$\begin{aligned}
\frac{d}{d\tau} h(\tau) &= \frac{d}{d\tau} f(\boldsymbol{g(\tau)}) \\
&= \sum_{i=1}^{n} \left(\frac{d}{d\tau} \boldsymbol{g(\tau)} \right) \frac{\partial}{\partial g(\tau)_i} f(\boldsymbol{g(\tau)}) \\
&= \sum_{i=1}^{n} d_i^{(k)} \frac{\partial}{\partial g(\tau)_i} f(\boldsymbol{g(\tau)}) \\
&= \nabla f(\boldsymbol{g(\tau)}) \cdot \boldsymbol{d}^{(k)} \\
&= \nabla f(\boldsymbol{x}^{(k)} + \tau \boldsymbol{d}^{(k)}) \cdot \boldsymbol{d}^{(k)}
\end{aligned}$$