



Technische Universität München

Fakultät für Mathematik



Bachelor-Arbeit

Titel

Markus Stachl

Aufgabensteller: ...

Betreuer: ...

Abgabetermin: ...

Ich erkläre hiermit, dass ich die Bachelor-Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe.

Garching, den

Summary

Bei einer in deutscher Sprache verfassten Arbeit muss eine Zusammenfassung in englischer Sprache vorangestellt werden. Dafür ist hier Platz.

Inhaltsverzeichnis

| | | |
|----------|---|----------|
| 1 | Inhalt | 1 |
| 2 | Einführung und Motivation | 1 |
| 2.1 | Optimale Kontrolle | 1 |
| 2.2 | Berechnung der Wertefunktion | 2 |
| 3 | Niedring-Rang-Appriximation einer Matrix | 3 |
| 3.1 | Singulärwertzerlegung (SVD) | 4 |
| 3.2 | CX-Zerlegung | 5 |
| 3.3 | CUR-Zerlegung | 5 |

1 Inhalt

Und los geht's

- Optimale Kontrolle/Optimales Feedback (Einführung)
 - set oriented approach
 - Konstruktion der Wertefunktion
 - Optimalitätsprinzip Bellman (Dynamic Programming (vgl (8)))
- Niedrig-Rank-Approximation einer Matrix
 - SVD
 - CUR
 - CX
 - Vorteile Nachteile
- CUR Decomp
 - Rating der Spalten/Zeilen
 - Algo
- Ausblick: NNCUR NNCX
- Fallbeispiel invertiertes Pendel

2 Einführung und Motivation

Die Vorarbeit und Basis zu dieser Arbeit besteht im Wesentlichen aus den beiden Papern...

2.1 Optimale Kontrolle

Gegeben sei das Problem der optimalen Stabilisierung eines (instabilen) diskreten Systems

$$x_{k+1} = f(x_k, u_k), \quad k = 0 \dots n \quad (1)$$

um einen Fixpunkt $0 \in X$, also $f(0, 0) = 0$. Der Fixpunkt kann o.B.d.A gewählt werden, ein anderer Fixpunkt kann gegebenenfalls in den Nullpunkt geshiftet werden. Die x_k sind aus dem Zustandsphasenraum $X \subset \mathbb{R}^n$, die u_k aus der Feedback-Menge $U \subsetneq \mathbb{R}^m$. X und U sind jeweils kompakte Teilmengen der Grundräume. Die Kosten pro Schritt sind dabei gegeben durch eine (stetige) Kostenfunktion $g : X \times U \rightarrow [0, \infty)$ mit $g(x, u) \geq 0 \ \forall x \in X, u \in U$ und $g(x, u) = 0 \Leftrightarrow x = 0, u = 0$.

Das Feedback u soll nun so gewählt werden, dass die entstehenden Kosten minimiert werden und der Fixpunkt 0 zu einem asymptotisch stabilen Fixpunkt wird. Anders ausgedrückt: ausgehend von einem Punkt $x \in X$ soll ein Kontrollvektor $u = (u_1, u_2 \dots)$ gewählt werden, sodass die Trajektorie

$$x_0(x, u) = x, \quad x_{k+1} = f(x_k(x, u), u_k), \quad k = 0, 1, \dots$$

für alle k in X verbleibt und $x_k(x, u) \rightarrow 0$ für $k \rightarrow \infty$ (vgl. (6)). Alle Punkte x , für die eine Stabilisierung möglich ist werden in der Menge

$$S = \{x \in X \mid \exists u \in U^{\mathbb{N}} : x_k(x, u) \rightarrow 0\}$$

zusammengefasst.

Somit ergibt sich für die gesamten Kosten zur Stabilisierung eines Punktes $x \in X$ entlang einer Trajektorie

$$J(x, u) = \sum_{k=0}^{\infty} q(x_k(x, u), u_k) \in [0, \infty] \quad (2)$$

Ist x nicht stabilisierbar, d.h. es existiert kein $u \in U^{\mathbb{N}}$ sodass $x_k(x, u) \rightarrow 0$, so ist $J(x, u) = \infty$ und $x \in X \setminus S$.

Gesucht sind nun die minimalen Kosten zur Stabilisierung (vgl. (7)) eines Punktes x , die Kostenfunktion

$$V(x) = \inf_{u \in U^{\mathbb{N}}} J(x, u) \quad (3)$$

2.2 Berechnung der Wertefunktion

Im Folgenden soll nun die zuvor definierte optimale Wertefunktion 3 numerisch berechnet werden. Hierzu wird das *Optimalitätsprinzip nach Bellman* (vgl. (1)) zu Hilfe genommen:

$$V(x) = \inf_{u \in U^{\mathbb{N}}} \{q(x, u) + V(f(x, u))\} \quad (4)$$

wobei sich die optimale Steuerungssequenz ergibt aus

$$u(x) = \operatorname{argmin}_{u \in U^{\mathbb{N}}} \{q(x, u) + V(f(x, u))\} \quad (5)$$

Zur numerischen Lösung dieses Optimierungsproblems wird ein Ansatz aus der Graphentheorie gewählt:

Die (kompakte) Menge X wird partitioniert, d.h. es entsteht eine Menge \mathcal{P} von Teilmengen $P_i \subset X, i = 1, \dots, l$, wobei $\cup_{i=1}^l P_i = X$ und $m(P_i \cap P_j) = 0$ für $i \neq j$ (m bezeichnet hier das Lebesgue-Maß).

Auf dieser Partition wird nun ein gerichteter Graph

$$G_{\mathcal{P}} = (\mathcal{P}, E_{\mathcal{P}}), \quad E_{\mathcal{P}} = \{(P_i, P_j) \in \mathcal{P} \times \mathcal{P} \mid f(P_i, U) \cap P_j \neq \emptyset\}$$

definiert. Hierbei sind die Gewichte der Kanten $e = (P_i, P_j)$ (vgl. (7)) alle nichtnegativ und gegeben durch

$$w(e) = \min_{x \in P_i, u \in U} \{q(x, u) \mid f(x, u) \in P_j\}.$$

Bild 1 zeigt schematisch solch einen Graphen ausgehend von einem Punkt x .

Bild des Graphen

Anstatt der exakten Wertefunktion $V(x)$ (4) wird nun die approximative Wertefunktion

$$V_{\mathcal{P}}(x) = \min\{w(p(x)) \mid \text{Pfad } p(x) = (e_1, \dots, e_m), e_k \in E_{\mathcal{P}} \text{ verbindet } x \text{ mit dem Ursprung}\}$$

berechnet.

Um, ausgehend von dem bestimmten Graphen, den kürzesten Weg und damit die bestmögliche Stabilisierung bezüglich der Kosten zwischen einem Punkt x und einem Fixpunkt 0 zu bestimmen, können Standard-*shortest-path*-Algorithmen wie beispielsweise der Algorithmus von Dijkstra (vgl. (2)) verwendet werden.

Somit kann für jedes Element der Partition \mathcal{P} eine optimale Stabilisierung, d.h. bezüglich der geringsten Kosten, berechnet werden.

Konvergenz: Für alle Punkte $x \in S$, für die eine Stabilisierung möglich ist, und für den Partitionsdurchmesser $diam(\mathcal{P}) := \max_i \{diam(P_i)\} \rightarrow 0$ gilt:

$$V_{\mathcal{P}}(x) \rightarrow V(x)$$

Für einen Beweis siehe (7).

Schreibt man den resultierenden Boxplot in Matrixform, erhält man Folgendes:

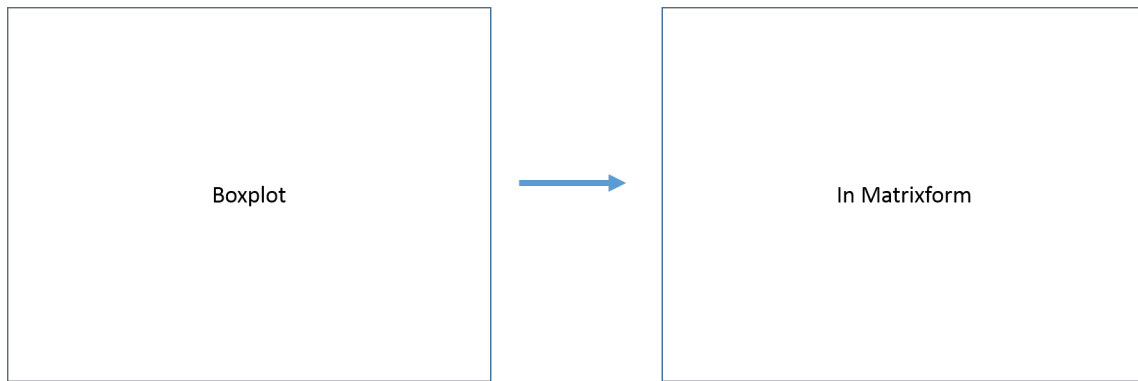


Abbildung 1: Umformung des Boxplots in Matrixform

Da diese Matrizen je nach Parameter ziemlich groß werden können, sollen sie approximiert werden. Im nächsten Kapitel werden diverse Approximationsmethoden vorgestellt.

3 Niedring-Rang-Approximation einer Matrix

Je nach Tiefe der Simulation bzw. der Dimension des zugrunde liegenden Raumes (vgl. Fluch der Dimension **Bellman**) können gigantische Matrizen entstehen. Da die gängigen Rechner bei der Speicherung solcher Matrizen schnell an ihre Grenzen stoßen, soll nun für die Wertematrix V eine Niedrig-Rang-Approximation der Form

$$V(x) \approx \sum_{i=1}^k \sigma_i u_i(x) v_i^T(x) \quad (6)$$

gefunden werden, wobei $\sigma_i \in \mathbb{R}$ Konstanten und u_i, v_i Vektoren im \mathbb{R}^n sind. Im Folgenden werden einige Näherungen vorgestellt.

3.1 Singulärwertzerlegung (SVD)

Dies klassische Singulärwertzerlegung und ist folgendermaßen definiert.

Definition 3.1 Sei A eine komplexe $m \times n$ -Matrix von Rang r . Die Singulärwertzerlegung von A ist gegeben durch (vgl. (1))

$$A = U\Sigma V^*$$

wobei

- U eine unitäre $m \times m$ -Matrix ist,
- V^* die Adjungierte einer $n \times n$ -Matrix V ist und
- Σ eine reelle $m \times n$ -Diagonalmatrix der Form

$$\Sigma = \left(\begin{array}{ccc|ccc} \sigma_1 & & & & \vdots & \\ & \ddots & & \dots & 0 & \dots \\ & & \sigma_r & & \vdots & \\ \hline & \vdots & & & \vdots & \\ \dots & 0 & \dots & \dots & 0 & \dots \\ & \vdots & & & \vdots & \end{array} \right)$$

Die Einträge $\sigma_i > 0$, $i = 1 \dots r$, $\sigma_1 > \dots > \sigma_r$ heißen Singulärwerte von A . Die Spalten von U , u_i , heißen linke Singulärvektoren, die Spalten von V , v_i , heißen rechte Singulärvektoren.

Bemerkung 1 A kann auch geschrieben werden als die Summe von Rang-1-Matrizen:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

Die σ_i sind dabei die Singulärwerte von A und u_i und v_i die Zeilen von U bzw. V .

Die Berechnungskosten, die für das Ausführen der Matrixzerlegung benötigt werden, betragen $\mathcal{O}(\min(mn^2, m^2n))$ (vgl (8)).

Wir wollen die Wertematrix V nun so approximieren, dass der Fehler

$$e(D) = \|V - D\|_\xi$$

bezüglich einer Norm ξ möglichst klein wird für Matrizen D von Rang $\text{rank}(D) \leq k$. $\xi = 2$ bezeichnet hierbei die 2-Norm ($\|A\|_2 = \left(\sum_{i,j} |a_{ij}|^2\right)^{\frac{1}{2}}$), $\xi = \max$ bezeichnet die Maximumsnorm ($\|A\|_{\max} = \max_{i,j} |a_{ij}|$) und $\xi = F$ verwendet die Frobeniusnorm ($\|A\|_F = (\sigma_1^2 + \dots + \sigma_r^2)^{\frac{1}{2}}$).

Die erste Frage, die auftaucht: gibt es eine "beste" Rang-k-Approximation.

Satz 3.2 (Eckart-Young) Sei eine Matrix V von Rang r gegeben durch 3.1 und $k < r$. Die beste Rang- k -Approximation von V , also eine Matrix, die

$$\min_{\text{rank}(B) \leq k} \|V - B\|_2$$

erfüllt, ist gegeben durch

$$V_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

und

$$\|V - V_k\|_2 = \sigma_{k+1}$$

wobei σ_{k+1} der $(k+1)$ -te Singulärwert von V ist. Dies impliziert, dass für Singulärwertzerlegungen von Rang $k \geq r$ gilt: $\|V - V_k\|_2 = 0$.

Beweis 1 Für einen Beweis siehe (5).

Die Kosten der k -Rang-Singulärwertzerlegung betragen $\mathcal{O}(mnk)$ (9).

Nachteil: Es werden keine konkreten Einträge der ursprünglichen Matrix V verwendet, deswegen.....

In den nächsten beiden Abschnitten werden zwei Verfahren vorgestellt, welche Spalten und/oder Zeilen der Eingabematrix verwenden.

3.2 CX-Zerlegung

Die erste approximative Matrix-Zerlegung, welche konkrete Einträge der ursprünglichen Matrix verwendet, ist die CX-Zerlegung (vgl. (4)).

Definition 3.3 Sei A eine $m \times n$ -Matrix. Sei $C \in \mathbb{R}^{m \times c}$ eine Matrix bestehend aus c Spalten der Matrix A (in der Regel ist $c \ll n$). Dann ist die approximative Matrix $A' = CX$ für jede beliebige Matrix $X \in \mathbb{R}^{c \times n}$ eine spaltenbasierte Approximation von A , beziehungsweise eine CX-Zerlegung von A .

3.3 CUR-Zerlegung

Bei der CUR-Zerlegung werden im Gegensatz zur CX-Zerlegung Spalten und Zeilen der zu approximierenden Matrix verwendet. Die Matrix C enthält dabei Spalten von V und Zeilen werden in R gefasst. Die Matrix U wird dann so gewählt, dass der Fehler zwischen V und CUR bezüglich einer Norm minimal wird.

Die mathematische Definition der CUR-Zerlegung ist wie folgt:

Definition 3.4 (vgl. (4)) Gegeben sei eine Matrix $V \in \mathbb{R}^{m \times n}$. Sei $C \in \mathbb{R}^{m \times c}$ eine Matrix, welche aus c Spalten von V besteht, und $R \in \mathbb{R}^{r \times n}$ eine Matrix, deren r Zeilen aus Zeilen von V bestehen. Dann ist die Matrix $V' = CUR$ für jede beliebige Matrix $U \in \mathbb{R}^{c \times r}$ eine Spalten-Zeilen-basierte Approximation von V , beziehungsweise eine CUR-Zerlegung von V .

Im Folgenden wird ein Verfahren zur Berechnung der *CUR*-Zerlegung von Mahoney und Drineas (8) vorgestellt.

Besonders kritisch bei der Berechnung der Zerlegung ist, welche Zeilen bzw. Spalten für die Approximation verwendet werden. Dafür wird ein Rating erzeugt (siehe (8)), welche den Einfluss einer Spalte/Zeile auf die restlichen Einträge der Matrix beschreibt.

Bestimmung des statistischen Einflusswertes einer Spalte/Zeile:

Aus 1 wissen wir, dass sich die j -te Spalte einer Zielmatrix V von Rang r darstellen lassen kann als

$$V^j = \sum_{\xi=1}^r (\sigma_{\xi} u_{\xi}^j) v_j^{\xi}$$

bzw. deren "beste" Rang- k -Approximation durch

$$V^j = \sum_{\xi=1}^k (\sigma_{\xi} u_{\xi}^j) v_j^{\xi}$$

wobei u_{ξ}^j der ξ -te linke Singulärvektor und v_j^{ξ} die j -te Koordinate des ξ -ten rechten Singulärvektors ist. Der statistische Einflusswert einer Spalte von V kann nun gemessen werden durch

$$\pi_j = \frac{1}{k} \sum_{\xi=1}^k (v_j^{\xi})^2 \quad j \in [1, n] \quad (7)$$

Diese Einflusswerte bilden eine Wahrscheinlichkeitsverteilung auf den n Spalten und es gilt: $\sum_{j=1}^n \pi_j = 1$ und $\pi_j \geq 0$

Das Selektieren der Spalten wird anschließend mithilfe des Algorithmus' COLUMNSELECT (vgl. (8)) durchgeführt:

- (i) **Input:** Matrix $V \in \mathbb{R}^{m \times n}$ und Fehlerparameter ϵ
- (ii) Berechne die ersten k rechten Singulärvektoren (v_1, \dots, v_k) von V und die statistischen Einflusswerte π_j der Spalten von V .
- (iii) Verwende die j -te Spalte von V mit Wahrscheinlichkeit $p_j = \min\{1, c\pi_j\}$, wobei $c = \mathcal{O}(k \log \frac{k}{\epsilon^2})$
- (iv) **Output:** Matrix C , welche aus den selektierten Spalten von A besteht.

Mithilfe dieses Algorithmus' entsteht eine Matrix C , die $c' \leq c$ der selektierten Spalten verwendet. Die Kosten zur Ausführung des Algorithmus' beruhen hauptsächlich auf der Berechnung der ersten k Singulärvektoren und sind daher $\mathcal{O}(mnk)$ (8).

Zur Berechnung der Vektoren C, U und R wird abschließend folgender Algorithmus ALGORITHMCUR (vgl. (8)) verwendet:

- (i) Wende COLUMNSELECT auf A an mit $c = \mathcal{O}(k \log \frac{k}{\epsilon^2})$ und berechne somit die Matrix C

- (ii) Wende COLUMNSELECT auf A^T an mit $r = \mathcal{O}(k \log \frac{k}{\epsilon^2})$ und berechne somit die Matrix R
- (iii) Definiere die Matrix U als $U = C^+ A R^+$, wobei X^+ die Moore-Penrose-Inverse von X ist

Die so konstruierte approximative Matrix $V' = CUR$ erfüllt nun, dass

$$\|V - V'\|_F \leq (2 + \epsilon) \|V - V_k\|_F$$

mit hoher Wahrscheinlichkeit (m.h.W.) (vgl. (8)).

Die Rate $(2 + \epsilon)$ kann mit einer Variation des Algorithmus' noch weiter verbessert werden (4). Bei dieser Methode wird die Matrix R abhängig von den für die Matrix C gewählten Zeilen berechnet. Durch diese Abwandlung verringert sich der Fehler m.h.W. zu

$$\|V - CUR\|_F \leq (1 + \epsilon) \|V - V_k\|_F$$

Dies ist nur eine Möglichkeit eine approximative CUR-Zerlegung der Matrix V zu finden. Weitere werden in folgender Tabelle gezeigt:

| | Anzahl Spalten c | Anzahl Reihen r | rank(U) | $\ V - CUR\ _F^2 \leq$ |
|-------------------------|-------------------------------|-------------------------------|-------------------------------|--|
| Drineas et al. 2003 (3) | $\frac{k}{\epsilon^2}$ | $\frac{k}{\epsilon^2}$ | k | $\ V - V_k\ _F^2 + \epsilon \ A\ _F^2$ |
| Drineas et al. 2008 (4) | $\frac{k \log k}{\epsilon^2}$ | $\frac{k \log k}{\epsilon^4}$ | $\frac{k \log k}{\epsilon^2}$ | $(1 + \epsilon) \ V - V_k\ _F^2$ |
| Wang et al. 2013 (9) | $\frac{k}{\epsilon}$ | $\frac{k}{\epsilon^2}$ | $\frac{k}{\epsilon}$ | $(1 + \epsilon) \ V - V_k\ _F^2$ |

Abbildung 2: Tabelle über verschiedene CUR-Approximationen

Vorteil: Zeilen und Spalten der ursprünglichen Matrix werden verwendet

Nachteil: Die approximierende Matrix ist nicht mehr von der Form 6.

Abkürzungen

SVD Singulärwertzerlegung
m.h.W. mit hoher Wahrscheinlichkeit

Literatur

- [1] Deuffhard, Peter, & Hohmann, Andreas. 2008. *Numerische mathematik 1: Eine algorithmisch orientierte einföhrung*. 4. überarb. u. erw. aufl edn. de Gruyter Lehrbuch. Berlin u.a.: de Gruyter.
- [2] Dijkstra, E.W. 1959. *A note on two problems in connection with graphs*.
- [3] Drineas, Petros, & Kannan, R. 2003. Pass efficient algorithms for approximating large matrices. *Proc. of the 14th annual acm-siam symposium on discrete algorithms (soda)*, 223–232.
- [4] Drineas, Petros, Mahoney, Michael W., & Muthukrishnan, S. 2009. Relative-error cur matrix decompositions. *Siam journal on matrix analysis and applications*, **30**(3), 844–881.
- [5] Golub, Gene H., & Van Loan, Charles F. 2013. *Matrix computations*. 4. ed. edn. Johns Hopkins studies in the mathematical sciences. Baltimore: Johns Hopkins Univ Press.
- [6] Grüne, Lars, & Junge, Oliver. 2005. A set oriented approach to optimal feedback stabilization. *Systems & control letters*, **54**(2), 169.
- [7] Junge, O., & Osinga, H. M. 2004. A set oriented approach to global optimal control. *Control, optimisation and calculus of variations : Cocv*, **10** (2004), 259–270.
- [8] Mahoney, Michael W., & Drineasb, Petros. 2008. Cur matrix decompositions for improved data analysis. *Pnas*, **2009**(vol. 106), 697–702.
- [9] Wang, Shusen, & Zang, Zhihua. 2013. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *Journal of machine learning research*, **2013**(14), 2729–2769.