Technische Universität Chemnitz Chemnitz, 30.11.2009
Prof. Dr. C. Helmberg, A. Lau

Optimierung für Nichtmathematiker
Übung 7

1. Laden Sie die Matlab-Ergänzungen SDPT3 version 4.0 (beta) – a MATLAB software for semidefinite-quadratic-linear programming auf
`http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`
sowie CVX: Matlab Software for Disciplined Convex Programming auf
`http://www.stanford.edu/~boyd/cvx/download.html`
herunter und entpacken Sie die Dateien.
   (i) SDPT3: Gehen Sie mit Matlab in den eben erzeugten Ordner und führen Sie den Befehl `Installmex` aus.
   (ii) CVX: Gehen Sie mit Matlab in den eben erzeugten Ordner und führen Sie den Befehl `cvx_setup` aus.

2. **Beispiel**: Gegeben ist eine Matrix $A \in \mathbb{R}^{m\times n}$ und ein Vektor $b \in \mathbb{R}^m$. Gesucht wird ein $x \in \mathbb{R}^n$, so dass die Norm $\|Ax-b\|_2$ minimal wird. Als Optimierungsproblem ergibt sich

$$\begin{aligned} &\min \|Ax-b\| \\ &\text{s.t. } x \in \mathbb{R}^n \end{aligned}$$

bzw.

$$\begin{aligned} &\min x_0 \\ &\text{s.t. } x_0 \geq \|Ax-b\| \\ &\qquad x_0 \in \mathbb{R}, x \in \mathbb{R}^n \end{aligned}$$

oder

$$\begin{aligned} &\min x_0 \\ &\text{s.t. } \bar{x} = Ax-b \\ &\qquad \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_{\mathcal{Q}} 0, x \in \mathbb{R}^n \end{aligned}$$

Erzeugen Sie in Matlab zufällige A, b
`m = 10; n = 5; A = randn(m,n); b = randn(m,1);`
und untersuchen Sie die Ergebnisse von

```
cvx_begin
  variable x(n);
  minimize (norm(A*x-b));
cvx_end
x
```

und

```
cvx_begin
  variable x(n);
  variable x_0;
  minimize x_0;
  subject to
    x_0 >= norm(A*x-b);
cvx_end
x
```

CVX: Jeder Aufruf muss mit `cvx_begin` begonnen werden, `cvx_end` schließt das Problem ab und ruft den Solver auf. Für die Zielfunktion stehen `minimize/maximize` zur Verfügung, die Nebenbedingungen werden durch `subject to` eingeleitet.

3. **Methode der kleinsten Quadrate**: Gegeben seien Messpunkte $(x_i, y_i), i = 1, \ldots, n$. Gesucht wird ein Polynom $f(x) = a_0 + a_1 \cdot x + \ldots + a_d x^d$ vorgegebenen Grades $d < n$, so dass die Summe der quadratischen Abstände jedes Punktes zum Polynom, also $\sum\limits_{i=1}^{n} \|f(x_i) - y_i\|_2^2$, minimiert wird. Suchen wir ein quadratisches Polynom, so ergibt sich

$$\min_{a_0,a_1,a_2} \sum_{i=1}^{n} \|a_0 + a_1 \cdot x_i + a_2 \cdot x_i^2 - y_i\|_2^2,$$

wobei man beachte, dass x_i, y_i vorgegebene Werte sind. Nutzen wir die vorherige Aufgabe, so können wir dies auch einfacher schreiben als $\min\limits_{a_0,a_1,a_2} \|Xa - y\|_2$ mit

$$a = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix}^T, \quad y = \begin{bmatrix} y_1 & y_2 & \ldots & y_n \end{bmatrix}^T, \quad X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}.$$

Aufgabe: Bestimmen Sie das quadratische Polynom $p(x)$, so dass die Summe der quadratischen Abstände der folgenden Messwerte zu $p(x)$ minimiert wird.

x_i	-2	-1	0	1	2
y_i	5	4	2	5	6

4. Gegen sind k Punkte $x_1, \ldots, x_k \in \mathbb{R}^n$. Gesucht ist ein Punkt $x \in \mathbb{R}^n$, so dass die Summe der quadratischen Abstände dieses Punktes zu den anderen Punkten minimiert wird. Stellen Sie ein Optimierungsmodell auf und lösen Sie dies mit Matlab für zufällig erstellte Instanzen. Wie ändert sich die Lösung, wenn man statt der 2-Norm die 1-Norm oder die Supremumsnorm nimmt?

5. Gegeben sei eine symmetrische Matrix A, wobei einige Einträge bereits festgelegt wurden. Wie muss man die Einträge ergänzen, damit A eine positiv semidefinite Matrix wird und der Logarithmus der Determinante von A ($\log(\det(A))$) maximiert wird? Stellen Sie ein Optimierungsmodell auf! Testen Sie Ihr Modell mit Matlab für

$$A = \begin{bmatrix} 1 & 0.5 & & \\ 0.5 & 2 & 0.75 & \\ & 0.75 & 3 & \\ & & & 4 \end{bmatrix}.$$

```
cvx_begin sdp
  variable A(n,n) symmetric;
  A >= 0;
  A(1,1) == 1; A(2,2) == 2; A(3,3) == 3; A(4,4) == 4; A(1,2) == 0.5; A(2,3) == 0.75;
  maximize( log_det( A ) )
cvx_end
```

6. Gegeben seien zwei symmetrische Matrizen $A, B \in \mathbb{R}^{n \times n}$, B positiv semidefinit. Gesucht ist das maximale t, so dass $A - tB \succeq 0$. Stellen Sie ein Optimierungsmodell auf. Testen Sie Ihr Modell mit Matlab, wobei `A=randn(4), A= A'+A, B=randn(4), B=B'*B`. Welche Werte kann t annehmen?