HEMUPPGIFT I

1. Instruktioner

I denna hemuppgift är det tillåtet att jobba i grupper om högst tre personer. En rapport per grupp skall lämnas in, i vilken ni skall lösa nedanstående uppgifter. En utskrift av MATLAB-filen och plottar skall vara inkluderad i rapporten.

Skriv namn, personnummer och email-adresser till alla medlemmar i gruppen på framsidan av rapporten.

Rapporten skall lämnas in till en av lärarna på lektionen (13:00–15:00) fredagen den 17 september, 2010. Denna deadline är inte förhandlingsbar.

2. Inledning

I spektralskattning analyseras frekvensinnehållet av en signal.

2.1. Periodogramskattningen. Antag att dataföljden

$$y_0, \dots, y_N$$

är given. I MATLAB kan man skatta spektraltätheten Φ av signalen $y = (y_0, \dots, y_N)$ med

```
>> Phi = abs(fft(y)).^2;
```

(Man kan säga att spektraltätheten mäter signalens energiinnehåll vid frekvensen θ.) Denna uppskattning kallas för *periodogramskattningen* för signalen y. Det är vanligt att använda deciBel (dB) för att uttrycka spektraltätheten för ljudsignaler. Därför definierar vi

$$\Psi = 10 \log_{10} \Phi.$$

2.2. En rationell modell. Nu är uppgiften att anpassa en rationell modell till den skattade spektraltätheten Ψ ovan, det vill säga, vi vill hitta en $\widehat{\Psi}$, sådan att $\widehat{\Psi}$ är nära Ψ, och dessutom,

$$\widehat{\Psi}(\theta) = \frac{S(\theta)}{T(\theta)}, \quad \theta \in [0, 2\pi],$$

där

$$\begin{aligned} S(\theta) &= p_0 + p_1 \cos\theta + \cdots + p_n \cos(n\theta), \\ T(\theta) &= 1 + q_1 \cos\theta + \cdots + q_n \cos(n\theta). \end{aligned}$$

Varför vill vi göra det? Målsättningen är att den långa dataföljden y (som har N komponenter), kan ersättas med den korta följden med komponenter $p_0, \dots, p_n$ och $q_1, \dots, q_n$, och om $2n + 1 \ll N$, så har vi sparat en hel del plats i minnet. Denna idé används i talbehandling för att karakterisera korta talsegment. Saken är den att om vi jämför två samplingar där en person säger samma ord två gånger, så är dataföljderna i tidsdomänen ganska olika, medan frekvensdomänbilderna är väldigt lika, och en approximation av frekvensdomänbilden är tillräcklig för att reproducera ljudet av ordet. Därför är det vettigt att approximera Ψ. På detta sätt kan den rationella modellen användas för taligenkänning, röstigenkänning, talkompression, och så vidare.

2.3. **Problemet.** Så vad vi helst vill göra är att minimera något slags norm av differensen mellan Ψ och $\widehat{\Psi}$, det vill säga $\|\Psi - \widehat{\Psi}\|$. Men just nu vill vi använda oss av linjärprogrammering för att lösa problemet. Så i stället för att göra detta, så betraktar vi problemet att minimera normen av

$$T(\Psi - \widehat{\Psi}) = T\Psi - S.$$

Vi kommer att ange normen nedan.

Till att börja med kommer vi att använda oss av en standardljudsignal som vår signal y i Matlab. Denna ljudsignal kallas för *handel*, och är ett utdrag från när en kör sjunger Händels Messias. För att få periodogramuppskattningen Ψ för detta y, så går vi tillväga på följande sätt:

```
>> load handel;
>> N = 1024;
>> Psi = 10*log10((1/N)*abs(fft(y(10001:10000+N))).^2);
>> Psi = Psi(1:N/2);
```

(Man kan lyssna på det y vi använder genom att använda kommandot `sound(y,Fs)`. Eftersom vår signal y är en reell följd, så räcker det med att ta de första $\frac{N}{2}$ komponenterna av Ψ; detta är anledningen till att det står $N/2$ i det fjärde MATLAB-kommandot ovan.)

Ovanstående kommandosekvens i MATLAB ger oss en vektor med värden $\Psi(\theta_k)$, där

$$\theta_k = \frac{k\pi}{\frac{N}{2}}, \quad k = 0, \dots, \frac{N}{2} - 1.$$

Vi kommer att använda ℓ^1-normen för följder, d v s om vi har en följd $f = (f_k)_{0 \le k \le \frac{N}{2}-1}$, så är

$$\|f\|_1 := \sum_{k=0}^{\frac{N}{2}-1} |f_k|.$$

Eftersom vi betraktar problemet att minimera normen av $\Psi T - S$, så leds vi till följande optimeringsproblem:

$$(P): \text{ minimera } \left\| \left(\Psi(\theta_k) \cdot T(\theta_k) - S(\theta_k) \right)_{0 \le k \le \frac{N}{2}-1} \right\|_1 . \tag{1}$$

Variablerna in ovanstående minimeringsproblem är koefficienterna $p_0, \dots, p_n$ och $q_1, \dots, q_n$.

3. Uppgifter

(1) Skriv om optimeringsproblemet (P) i (1) som ett linjärprogrameringsproblem.

Motivera din formulering, d v s, förklara varför ditt linjärprogrammeringsproblem löser det ursprungliga optimeringsproblemet (P). I ditt linjärprogrammeringsproblem ska du tydligt specificera variablerna, bivillkoren och målfunktionen.

(2) Använd Optimization toolbox i MATLAB för att lösa ditt linjärprogrammeringsproblem. Använd $n = 8$. MATLAB-kommandot `help linprog` visar en hjälptext. Problemet

$$\begin{cases} \text{minimize} & c^\top x, \\ \text{subject to} & Ax \leq b, \end{cases}$$

kan lösas med hjälp av simplexmetoden genom att använda följande kommandon:

```
>> options=optimset('Simplex','on','Largescale','off','Display','iter')
>> [x,fval,exitflag,output]=linprog(c,A,b,[],[],[],[],[],options);
```

(3) Plotta $(\Psi(\theta_k) \cdot T(\theta_k))_{0 \leq k \leq \frac{N}{2}-1}$ och $(S(\theta_k))_{0 \leq k \leq \frac{N}{2}-1}$ i samma graf. Vad observerar du?

(4) Plotta $(\Psi(\theta_k))_{0 \leq k \leq \frac{N}{2}-1}$ och $(\widehat{\Psi}(\theta_k))_{0 \leq k \leq \frac{N}{2}-1}$ i samma graf. Vad observerar du?

(5) Repetera ovanstående uppgifter för ett annat segment av ljudsignalen y. Till exempel, i stället för att ladda `handel`, så kan man använda `chirp`, `laughter`, eller `gong`.