

Domaći zadatak iz interpolacije

Tabela 1 prikazuje podatke o vremenskim uslovima na međunarodnom aerodromu u Vankuveru za novembar 2014.godine koji su javno dostupni na <http://climate.weather.gc.ca/> i mere se svakog sata.

Table 1: weather data record for YVR for November 1, 2014 from <http://climate.weather.gc.ca/>

Time	Temp (°C)	Dew Point Temp (°C)	Relative Humidity (%)	Wind Dir. (10s deg)	Wind Speed (km/h)	Visibility (km)	Pressure (kPa)
0:00	9.2	8.1	93	8	9	16.1	101.15
1:00	9.6	7.7	88	7	8	24.1	101.17
2:00	9.0	7.1	88	9	9	24.1	101.17
3:00	7.7	5.7	87	6	8	24.1	101.13
4:00	6.8	5.6	92	1	5	24.1	101.15
5:00	6.0	5.1	94	10	8	24.1	101.16
6:00	5.9	5.0	94	9	3	24.1	101.16
7:00	4.3	3.9	97	35	8	19.3	101.19
8:00	7.2	5.7	90	36	9	24.1	101.22
9:00	8.9	6.4	84	36	3	24.1	101.23
10:00	9.2	6.7	84	28	3	24.1	101.24
11:00	10.2	7.1	81	30	17	24.1	101.27
12:00	10.9	8.5	85	29	16	24.1	101.25
13:00	10.6	8.9	89	26	9	24.1	101.25
14:00	10.6	8.5	87	26	16	24.1	101.25
15:00	10.7	8.8	88	28	12	24.1	101.31
16:00	10.3	8.3	87	24	7	24.1	101.32
17:00	8.6	7.2	91	18	10	24.1	101.37
18:00	8.2	6.7	90	14	11	24.1	101.43
19:00	8.0	6.6	91	14	12	24.1	101.48
20:00	8.8	7.6	92	15	13	24.1	101.54
21:00	7.1	5.9	92	10	8	24.1	101.58
22:00	7.5	6.5	93	10	4	24.1	101.61
23:00	7.5	6.6	94	8	7	24.1	101.64

Zadatak 1: Iz Tabele 1 iskopirati vrednosti temperature u parnim satima (0:00, 2:00, ..., 22:00) u vektor temp_even, I vrednosti temperature u neparnim satima u vektor temp_odd. Takođe, sačuvajte parne sate u vektor time_even i neparne sate u vector time_odd. Cilj je interpolirati vrednosti u neparnim satima i izračunati grešku interpolacije koristeći stvarne vrednosti u neparnim satima.

Zadatak rešiti u tri koraka:

1) Fitovati linearnom funkcijom podatke u `temp_even`:

```
P = polyfit(time_even,temp_even,1);
```

2) Koristeći funkciju `polyval`, izračunati vrednosti dobijenog polinoma P u `time_odd`.
Sačuvati dobijene rezultate u vektoru `temp_int_odd`.

3) Izračunati srednju kvadratnu grešku između pravih vrednosti u `temp_odd` i interpoliranih vrednosti u `temp_int_odd` koristeći formulu:

```
MSE = sum((temp_int_odd - temp_odd).^2)/length(temp_odd)
```

Kada uradite ova tri koraka skicirajte sledeće tri krive: (1) **plavu** krivu koja prikazuje stvarne vrednosti temperature u svim satima od 00 do 23 časa, (2) **crvenu** krivu koja prikazuje stvarne vrednosti u parnim satima i interpolirane vrednosti u neparnim satima i (3) pravu liniju prikazanu **zelenom**, koja predstavlja fitovani polinom P izračunat u svim satima.

Šta zaključujete iz grafika? Da li su interpolirane vrednosti bliske pravim u neparnim satima? Da li je linearna interpolacija pogodna za ove podatke?

Zadatak 2: Napraviti for petlju za tri koraka iz Zadatka 1. Brojač u petlji treba da bude stepen polinom N i neka uzima vrednosti N=1:9. Ne zaboravite da čuvate vrednosti za grešku za različito N u različitim vektorima, možda najbolje da napravite vector dužine 9:

```
MSE(N) = sum((temp_int_odd - temp_odd).^2)/length(temp_odd);
```

Kad prodjete kroz petlju nacrtajte

```
plot(1:9, MSE);
```

Označite ose i analizirajte oblik dobijene krive. Koji je stepen polinoma koji najbolje interpolira date podatke?

Task 3: Sada ćemo razmotriti bolji način interpolacije. Intuitivno je jasno da će vrednost temperature u 7 sati biti najbliža temperaturama u 8 ili u 8 sati, a manje slična temperature u 22 časa. U ovom zadatku primenjivaćete *nearest neighbor interpolation*. Predstavlja veoma jednostavan metod koji se koristi često u praksi. Na primer, da bismo interpolirali vrednost temperature u 7 potrebne su nam vrednosti u 6 ili u 8 časova. Za početak, izaberimo prethodnika, dakle vrednost u 6 časova i prosto je prekopirajte u 7 časova.

Izračunajte ovako dobijenu grešku MSE i uporedite je sa prethodno dobijenim vrednostima grešaka.

Nacrtajte sledeće krive: (1) **plavom** bojom krivu koja prikazuje stvarne vrednosti temperature iz tabele, (2) **crvenom** bojom krivu koja prikazuje stvarne temperature u parnim satima i interpolirane temperature u neparnim satima. Prokomentarišete dobijene krive.

Zadatak 4: U ovom zadatku primenjivaćemo *local linear interpolation*. Za svaku traženu vrednost koristićemo prethodnu i narednu vrednost i traženu vrednost dobiti linearnom interpolacijom između posmatrane dve vrednosti. U našem slučaju je situacija još dodatno pojednostavljena jer se tražena vrednost uvek nalazi na sredini i može se dobiti kao srednja vrednost dve susedne vrednosti. Na primer, vrednost temperature u 7 možemo dobiti kao srednju vrednost temperatura u 6 i u 8 časova. Za poslednju interpolaciju u 23 časa koristite samo 22 časa jer nema sledbenika.

Izračunajte grešku MSE i ovako dobijene procene.

Skicirajte sledeće dve krive: (1) **plavom** bojom krivu koja prikazuje stvarne temperature iz tabele za sve sate (0:00, 1:00, ..., 23:00), (2) **crvenom** bojom krivu koja prikazuje stvarne temperature u parnim satima i interpolirane vrednosti temperature u neparnim satima. Koliko dobro se krive poklapaju?

Task 5: Šta se dobija ako, umesto linearne interpolacije, između susedne dve vrednosti interpoliramo polinomom višeg stepena? Međutim, tada nam nisu dovoljne dve vrednosti, već nam treba $N+1$ vrednost, gde je N stepen polinoma kojim fitujemo. Ovakva interpolacija zove se *piecewise polynomial interpolation*.

Poseban tip ove interpolacije polinomom trećeg stepena zove se *cubic spline interpolation*. MATLAB funkcija koja tako računa zove se spline. Pogledajte `help spline`.

Ponovite Zadatak 4 koristeći spline interpolaciju.

Sve u svemu, koji vid interpolacije procenjujete kao najbolji u ovom primeru?