UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Dejan Petelin

Sprotno učenje modelov na podlagi Gaussovih procesov

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Janez Demšar

Somentor: prof. dr. Juš Kocijan

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01588/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DEJAN PETELIN**

Naslov: **SPROTNO UČENJE MODELOV NA PODLAGI GAUSOVIH PROCESOV**
INCREMENTAL LEARNING OF GAUSSIAN PROCESS MODELS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Gaussovi procesi so stohastični procesi, v katerih so vektorji neodvisnih spremenljivk porazdeljeni po Gaussovi porazdelitvi, korelacije med njimi pa določa kovariančna funkcija. Gaussovi procesi se vedno pogosteje uporabljajo tudi kot osnova za modeliranje v regresijskih (in tudi klasifikacijskih) problemih.

Glavna omejitev pri učenju Gaussovih procesov je njegova časovna zahtevnost. V diplomi preglejte metode za pospešitev učenja Gaussovih procesov in izberite tiste, na podlagi katerih je mogoče izvesti sprotno (inkrementalno) učenje. Izbrane metode ovrednotite na primernem izboru umetnih in realnih podatkov.

Mentor:
doc. dr. Janez Demšar

Dekan:
prof. dr. Franc Solina

Somentor:
prof. dr. Juš Kocijan

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Dejan Petelin,

z vpisno številko 63030027,

sem avtor/-ica diplomskega dela z naslovom:

Sprotno učenje modelov na podlagi Gaussovih procesov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom
  doc. dr. Janez Demšar
  in somentorstvom
  prof. dr. Juš Kocijan
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 09.09.2009 Podpis avtorja/-ice:

Zahvala

Iskreno se zahvaljujem mentorju doc. dr. Janezu Demšarju in somentorju prof. dr. Jušu Kocijanu za pomoč in vzpodbudo pri izdelavi tega diplomskega dela. Prav tako bi se rad zahvalil tudi kolegom Odseka za sisteme in vodenje Instituta Jožef Stefan.

Kazalo

Seznam uporabljenih kratic in simbolov

$\mathbf{x}$ - vhodni vektor
$\mathbf{y}$ - izhodni vektor
$\mathcal{D} = (\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$ - množica podatkov
n - število učnih primerov
$C(\mathbf{x}, \mathbf{x}')$ - kovariančna funkcija
$\mathbf{K}$ - kovariančna matrika
$\boldsymbol{\Theta}$ - hiperparametri
$\mathcal{BV}$ - množica baznih vektorjev
$\boldsymbol{\alpha}, \boldsymbol{\Gamma}$ - parametri srednje vrednosti in kovariance

Povzetek

Modeliranje na podlagi Gaussovih procesov je razmeroma nova metoda modeliranja, ki je zaradi svojih dobrih lastnosti vedno bolj uporabljana, vendar pa zaradi časovne zahtevnosti, ki v raste s tretjo potenco glede na število učnih primerov, v osnovi ni primerna za sprotno učenje. Zato smo v tem diplomskem delu pregledali metode za zmanjšanje časovne odvisnosti in izbrali primerno za sprotno učenje. Izbrano metodo smo tudi podrobneje opisali ter preizkusili tako na simuliranem kot praktičnem problemu.

Izbrana metoda temelji na kombinaciji sprotnega učenja Bayesovega modela in zaporedni izgradnji podmnožice relevantih vhodnih primerov, ki opisujejo model.

Na podlagi primerov (slik in napak izmerjenih z različnimi merami) smo ugotovili, da je model, pridobljen s to metodo, odvisen od zaporedja vključevanja učnih primerov. To sicer ne preseneča, saj je vključitev vhodnega primera v množico odvisna od ocene prispevka glede na trenutni model. Bolj presentljiva pa je ugotovitev, da se metoda bolje obnese pri modeliranju eno-vhodnih problemov kot pri modeliranju več-vhodnih problemov.

Ključne besede:

Strojno učenje, Gaussovi procesi, model na podlagi Gaussovih procesov, sprotno učenje modelov

Abstract

Gaussian processes modeling is a relatively new modeling method which is due to its good features more and more applied. Unfortunately the computation time grows with third power as for size of training data set. Therefore this method is not convenient for online learning in principle. Variety of approximation methods and chose the convenient one for online learning were reviewed. The chosen method is described and demonstrated on a simulated and a real life problem.

The method is based on the combination of a Bayesian online algorithm together with the sequential construction of a relevant data subsample which specifies the model.

It was found on the basis of experiments (figures and errors measured with several measures) that the model obtained with this method depends on sequence of incorporated data. That is a result of the inclusion criterion which depends on the score of contribution regarding to the current model. Surprisingly, we also found the method is more effective on one-input problems than on multi-input problems.

Key words:

Machine learning, Gausian processes, Gaussian processes modeling, on-line learning

Poglavje 1

Uvod

1.1 Področje diplomskega dela

Strojno učenje je veja umetne inteligence, ki se uporablja za analizo podatkov in odkrivanje zakonitosti v podatkovnih bazah, za avtomatsko tvorjenje baz znanja za ekspertne sisteme, za razpoznavanje naravnega jezika, slik in govora, za gradnjo numeričnih ter kvalitetnih modelov itd. Osnovni princip strojnega učenja je avtomatsko opisovanje (modeliranje) pojavov iz podatkov. Rezultat učenja iz podatkov so lahko pravila, funkcije, relacije, sistemi enačb, verjetnostna porazdelitev ipd., ki so lahko predstavljene z različnimi formalizmi: odločitvenimi pravili, odličitvenimi drevesi, nevronskimi mrežami, jedri itd. Naučeni modeli poskušajo razlagati podatke, iz katerih so bili modeli tvorjeni, in se lahko uporabijo za odločanje pri opazovanju modeliranega procesa v bodočnosti (napovedovanje, diagnosticiranje, nadzor, preverjanje, simulacije itd.).

Razmeroma nova metoda modeliranja je modeliranje z Gaussovimi procesi. Model na osnovi Gaussovih procesov ali krajše GP model je neparametričen, kar pomeni, da neznanega sistema ne poskuša opisati s prilagajanjem parametrov (navadno velikega števila) baznih funkcij, ki sestavljajo model, kot je to značilno npr. za umetne nevronske mreže. Sestavljen je iz vhodno-izhodnih podatkov, ki opisujejo obnašanje opisovanega sistema in jih model uporablja za napovedovanje, in kovariančne funkcije, ki pove, v kakšni medsebojni odvisnosti so ti podatki oz. kakšne funkcije so verjetneje uporabljene pri opisu sistema. Izhod modela je verjetnostna porazdelitev v obliki Gaussove porazdelitve, pri čemer je srednja vrednost najbolj verjetna vrednost izhoda, varianco pa lahko interpretiramo kot zaupanje v to napoved. Izražanje zaupanja v napoved je lastnost, ki GP model najbolj loči od ostalih metod za modeliranje.

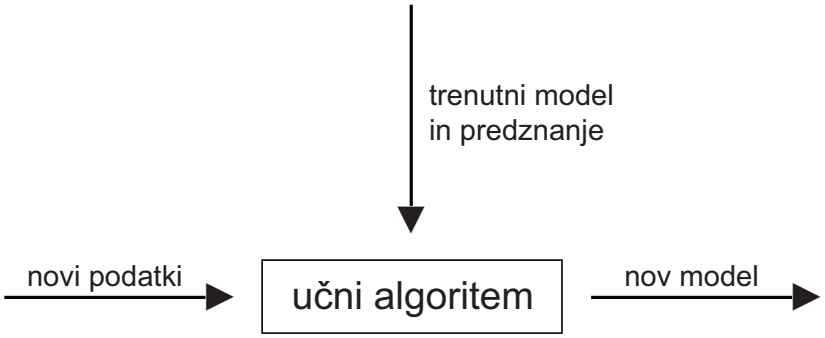
Poleg prej omenjene glavne lastnosti ima model GP še nekaj drugih dobrih lastnosti. Med njimi je zagotovo možnost vključevanja predznanja o sistemu, ki ga modeliramo. To dosežemo z izbiro ustrezne kovariančne funkcije oziroma kombinacijo le teh in izbiro regresorjev, ki najbolj vplivajo na sistem. Tudi uporaba GP modela je v primerjavi npr. z umetnimi nevronskimi mrežami, kjer je potrebno nastaviti ponavadi številne parametre, zelo enostavna, saj je potrebno določiti le majhno število parametrov. Zelo dobro se GP model obnese tudi pri primerih, kjer je malo podatkov, kar pomeni, da lahko opišemo področja, za katera je tipično pomanjkanje podatkov.

GP model je že dolgo znan na pordočju geostatistike, kjer je po Krigeju poznan pod imenom »kriging«. Kot orodje za reševanje regresijskih problemov ga je leta 1978 predstavil O'Hagan, popularnost v krogu ljudi, ki se ukvarjajo s strojnim učenjem, pa je v devetdesetih letih prejšnega stoletja pridobil najprej z deli Neila, ki je pokazal, kakšna je povezava med GP modelom in umetnimi nevronskimi mrežami, Rasmussena, ki je model umestil v Bayesov okvir, ter Gibsa in Williamsa. Več o razvoju GP modela najdemo npr. v [9].

GP model se lahko uporablja za reševanje klasifikacijskih problemov, pri katerih je izhod modela neka vrednost iz končne (ponavadi majhne) množice, in za reševanje regresijskih problemov, pri katerih je izhod modela zvezen. V tej diplomski nalogi se bomo omejili le na uporabo GP modela za regresijske probleme.

1.2 Motivacija

Kljub veliko dobrim lastnostim, ima model GP tudi nekaj omejitev, med katerimi je zagotovo računska zahtevnost, ki raste s tretjo potenco glede na velikost učne množice. To je resna omejitev za primere, ki imajo več kot nekaj tisoč podatkov, zato je bilo na tem področju opravljenih že veliko raziskav in s tem predlaganih kar nekaj pohitritev oziroma približkov. Veliko manj raziskano pa je področje sprotnega (inkrementalnega) učenja oziroma modeliranja z Gaussovimi procesi, ki zahteva zmožnost prilagajanje modela vsakemu novemu učnemu primeru. To pomeni, da z vsakim primerom poskuša poiskati najmanjšo potrebno spremembo trenutnega modela, tako da le-ta ustreza vsem do sedaj obravnavanim primerom. Pri realnih primerih, kjer je potrebno sprotno učenje oziroma modeliranje, so zahteve navadno še bolj stroge, saj je poleg prej omenjene zahteve, omejen tudi čas učenja, kajti učni primeri prihajajo v sistem v določenih intervalih. Zato se mora učenje izvesti v času krajšem od tega intervala, sicer bi se učni primeri nabirali v nedogled, česar pa si seveda



Slika 1.1: Sprotno (inkrementalno) učenje

ne želimo.

1.3 Namen in cilj

V diplomski nalogi smo želeli pregledati metode za pohitritev modeliranja z Gaussovimi procesi, ter med njimi izbrati tisto, ki je primerna za sprotno učenje oziroma modeliranje. Metod za sámo pohitritev je bilo razvitih že veliko, a vendar le malo takih, ki ustrezajo vsem zahtevam sprotnega učenja, zato smo jih želeli zbrati na enem mestu, eno izmed njih preizkusiti na praktičnem primeru z dovolj veliko učnimi podatki (več kot 10.000), ter rezultate primerjati z osnovnim algoritmom modeliranja z Gaussovimi procesi.

1.4 Struktura diplomskega dela

Po uvodu bomo v drugem poglavju najprej predstavili osnovne pojme GP modela, njegovo delovanje, uporabo v regresijskih modelih in prikazali delovanje na enostavnem statičnem primeru.

V tretjem poglavju se bomo posvetili metodam za pohitritev učenja modelov na podlagi Gaussovih procesov. Najprej jih bomo razdelili v dva sklopa glede na način delovanja. Prvi sklop so metode, ki pohitrijo množenje matrik in vektorjev, v drugem sklopu pa so metode, ki razredčijo oziroma aproksimirajo kovariančno matriko. V nadaljevanju se bomo posvetili le slednjim, saj prve zaradi še vedno eksponentne rasti časovne odvisnosti niso primerne za sprotno učenje. Opisali bomo osnovno idejo teh metod, jih razdelili v štiri sklope ter podali njihove značilnosti, prednosti in slabosti.

V četrtem poglavju bomo podrobneje predstavili idejo, zgradbo in delovanje metode, ki je primerna za sprotno učenje modelov na podlagi Gaussovih procesov. Metoda združuje ideji sprotnega učenja Bayesovega postopka in zaporednega grajenja podmnožice relevantnih učnih primerov, na katerih temelji model. Do te metode za sprotno učenje pridemo z uporabo parametrizacije, projekcijskih tehnik, rekurzije in aproksimacije.

V petem poglavju bomo ilustrirali uporabo opisane metode na dveh primerih. S prvim primerom smo želeli na enostaven način prikazati delovanje metode. Modelirali smo polinoma pete stopnje podanim s 76 točkami tako v zaporednem kot v naključnem razporedu točk. Z drugim primerom smo želeli pokazati učinkovitost metode, zato smo modelirali proces priprave plina, za katerega smo imeli na voljo 14.520 učnih in testih podatkov. Proces smo modelirali kot časovno vrsto ter kot model dinamičnega sistema.

V zaključku bomo povzeli poglavitne rezultate tega dela, še enkrat opisali najpomembnejše prispevke in dali nekaj napotkov za nadaljnje delo.

Poglavje 2

Modeliranje z Gaussovimi procesi

V tem poglavju je predstavljen model na osnovi Gaussovih procesov ali krajše GP model. V prvem razdelku so predstavljeni osnovni pojmi: kako deluje, kako ga učimo, kako ga uporabljamo za napovedovanje in kako lahko interpretiramo rezultate. Podrobneje so osnove modeliranja z modelom na podlagi Gaussovih procesov opisane v [21] ali začetnih poglavjih v [15], bolj podrobno razlago najdemo v [15, 14]. Na koncu poglavja je na enostavnem primeru predstavljena uporaba GP modela za reševanje statičnega regresijskega problema.

2.1 Osnovni pojmi in delovanje

Gaussovi procesi so naključni procesi. Naključni proces je posplošitev naključne spremenljivke (npr. vektor, skalar) na neki od neodvisnih spremenljivk odvisen prostor. Če je vrednost naključne spremenljivke v vsaki točki tega prostora porazdeljena po Gaussovi (normalni) porazdelitvi, takemu procesu pravimo *Gaussov proces* (GP) [4]. Drugače: če je vhod v proces vektor neodvisnih spremenljivk $\mathbf{x}$, je ta proces Gaussov, če je porazdelitev vrednosti funkcije $f(\mathbf{x})$ za vsak vhodni vektor $\mathbf{x}$ Gaussova.

Model na podlagi Gaussovih procesov (krajše GP model) je verjetnostni model [4]. Namesto za modeliranje bolj običajne omejitve na nek razred (parametriziranih) funkcij s tem načinom *a priori* dopuščamo opis neznanega sistema z neskončno množico funkcij. Pri tem dopuščamo večjo verjetnost funkcij, za katere menimo, da se pri opisu sistema bolj verjetno ponavljajo npr. gladke, stacionarne, periodične.

Vhod v GP model so posamezne vrednosti neodvisnih spremenljivk, zbrane

v vhodnem vektorju $\mathbf{x}$, medtem ko je izhod iz GP modela verjetnostna porazdelitev izhodne vrednosti $f(\mathbf{x})$ pri danem vhodnem vektorju,

Za poljubni nabor N vhodnih vektorjev $\mathbf{x}_i, i = 1, \ldots, N$, je GP določen z vektorjem srednjih vrednosti $\mathbf{m} = [m_1(\mathbf{x}_1) \ldots m_N(\mathbf{x}_N)]^T$ in kovariančno matriko $\mathbf{K}$,

$$\mathbf{K} = \begin{bmatrix} K_{11} & \ldots & K_{1N} \\ \vdots & \ddots & \vdots \\ K_{N1} & \ldots & K_{NN} \end{bmatrix} \tag{2.1}$$

kjer

$$m_i(\mathbf{x}_i) = \mathbb{E}[f(\mathbf{x}_i)] \tag{2.2}$$

in so elementi kovariančne matrike K_{ij}, običajno dobljeni z neko *kovariančno funkcijo* $C(\mathbf{x}_i, \mathbf{x}_j)$, določeni kot:

$$\mathrm{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = \mathbb{E}[(f(\mathbf{x}_i) - m(\mathbf{x}_i))\ (f(\mathbf{x}_j) - m(\mathbf{x}_j))], \tag{2.3}$$

kjer je $\mathbb{E}[x] = \int_{-\infty}^{\infty} xp(x)dx$ matematično upanje (povprečna vrednost) naključne spremenljivke x z verjetnostno porazdelitvijo $p(x)$.

Če je porazdelitev neke množice spremenljivk Gaussova, je Gaussova tudi porazdelitev katerekoli naključno izbrane podmnožice elementov te množice, kar imenujemo zahteva po konsistenci (angl. *consistency requirement*). To pomembno lastnost za delovanje GP modela vedno dosežemo, če so elementi kovariančne matrike $\mathbf{K}$ GP-ja dobljeni s kovariančno funkcijo.

Kovariančna funkcija

Vrednost kovariančne funkcije $\mathbf{K}_0(\mathbf{x}_i, \mathbf{x}_j)$ izraža medsebojno odvisnost vrednosti izhodov $f(\mathbf{x}_i)$ in $f(\mathbf{x}_j)$ na podlagi vrednosti vhodnih vektorjev $\mathbf{x}_i$ in $\mathbf{x}_j$. Kovariančna funkcija je lahko različnih oblik, potrebno je le, da za poljubni nabor N vhodnih vektorjev $\mathbf{x}_i, i = 1, \ldots, N$, tvori pozitivno definitno kovariančno matriko $\mathbf{K}$. Kovariančne funkcije so lahko stacionarne, nestacionarne, periodične itd., med seboj pa jih lahko tudi seštevamo in množimo. Kovariančne funkcije, ki določa obliko neznane funkcije $f(\mathbf{x})$, navadno ne poznamo vnaprej, lahko pa iz znanja o splošnih lastnostih funkcije $f(\mathbf{x})$ sklepamo o njeni obliki. Podrobneje so opisane v [18].

Najpogosteje, zlasti kadar o lastnostih funkcije $f(\mathbf{x})$ ne vemo dovolj, se uporablja Gaussova kovariančna funkcija [18], ki izraža dve pogosti lastnosti procesov:

- gladkost, ki pove, da se bo izhod procesa z majhno spremembo vhoda razmeroma malo spremenil (medsebojna odvisnost dveh vrednosti izhodov je večja, če ustrezni vrednosti vhodov ležita blizu skupaj), in
- stacionarnost, pri kateri je kovarianca med dvema vhodnima vektorjema odvisna od njune medsebojne razdalje in ne tudi od njune absolutne lege v prostoru.

Pri Gaussovi kovariančni matriki je kovarianca med dvema izhodoma $y_i = f(\mathbf{x}_i)$ in $y_j = f(\mathbf{x}_j)$:

$$K_{ij} = C(\mathbf{x}_i, \mathbf{x}_j) = v \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_i^d - x_j^d)^2\right] \tag{2.4}$$

D je dimenzija vhodnega prostora in določa dolžino vhodnega vektorja $\mathbf{x}$. Parametri v in $w_d, d = 1, \ldots, D$, so poljubno določljivi parametri kovariančne funkcije. Imenujemo jih *hiperparametri*[1] [14, 11]; s tem poudarimo, da so to parametri sicer neparametričnega modela[2], ki določajo obliko neznane funkcije $f(\mathbf{x})$. Parameter v govori o varianci izhoda , parametri w_d pa odražajo pomembnost posamezne komponente vhodnega vektorja; večji je parameter w_d, vplivnejša je sprememba komponente vektorja x^d na vrednost izhoda. Da dana kovariančna funkcija tvori pozitivno definitno kovariančno matriko, morajo biti vsi parametri Gaussove kovariančne funkcije večji od nič.

Modeliranje

Najlažje predstavimo delovanje GP modela na primeru. Vzemimo, da bi radi opisali sistem:

$$y = f(\mathbf{x}) + v, \tag{2.5}$$

kjer je v beli Gaussov šum z varianco $v_0, v \sim \mathcal{N}(0, v_0)$. Na podlagi N-tih vhodno-izhodnih vzorcev, tj. parov vektorjev $(\mathbf{x}_i, y_i)$, zbranih v množici $\mathcal{D} = \{\mathbf{X}, y\}$, želimo določiti neznano vrednost izhoda y^* pri vrednostih vhodnega vektorja $\mathbf{x}^*$. V nadaljevanju v kontekstu GP modela $N \times D$ matriko

[1] Neal [14] je pokazal, da je vnaprejšnja nevronska mreža z enim skritim nivojem z neskončnim številom nevronov in pri določenih porazdelitvah parametrov nevronske mreže enaka GP modelu. Hiperparametri določajo distribucijo vrednosti (sicer neskončnega števila) parametrov te nevronske mreže

[2] Model je neparametričen, saj za napovedovanje poleg hiperparametrov in kovariančne funkcije potrebujemo še informacijo o obnašanju sistema v obliki vhodno/izhodnih podatkov, uporabljenih pri modeliranju

$\mathbf{X}$ in $N \times 1$ vektor $\mathbf{y}$ označimo kot *učno množico*, saj jih uporabljamo za učenje (angl. *training*) GP modela. Posamezni vhodno/izhodni par $(\mathbf{x}_i, y_i)$ iz te množice imenujemo tudi učni vektor oz. učna točka. Par $(\mathbf{x}^*, y^*)$ označimo kot preizkusno oz. testno množico ali tudi kot *testni vhod/izhod*.

Učni izhodi $y_i, i = 1, \ldots, N$ predstavljajo vrednosti naključnih spremenljivk, izhajajočih iz Gaussovega procesa. Predpostavimo, da je izhod sistema gladek in da je sistem stacionaren ter za tvorjenje kovariančne matrike $\mathbf{K}$ uporabimo Gaussovo kovariančno funkcijo (2.4) z na začetku neznanimi parametri.

Dobimo: $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, kjer so elementi kovariančne matrike $K_{ij} = \Sigma_{ij} + v_0\delta_{ij}$. Σ_{ij} so elementi kovariančne matrike, dobljeni s kovariančno funkcijo (2.4), $v_0\delta_{ij}$ pa opisuje vpliv šuma na izhodu porocesa, kjer je δ_{ij} Kroneckerjev operator. Ker smo predpostavili beli šum, so njegove vrednosti korelirane le same s seboj.

Z uporabo podatkov $\mathcal{D} = \mathbf{X}, \mathbf{y}$, ki jih imamo na voljo, bi radi določili neznano funkcijo $f(\mathbf{x})$ iz enačbe (2.5). Funkcijo modeliramo z uporabo Bayesovega pristopa [10]:

$$p(f(\mathbf{x})|\mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}|f(\mathbf{x}), \mathbf{X})\, p(f(\mathbf{x}))}{p(\mathbf{y}|\mathbf{X})}\,. \tag{2.6}$$

Prvi izraz v števcu enačbe (2.6) $p(\mathbf{y}|f(\mathbf{x}), \mathbf{X})$ predstavlja verjetnost učnih izhodov glede na funkcijo $f(\mathbf{x})$ (in učne vhode $\mathbf{X}$) in je v regresijskih problemih navadno predpostavljena Gaussova [10]. Drugi izraz v števcu predstavlja apriorno verjetnost posameznih funkcij, ki sestavljajo model. Ideja modela na osnovi Gaussovih procesov je, da funkcije $f(\mathbf{x})$ ne parametriziramo, ampak določimo apriorno verjetnosti direktno v funkcijskem prostoru [10].

Ker je (zaenkrat še neznani) izhod $y*$ udejanjenje istega procesa kot učni izhod $\mathbf{y}$, lahko zapišemo [2]: $\mathbf{y}_{N+1} = \left[\begin{smallmatrix}\mathbf{y}\\ y^*\end{smallmatrix}\right] \sim \mathcal{N}(0, \mathbf{K}_{N+1})$. Skupno kovariančno matriko $\mathbf{K}_{N+1}$ vektorja $\mathbf{y}_{N+1}$ lahko razdelimo:

$$\mathbf{K}_{N+1} = \begin{bmatrix} \begin{bmatrix}\mathbf{K}\end{bmatrix} & \begin{bmatrix}\mathbf{k}(\mathbf{x}^*)\end{bmatrix} \\ \begin{bmatrix}\mathbf{k}(\mathbf{x}^*)^T\end{bmatrix} & \begin{bmatrix}k(\mathbf{x}^*)\end{bmatrix} \end{bmatrix}. \tag{2.7}$$

Matrika $\mathbf{K}$ je kovariančna matrika učnih podatkov, $\mathbf{k}(\mathbf{x}^*)$ je vektor kovarianc med učnimi izhodi in testnim izhodom, $k(\mathbf{x}^*)$ pa avtokovarianca testnega izhoda.

Po Bayesovem načinu lahko verjetnostno porazdelitev vrednosti izhoda y^* razdelimo na dva dela: na del, ki določa verjetnost učnih izhodov glede na

učne vhode (angl. *marginal part*), $p(\mathbf{y}|\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$, in na pogojni del (angl. *conditional part*), ki glede na prvi del in vhod $\mathbf{X}^*$ napoveduje verjetnostno porazdelitev izhoda y^*. Formalno zapisano je izračun porazdelitve izhodne verjetnosti odziva y^* [2, 9]:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \int p(y^*|\mathbf{x}^*, \boldsymbol{\Theta}, \mathbf{y}, \mathbf{X}) p(\boldsymbol{\Theta}|\mathbf{y}, \mathbf{X}) d\boldsymbol{\Theta}. \tag{2.8}$$

Običajno je ta integral analitično neizračunljiv, imamo pa na voljo več alternativ [9, 15, 1]. Osnovna, bolj pogosta, je aproksimacija integrala z uporabo najbolj verjetnih vrednosti neznanih hiperparametrov $\boldsymbol{\Theta}_{\mathbf{MP}}$:

$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) \approx p(y^*|\mathbf{x}^*, \boldsymbol{\Theta}_{\mathbf{MP}}, \mathbf{y}, \mathbf{X}). \tag{2.9}$$

Uporabimo tiste vrednosti hiperparametrov $\boldsymbol{\Theta}_{\mathbf{MP}}$, pri katerih je verjetnost učnih izhodov $\mathbf{y}$ glede na vrednosti učnih vhodov $\mathbf{X}$ in kovariančno funkcijo $C(.,.)$ največja. Dobimo jih z metodo največje podobnosti (angl. *maximum likelihood method* - ML). Da se izognemo optimizaciji z omejitvami, za optimizacijo uporabimo logaritem porazdelitve učnih podatkov (angl. *log-marginal likelihood*):

$$\mathcal{L}(\boldsymbol{\Theta}) = \log(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\Theta})) = -\frac{1}{2}\log(|\mathbf{K}|) - \frac{1}{2}\mathbf{y}^{\mathbf{T}}\mathbf{K}^{-\mathbf{1}}\mathbf{y} - \frac{N}{2}\log(2\pi), \tag{2.10}$$

kjer je $\boldsymbol{\Theta} = [w_1 \ldots w_D \, v \, v_0]^T$ vektor parametrov in $\mathbf{K}$ kovariančna matrika učnih podatkov $\mathcal{D}$. Če je optimizacija izvedena z metodo konjugiranih gradientov (ali katero drugo gradientno metodo), je potreben še izračun odvodov po vseh hiperparametrih:

$$\frac{\partial \mathcal{L}(\boldsymbol{\Theta})}{\partial \Theta_i} = -\frac{1}{2}\mathrm{sled}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\right) + \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \Theta_i}\mathbf{K}^{-1}\mathbf{y} \tag{2.11}$$

Ob vsakem koraku optimizacije je potrebno izračunati inverz kovariančne matrike $\mathbf{K}^{-1}$, kar je računsko zahtevno za velike N. Temu pa se lahko izognemo s sprotno aproksimacijo oziroma učenjem [12], ki ga bomo opisali v poglavju 4.

Naj kot možnost za aproksimacijo integrala (2.8) omenimo še numerično integracijo nad celotno porazdelitvijo hiperparametrov (MCMC metode, [9]), dobljeno z optimizacijo verjetnosti učnih podatkov (2.10). Ta je primerna v primeru kadar imamo veliko hiperparametrov in jim težko določimo začetne vrednosti.

Napovedovanje

Skupna porazdelitev $p(\mathbf{y}_{N+1})$ je Gaussova; torej je Gaussova tudi pogojna porazdelitev $p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \frac{p(\mathbf{y}_{N+1})}{p(\mathbf{y}|\mathbf{X})}$. Po poenostavitvi [9, 21] kot napovedan izhod sistema (2.5) dobimo Gaussovo porazdelitev:

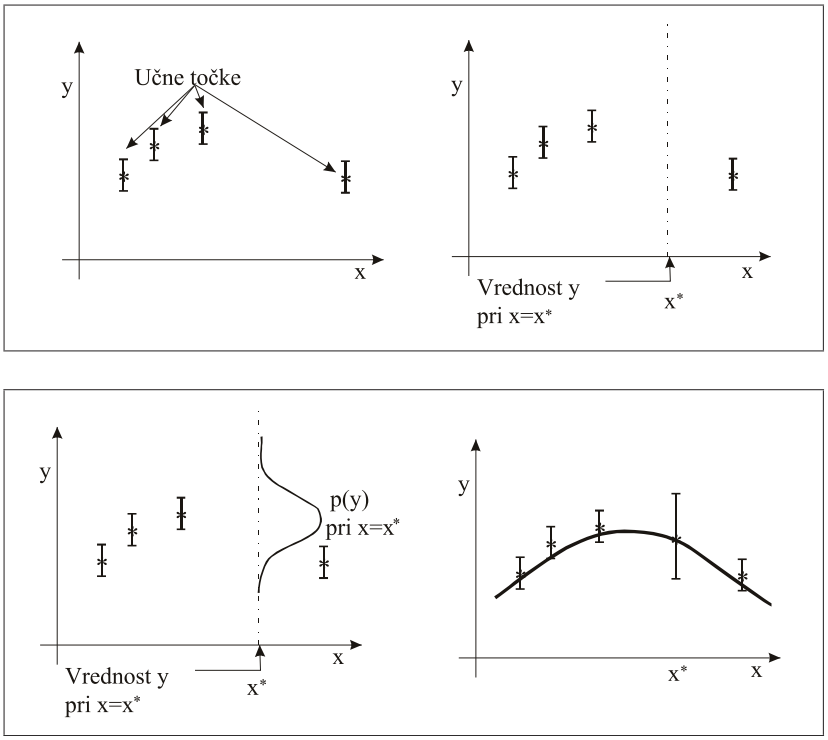
$$p(y^*|\mathbf{x}^*, \mathbf{y}, \mathbf{X}) = \mathcal{N}(\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*)) \tag{2.12}$$

s srednjo vrednostjo $\mu(\mathbf{x}^*)$ in varianco $\sigma^2(\mathbf{x}^*)$:

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{y} \tag{2.13}$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*), \tag{2.14}$$

kjer je $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x_1}, \mathbf{x}^*) \dots C(x_N, \mathbf{x}^*)]$ že omenjeni $N \times 1$ kovariančni vektor med testnim izhodom in učnimi izodi ter $k(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ avtokovarianca testnega izhoda. Ilustracijo opisanega prikazuje slika 2.1.



Slika 2.1: Princip modeliranja z Gaussovimi procesi

Interpretacija

GP model je sestavljen iz dveh delov:

- iz parov vhodno/izhodnih učnih podatkov (točk) $\mathcal{D}$, ki predstavljajo obnašanje neznanega sistema, in
- kovariančne funkcije $C(.,.)$ z znanimi oz. optimiziranimi hipermarametri $\mathbf{\Theta}$, ki povedo, v kakšnem razmerju so podatki $\mathcal{D}$.

Ker GP model vsebuje informacijo o neznani funkciji v obliki učnih vhodov in izhodov tudi po učenju, je model neparametričen. Hiperparametri namreč prek kovariančne funkcije samo povedo, kako se učna informacija uporabi za napovedovanje, ni pa v njih spravljena informacija o opisovani funkciji/sistemu.

Na vektor $\mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}$ v izrazu za srednjo vrednost napovedanega izhoda (2.13) lahko gledamo kot na vektor uteži, ki določa utežitev posameznih učnih izhodov y_i v $\mathbf{y}$ glede na razdaljo med učnimi in testnim vhodnim vektorjem. Ta linearna kombinacija učnih izhodov (angl. *linear predictor*) se lahko razume kot glajenje v GP modelu vsebovane informacije o neznanem sistemu (učni podatki). Še drugače si lahko napoved $\mu(\mathbf{x}^*)$ predstavljamo kot linearno kombinacijo N jedrnih (angl. *kernel*) funkcij, usrediščenih v učnih točkah; $y^* = \sum_{i=1}^{N} \alpha_i C(\mathbf{x}^*, \mathbf{x}_i)$. Izhod iz sistema je en vzorec iz dobljene normalne porazdelitve (2.12).

Majhna varianca $\mu(\mathbf{x}^*)$ napovedane porazdelitve izhoda pomeni večje *zaupanje* v napoved. Če si ogledamo izraz za varianco, vidimo da je sestavljen iz dveh delov [15]. Od prvega dela $k(\mathbf{x}^*)$, ki predstavla apriorno varianco GP, je odštet izraz $\mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*)$. Ta predstavlja zmanjšanje apriorne variance GP pri $\mathbf{x}^*$ zaradi učnih podatkov in se veča z večjo kovarianco med učnimi in testnim vhodom. Preprosteje: bolj ko je testni vhod podoben že znanim (učnim) vhodom, večje je zaupanje GP modela v točnost napovedi. Prav varianca, odvisna tudi od lege testnega vhoda glede na učne, je ena izmed glavnih prednosti GP modela pred drugačnimi modeli.

Vrednotenje

Vrednotenje, ki pove kako dober je dobljen model, je zelo pomemben korak pri modeliranju. Z vrednotenjem preverimo ujemanje matematičnega modela in obravnavanega sistema.

Kvaliteto napovedi modela lahko merimo na več načinov, najbolj pogoste mere pa so:

- srednja kvadratična napaka (angl. mean squared error - MSE),

- srednja absolutna napaka (angl. mean absolute error - MAE),
- logaritem gostote napake (angl. minus log-predicted density error - LPD),
- povprečna relativna kvadratična napaka (angl. mean relative square error - MRSE),
- logaritem verjetnostne porazdelitve učne množice (angl. minus log-likelihood).

Mera MSE predstavlja povprečni kvadrat razlike med napovedano vrednostjo $\hat{f}(i)$ in želeno vrednostjo $f(i)$:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(i) - \hat{f}(i))^2 \tag{2.15}$$

Druga pogosto uporabljena mera je MAE, ki predstavlja povprečno absolutno razliko med napovedano vrednostjo $\hat{f}(i)$ in želeno vrednostjo $f(i)$:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(i) - \hat{f}(i)| \tag{2.16}$$

Mera LPD poleg razlike med napovedano vrednostjo $\hat{f}(i)$ in želeno vrednostjo $f(i)$ upošteva tudi varianco napovedi σ. Tako mera LPD podaja informacijo o povprečni kvadratni napaki, normirani z vrednostjo variance napovedi. Uporablja se predvsem pri Bayesovem modeliranju, na katerem temelji tudi modeliranje na podlagi Gaussovih procesov. Podana je z izrazom

$$LPD = \frac{1}{2n}\sum_{i=1}^{n}\left(\log(2\pi) + \log(\sigma) + \frac{(f(i) - \hat{f}(i))^2}{\sigma}\right) \tag{2.17}$$

MRSE je mera, katere vrednost je neodvisna od vrednosti podatkov in je definirana z

$$MRSE = \sqrt{\frac{\sum_{i=1}^{n}(f(i) - \hat{f}(i))^2}{\sum_{i=1}^{n} f(i)^2}} \tag{2.18}$$

kjer je $\hat{f}(i)$ napovedana vrednost in $f(i)$ želena vrednost.

Kadar imamo na voljo malo vhodnih podatkov in potrebujemo za učenje vse razpoložljive primere, kljub temu pa želimo oceniti uspešnost modela, uporabimo postopek navzkrižnega vrednotenja (angl. cross-validation). Najzanesljivejša metoda tega postopka je metoda izločevanje enega (angl. leave-one-out

- LOO). Vsak primer izločimo iz učne množice in iz vseh preostalih primerov zgradimo model, ki ga zatem uporabimo za napoved izločenega primera. To ponovimo za vse primere in uspešnost modela, ki smo ga zgradili iz vseh učnih primerov, ocenimo kot povprečno uspešnost vseh zgrajenih modelov na ustreznem (izločenem) primeru. Ker je ta metoda velikokrat časovno nesprejemljiva, saj moramo zgraditi $N + 1$ modelov namesto ene same, jo pogosto posplošimo na »izloči N/K primerov«, ki ji pravimo tudi K-kratno navzkrižno vrednotenje (angl. K-fold cross-validation). Število K določa število modelov, ki jih moramo zgraditi. Na začetku množico razpoložljivih primerov razdelimo na K približno enako številčnih množic. Nato za vsako podmnožico zgradimo model, tako da za učenje uporabimo unijo presotalih podmnožic, in ga uporabimo za reševanje primerov iz dane podmnožice. Uspešnost končnega modela ocenimo kot povprečno uspešnost vseh K modelov na celotni množici testnih primerov. Bolj zanesljiva različica te metode je sorazmerno navzkrižno vrednotenje (angl. stratified cross-validation). To je navzkrižno vrednotenje, kjer ohranjamo približno enako distribucijo razredov v vseh podmnožicah [5].

2.2 Primer uporabe GP modela na statičnem regresijskem modelu

Ilustrirajmo uporabo GP modela na primeru. Želimo identificirati nelinearno funkcijo $f(x)$, odvisno od neodvisne spremenljivke x:

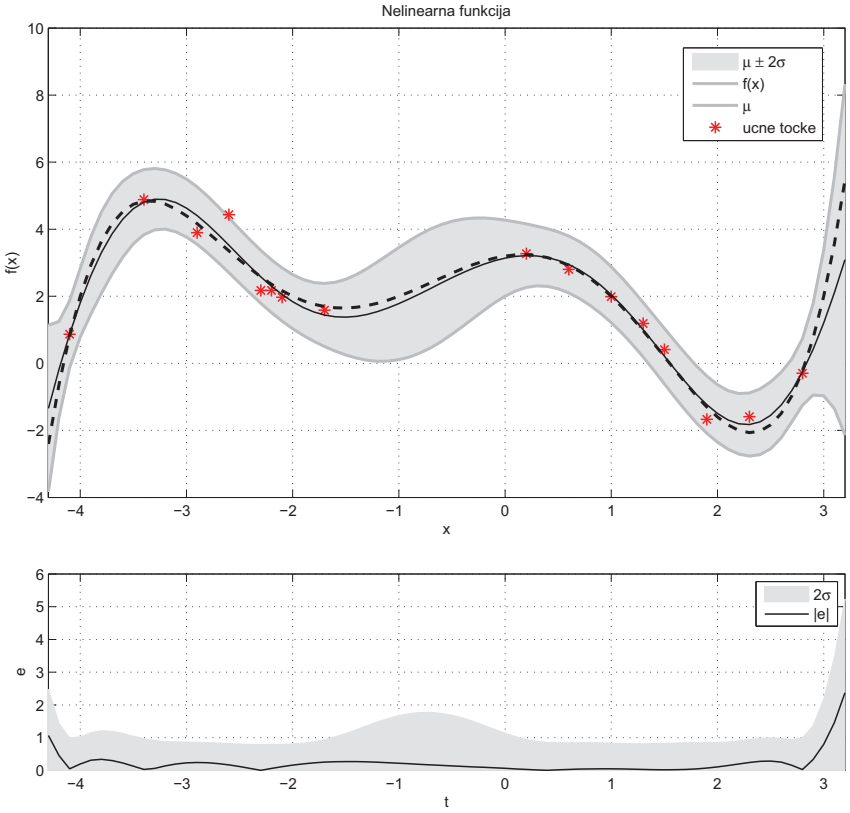
$$f(x) = \frac{1}{20}(x+4)(x+1)(x+1)(x-1)(x-3) + 2 + v \tag{2.19}$$

na intervalu $x \in [-4.3, 3.2]$. Varianca Gaussovega šuma v na izhodu je $\mu^2 = 0.001$. Nelinearna funkcija je predstavljena z osmimi neenakomerno porazdeljenimi učnimi pari (točkami), ki predstavljajo vhodno/izhodno relacijo $x/f(x)$. Funkcijo in učne točke lahko vidimo na sliki 2.2.

Za identifikacijo izberemo Gaussovo kovariančno funkcijo (2.4), in to zaradi samo enega vhoda poenostavljeno v:

$$C(x_i, x_j) = v_1 \exp\Big[-\frac{1}{2}w(x_i - x_j)^2\Big] + v_0\delta_{ij}. \tag{2.20}$$

Z optimizacijo določimo tri hiperparametre, ki dobijo vrednosti $v_1 = 0.7$, $w = 7.3$ ter $v_0 = 0.0014$. Rezultati modeliranja so prikazani na sliki 2.2. Lahko opazimo, da model slabo opisuje neznano funkcijo na področju, ki ni opisano z učnimi točkami $x > 2.8$, prav tako je napoved slabša na redkeje (to je z malo



Slika 2.2: Izhod modela (polna krivulja) z negotovostjo (siva krivulja) in nelinearne funkcije (zvezdice)

točkami) opisanih področjih $-2 < x < 0.7$. Dobra lastnost GP modela je, da nas na slabše opisano področje opozori povečana varianca (negotovost) na sliki 2.1, kar je vidno predvsem pri $x > 2.8$. Manj opazna (zaradi manjšega šuma) je druga lastnost GP modela, to je glajenje vsebovane učne informacije, pri katerem model vsebovane (učne) pošumljene vzorce zgladi za napoved novega izhoda.

Poglavje 3

Aproksimacija GP modelov

Kot je razvidno iz razdelka 2.1 časovna zahtevnost direktne implementacije regresije z Gaussovimi procesi raste s tretjo potenco glede na število učnih primerov - $O(n^3)$, saj je potrebno izračunati inverz kovariančne matrike $\alpha = \mathbf{K}^{-1}\mathbf{y}$ (2.13) oziroma rešiti linearni sistem $\mathbf{K}\alpha = \mathbf{y}$ za α. To pa predstavlja veliko težavo za uporabo pri sistemih, ki obsegajo veliko učnih primerov (več kot nekaj tisoč). Zato je bilo postopkom za zmanjšanje časovne zahtevnosti posvečenih že veliko raziskav in s tem razvitih veliko metod, ki se v splošnem delijo na:

- metode, ki uporabljajo hitro množenje matrik in vektorjev (angl. matrix-vector multiplication - MVM), s čimer aproksimirajo sámo implementacijo regresije z Gaussovimi procesi,
- razredčevalne metode, ki aproksimirajo kovariančno matriko.

Kljub izboljšavam direktne implementacije z MVM metodami, ki zmanjšajo časovno zahtevnost tudi za en red - $O(n^2)$ [15], te metode niso primerne za sprotno učenje, saj je njihova časovna zahtevnost vedno večja z vsakim korakom učenja. Zato so uporabne le za sisteme z malo primeri ali pa v kombinaciji z razpršitvenimi metodami. Ideja slednjih je ustrezno zmanjšati rang kovariančne matrike (število linearno neodvisnih vrstic), a kljub temu obdržati čim več informacij vsebovanih v polni učni množici. Ker lahko s temi metodami ohranjamo dovolj majhen konstanten rang kovariančne matrike in s tem zadostimo predpostavki sprotnega učenja, ki zahteva konstanten čas obdelave novega primera, se bomo v nadaljevanju posvetili le tem metodam. Večino metod smo zajeli z naslednjimi štirimi skupinami: podmnožica podatkov, podmnožica regresorjev, Nyströmova aproksimacija in aproksimacija s projekcijo procesa.

3.1 Zredčena matrika

Pri gradnji razpršene kovariančne matrike je prvi korak izbira podmnožice primerov. Izbira te takoimenovane aktivne podmnožice podatkov je skupna vsem metodam z redčenjem, pri katerih se izbrane spremenljivke upoštevajo pri GP modeliranju, ostale spremenljivke pa se aproksimirajo z računsko manj zahtevno metodo.

Izbrana podmnožica je velikosti $m < n$, kjer je n velikost celotne učne množice, in je označena kot $\mathcal{I}$ iz angleškega izraza za vsebovan (angl. included) [15], podmnožica z ostalimi primeri je velikosti m in označena kot $\mathcal{R}$ iz angleškega izraza preostali (angl. remaining). Če predpostavimo, da so učni primeri urejeni tako, da je podmnožica $\mathcal{I}$ na začetku, potem kovariančno matriko $\mathbf{K}$ lahko razdelimo na:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{mm} & \mathbf{K}_{m(n-m)} \\ \\ \mathbf{K}_{(n-m)m} & \mathbf{K}_{(n-m)(n-m)} \end{bmatrix} \tag{3.1}$$

pri čemer zgornji blok velikosti $m \times n$ lahko označimo kot $\mathbf{K}_{mn}$.

3.2 Podmnožica podatkov (SD)

Najenostavnejša med razpršitvenimi metodami je aproksimacija s podmnožico podatkov, pri kateri je aktivna podmnožica velikosti m izbrana iz celotne učne množice velikosti n, sama implementacija modeliranja pa ostane nespremenjena. S tem se sicer ohrani časovna zahtevnost, ki raste s tretjo potenco glede na število primerov, a vendar se število primerov samih zmanjša. Tako je časovna zahtevnost odvisna le od m - $O(m^3)$, kjer je $m < n$.

Uspešnost te metode je zelo odvisna od izbire podmnožice primerov, saj se ostali del primerov sploh ne upošteva za razliko od ostalih bolj naprednih metod, kjer se ostale primere upošteva oziroma aproksimira. Vendar lahko s pazljivo izbrano aktivno podmnožico ali z uporabo požrešnih metod (angl. Greedy methods), t.j. z uporabo ustreznega kriterija vključevanja, dosežemo zelo dober približek »polnemu« GP modelu.

3.3 Podmnožica regresorjev (SR)

Metoda SR izkorišča enakost med GP modelom in (končno-dimenzijskim) posplošenim linearnim modelom [20, 17]. Zato model SR s končnim številom parametrov vsebuje določeno predznanje o vrednostih uteži.

Za vsak vhod (primer) $\mathbf{x}^*$ obstaja funkcijska vrednost f^* določena z:

$$f(\mathbf{x}^*) = \sum_{i=1}^{n} \alpha_i \mathbf{K}_0(\mathbf{x}^*, \mathbf{X}_i), \text{kjer je } \alpha \sim \mathcal{N}(0, \mathbf{K}^{-1}) \tag{3.2}$$

Model enostavno aproksimiramo tako, da upoštevamo le podmnožico regresorjev:

$$f_{SR} = \sum_{i=1}^{m} \alpha_i \mathbf{K}_0(\mathbf{x}^*, \mathbf{x}_i), \text{kjer je } \alpha_m \sim \mathcal{N}(0, \mathbf{K}^*_{mm}) \tag{3.3}$$

Na podlagi tega lahko oblikujemo predikcijo porazdelitve enako, kot je opisano v interpretaciji uteženega GP modela [15] in s tem srednjo vrednost in varianco:

$$\overline{f}_{SR}(\mathbf{x}^*) = \mathbf{k}_m(\mathbf{x}^*)^T(\mathbf{K}_{mn}\mathbf{K}_{nm} + \sigma_n^2\mathbf{K}_{mm})^{-1}\mathbf{K}_{mn}\mathbf{y} \tag{3.4}$$

$$\mathbb{E}[f_{SR}(\mathbf{x}^*)] = \sigma_n^2\mathbf{k}_m(\mathbf{x}^*)^T(\mathbf{K}_{mn}\mathbf{K}_{nm} + \sigma_n^2\mathbf{K}_{mm})^{-1}\mathbf{k}_m(\mathbf{x}^*) \tag{3.5}$$

Iz zgornjih enačb je razvidno, da metoda SR, za razliko od metode SD, pri aproksimaciji upošteva vseh n primerov učne množice. Vendar pa je njena glavna pomankljivost, da temelji na modelu, ki je linearen v parametrih, zaradi česar GP model postane degeneriran in s tem omejen pri raznolikosti možnih funkcij, ki so dovolj verjetne za opis procesa, ki ga modeliramo.

Glavna slabost degeneracije je lahko zelo slab oziroma nezaupljiv rezultat napovedi. Za kovariančne funkcije velja, da se z večanjem razdalje med različnimi vhodi veča tudi varianca in s tem tudi nezaupanje v napoved. Na žalost pa zaradi omejitev metode SR pri aproksimaciji funkcij, v nekaterih primerih, kjer je razdalja med vhodi velika, napoved nima variance oziroma je zelo blizu nič, kar pa je v nasprotju s pričakovanim. V splošnem je metoda SR zelo uporaben postopek za aproksimiranje srednje vrednosti, vendar pa je varianca velikokrat določena preveč optimistično ali celo nesmiselno.

Časovna zahtevnost modeliranja metode SR je $O(m^2n)$, napovedovanje srednje vrednosti in variance pa $O(m)$ oziroma $O(m^2)$.

3.4 Nyströmova aproksimacija

Nyströmova metoda vključuje analizo in aproksimacijo lastnih funkcij in lastnih vektorjev jedra. Podrobneje je opisana v [13], za uporabo aproksimiranja pri regresiji z Gaussovimi procesi pa je bila predlagana v [23]. Ta metoda na podlagi kovariančne matrike $\mathbf{K}$ aproksimira novo zredčeno matriko $\tilde{\mathbf{K}}$, ki jo

potem lahko uporabimo pri napovedovanju. Če je izbrano število lastnih vrednosti oziroma vektorjev vključenih v aproksimacijo enako velikosti aktivne podmnožice $\mathcal{I}$, potem lahko Nyströmovo aproksimacijo kovariančne matrike $\mathbf{K}$ zapišemo kot:

$$\tilde{\mathbf{K}} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn} \tag{3.6}$$

Pri napovedovanju približek $\tilde{\mathbf{K}}$ nadomesti kovariančno matriko $\mathbf{K}$, vendar pa kovariančna funkcija $\mathbf{k}$ ni nadomeščena s funkcijo $\tilde{\mathbf{k}}$. To pa lahko povzroči tudi napake pri napovedovanju (negativna varianca).

Tako kot pri metodi SR je časovna zahtevnost modeliranja enaka $O(m^2n)$ in $O(n)$ za napoved testnega primera oziroma $O(mn)$ za napoved variance. Eksperimenti [24] so pokazali, da metoda SR in Nyströmova metoda pri velikem m dosežeta približno enake rezultate, pri majhnem m pa je Nyströmova metoda precej slabša.

3.5 Aproksimacija s projekcijo procesa (PP)

Metoda SR ima slabo lastnost, da je osnovana na degeneriranem modelu (3.2), česar posledica je slaba napoved variance. To slabost odpravlja metoda aproksimacije s projekcijo procesa, ki je nedegeneriran model in upošteva vseh n vhodnih primerov. Metoda je bila poimenovana tako, ker predstavlja le $m < n$ funkcijskih vrednosti, vendar pa pri modeliranju upošteva vseh n vhodnih primerov, tako da projecira m primerov na n dimenzij.

Srednja vrednost je določena z

$$\overline{f}_{PP}(\mathbf{x}^*) = \mathbf{k}_m(\mathbf{x}^*)^T(\mathbf{K}_{mn}\mathbf{K}_{nm} + \sigma_n^2\mathbf{K}_{mm})^{-1}\mathbf{K}_{mn}\mathbf{y} \tag{3.7}$$

iz česar je razvidno, da je identična kot pri SR. Varianca, kot smo že omenili, pa je različna. Določena je z

$$\begin{aligned}\mathbb{E}[f_{PP}(\mathbf{x}^*)] &= \mathbf{k}_m(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{k}_m(\mathbf{x}^*)^T\mathbf{K}_{mm}^{-1}\mathbf{k}_m(\mathbf{x}^*) + \\ &\quad +\sigma_n^2\mathbf{k}_m(\mathbf{x}^*)^T(\mathbf{K}_{mn}\mathbf{K}_{nm} + \sigma_n^2\mathbf{K}_{mm})^{-1}\mathbf{k}_m(\mathbf{x}^*)\end{aligned} \tag{3.8}$$

Opazimo, da je varianca enaka vsoti variance SR modela in $\mathbf{k}_m(\mathbf{x}^*,\mathbf{x}^*) - \mathbf{k}_m(\mathbf{x}^*)^T\mathbf{K}_{mm}^{-1}\mathbf{k}_m(\mathbf{x}^*)$. Zato varianca iz enačbe (3.8) ni nikoli manjša kot varianca pri SR modelu in je blizu $\mathbf{k}(\mathbf{x}_*,\mathbf{x}_*)$ kadar je $\mathbf{x}_*$ daleč od primerov (točk) v aktivni množici $\mathcal{I}$.

Tako kot pri metodi SR je časovna zahtevnost modeliranja $O(m^2n)$, napoved srednje vrednosti in variance za nov testni primer pa je $O(m)$ oziroma $O(m^2)$.

3.6 Izbira podmnožice

Ker je le aktivna podmnožica popolnoma obravnavana v zredčenih modelih, je izbira primerov, ki so vključeni v to podmnožico, ključnega pomena za uspešnost aproksimacije. Ena izmed možnosti je gradnja te podmnožice z ročno izbiro primerov na podlagi predznanja, ki ga imamo o značilnostih obravnavanega sistema. Vendar pa je to izredno težavno pri sistemih, o katerih nimamo dovolj predznanja ter pri zelo kompleksnih večdimenzionalnih in dinamičnih sistemih. V tem primeru pa je lahko celo primerna tudi strategija z naključno izbiro primerov.

Predstavljenih pa je bilo že kar nekaj bolj naprednih oziroma sistematičnih postopkov izbire aktivne podmnožice primerov. Eden izmed teh je požrešna aproksimacijska metoda (angl. Greedy Approximation), ki se je izkazala kot uspešna, kadar je aktivna podmnožica izbrana glede na nek kriterij. Sam postopek se prične z prazno množico $\mathcal{I}$ in množico $\mathcal{R}$, ki vsebuje vse učne primere, nato pa postopoma doda vsak primer posebej v aktivno množico. Če dodani primer ustreza kriterijem in je model bolj optimalen glede na prejšnjega, se primer obdrži, sicer pa se ga odstrani. Ta postopek se lahko uporablja pri vseh predhodno opisanih metodah: podmnožica regresorjev (SR), podmnožica podatkov (SD) in aproksimacija s projekcijo procesa (PP).

Glavno vprašanje, ki se pojavi, pa je, kakšen kriterij naj se uporabi za določitev aktivne podmnožice primerov. Predlaganih je bilo že kar nekaj metod, med njimi metoda informativnih vektorjev (angl. Informative Vector Machine - IVM) [6], kriterij informativnega prispevka (angl. Informative Gain) [16], metoda zredčenega vzorčenja spektra (angl. Sparse spectral Sampling) [3], sprotno redčenje (angl. Iterative Sparse) [1]. V naslednjem poglavju se bomo posvetili metodi sprotnega redčenja.

Poglavje 4

Sprotna aproksimacija GP modelov

Metoda sprotne aproksimacije združuje idejo razredčene predstavitve z algoritmom, ki omogoča sprotno učenje z Gaussovimi procesi. To omogoča sprotno gradnjo podmnožice relevantnih vhodnih primerov, na katerih temeljijo nadaljnje napovedi.

Bistvo postopka sta izraza za posterirorno srednjo vrednost $\mu_t(\mathbf{x}^*)$ in posteriorno kovarianco $C_t(\mathbf{x}, \mathbf{x}')$ (indeks t označuje število primerov), ki jih bomo izpeljali v naslednjem razdelku. Obe količini sta zvezni funkciji in jih lahko predstavimo kot končno število linearnih kombinacij jeder $C_0(\mathbf{x}, \mathbf{x_i})$ izračunanih za vhodni primer $\mathbf{x_i}$.

Z uporabo zaporednih projekcij posteriornega procesa na »področje« Gaussovih procesov, dobimo rekurzivno aproksimacijo, ki jo predstavimo s parametri. Ker število parametrov narašča s številom učnih primerov, uporabimo drugo vrsto projekcije za pridobivanje podmnožice relevantnih vhodnih primerov, na podlagi katere temeljijo napovedi.

4.1 Sprotno učenje GP

Kot smo že omenili v razdelku 2.1 učenje z Gaussovimi procesi temelji na Bayesovem načinu, kar pomeni, da moramo za pridobitev posteriorne porazdelitve verjetnosti p_{post} izračunati navadno analitično neizračunljiv integral. Omenili smo tudi metode za aproksimacijo tega integrala, med njimi metodo, ki je, za razliko od pogosteje uporabljenih, primerna za sprotno učenje. Ta metoda temelji na sledečem izreku, ki prikazuje zapis posteriorne srednje vrednosti in posterirorne kovariance procesa pri poljubnem vhodu kot kombinacijo

končnega števila parametrov, kateri so odvisni le od učnih primerov.

Izrek 1 (Parametrizacija)**.** *Rezultat Bayesove posodobitve z uporabo apriorne srednje vrednosti μ_o in jedra $C(\mathbf{x}, \mathbf{x}')$ ter podatkov $\mathcal{D} = \mathbf{X}, \mathbf{y}$ je proces s funkcijama srednje vrednosti in jedra določenima kot*

$$\begin{aligned} \mu_{post} &= \mu_0 + \sum_{i=1}^{N} C(\mathbf{x}, \mathbf{x}_i) q(i) \\ C_{post}(\mathbf{x}, \mathbf{x}') &= C(\mathbf{x}, \mathbf{x}_i) + \sum_{i,j=1}^{N} C(\mathbf{x}, \mathbf{x}_i) R(ij) C(\mathbf{x}_j, \mathbf{x}'). \end{aligned} \tag{4.1}$$

Parametra $q(i)$ in $R(ij)$ sta določena kot

$$q(i) = \frac{1}{Z} \int p_0(f) \frac{\partial p(D|f)}{\partial f(\mathbf{x}_i)} df \qquad R(ij) = \frac{1}{Z} \int p_0(f) \frac{\partial^2 p(D|f)}{\partial f(\mathbf{x}_i \partial f(\mathbf{x}_j))} df - q(i)q(j) \tag{4.2}$$

kjer je $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^T$ in $Z = \int p_0(f) P(D|f) df$ normalizacijska konstanta.

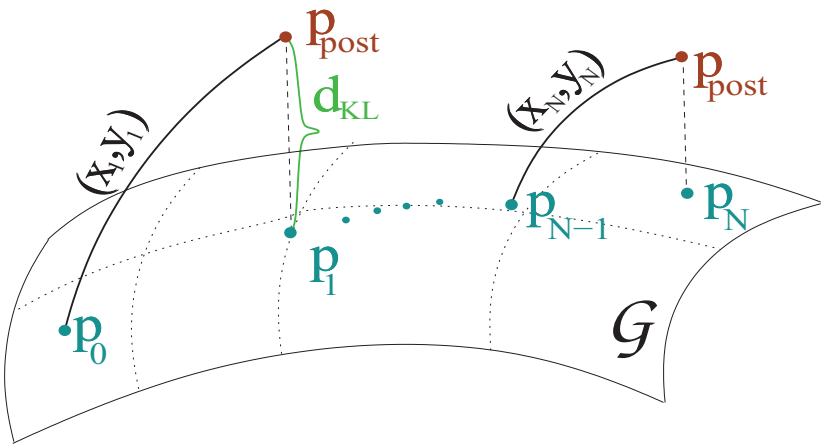
Dokaz tega izreka je opisan v [1].

Tudi v tej predstavitvi so uporabljeni navadno analitično neizračunljivi integrali, zato prav tako potrebujemo njihovo aproksimacijo. Vendar pa v temu postopku uporabljamo metodo, ki posterirorno porazdelitev verjetnosti aproksimira z Gaussovim procesom [22]. Izrazimo jo lahko z variacijskim postopkom, kjer minimiziramo razliko med »pravo« in aproksimirano porazdelitvijo verjetnosti. Najpogosteje se uporablja Kullback-Leiberjeva divergenca med dvema porazdelitvima, ki je definirana kot

$$KL(p|q) = \int p(\mathbf{\Theta}) \ln \frac{p(\mathbf{\Theta})}{q(\mathbf{\Theta})} d\mathbf{\Theta} \tag{4.3}$$

kjer je $\mathbf{\Theta}$ vektor parametrov. Če s $\hat{p}$ označimo aproksimirano porazdelitev, navadno minimiziramo $KL(\hat{p}||p_{post})$, saj za razliko od $KL(p_{post}||\hat{p})$, potrebuje le izračune napovedi sledljivih porazdelitev, t.j. tistih, ki jih lahko izračunamo z aproksimacijo.

Vendar pa v tej metodi, s katero želimo sprotno posodobiti model z vsakim primerom posebej, uporabimo nekoliko drugačen postopek. Recimo, da s $\hat{p}_t$ označimo aproksimacijo porazdelitve po t primerih in s p_{post} posteriorno porazdelitev, ki jo dobimo z Bayesoveim pravilom. Ker ta porazdelitev p_{t+1} ni več



Slika 4.1: Predstavitev sprotne aproksimacije posteriorne porazdelitve. Pri tem se uporablja aproksimirana porazdelitev iz prejšnjega koraka kot apriori porazdelitev pri naslednjem koraku.

Gaussova, jo približamo najbližji porazdelitvi Gaussovega procesa $\hat{p}_{t+1}$ (glej sliko 4.1) z minimizacijo divergence $KL(p_{post}||\hat{p})$. V tem primeru je to možno, saj posteriorna porazdelitev vsebuje verjetnost le za en primer, kar pomeni, da vsebuje le enojni integral, ki pa je analitično izračunljiv. S tem približanjem se srednja vrednost in kovarianca porazdelitve p_{post} in aproksimacije posteriorne porazdelitve $\hat{p}_{t+1}$ ujemata, kar je podrobneje prikazano v [12].

Za sprotno izračunavanje srednje vrednosti in kovariance zaporedoma uporabimo Izrek 1. To nas pripelje do:

$$\begin{aligned} \mu_{t+1} &= \mu_t + q^{(t+1)} C_t(\mathbf{x}_t, \mathbf{x}_{t+1}) \\ C_{t+1}(\mathbf{x}, \mathbf{x}') &= C_t(\mathbf{x}, \mathbf{x}') + r^{(t+1)} C_t(\mathbf{x}, \mathbf{x}_{t+1}) C_t(\mathbf{x}_{t+1}, \mathbf{x}') \end{aligned} \tag{4.4}$$

kjer sta $q^{(t+1)}$ in $r^{(t+1)}$ (izpeljava je opisana v prilogi B [1])

$$\begin{aligned} q^{(t+1)} &= \frac{\partial}{\partial \langle \mathbf{f}_{t+1} \rangle_t} \ln \langle p(\mathbf{y}_{t+1}|\mathbf{f}_{t+1}) \rangle_t \\ r^{(t+1)} &= \frac{\partial^2}{\partial \langle \mathbf{f}_{t+1} \rangle_t} \ln \langle p(\mathbf{y}_{t+1}|\mathbf{f}_{t+1}) \rangle_t \end{aligned} \tag{4.5}$$

kjer $\langle \cdot \rangle$ pomeni povprečje vrednosti.

Naj še enkrat omenimo, da je za izračun koeficientov v enačbi (4.5) potreben izračun enojnega integrala, kar lahko storimo analitično. Z razvojem rekurzije pravil (4.6) pridemo do parametričnega zapisa aproksimacije posterirorne porazdelitve po t primerih kot:

$$\begin{aligned} \mu_t &= \sum_{i=1}^{t} C(\mathbf{x}, \mathbf{x}_i)\alpha_t(i) = \boldsymbol{\alpha}_t^T \mathbf{k}_x \\ C_t(\mathbf{x}, \mathbf{x}') &= C(\mathbf{x}, \mathbf{x}') + \sum_{i,j=1}^{t} C(\mathbf{x}, \mathbf{x}_i)\gamma_t(ij)C(\mathbf{x}_j, \mathbf{x}') = C(\mathbf{x}, \mathbf{x}') + \mathbf{k}_x^T \boldsymbol{\gamma}_t \mathbf{k}_x \end{aligned} \tag{4.6}$$

kjer sta koeficienta $\alpha_t(i)$ in $\gamma_t(ij)$ neodvisna od x in x' (podrobnosti so v [1]). Za poenostavitev so vrednosti $\alpha_t(i)$ združene v vektor $\boldsymbol{\alpha}_t = [\alpha_t(1), \ldots, \alpha_t(t)]^T$, vrednosti $\gamma_t(ij)$ združene v matriko $\boldsymbol{\Gamma}_t = \gamma_t(ij)_{i,j=1,t}$ ter vrednosti $C(\mathbf{x}, \mathbf{x}_i)$ združene v vektor $\mathbf{k}_x = [C(\mathbf{x}, \mathbf{x}_1), \ldots, C(\mathbf{x}, \mathbf{x}_i)]^T$.

Parametre iz enačbe (4.6) izračunamo z rekurzijo:

$$\begin{aligned} \boldsymbol{\alpha}_{t+1} &= \mathrm{T}_{t+1}(\boldsymbol{\alpha}_t) + q^{(t+1)}\mathbf{s}_{t+1} \\ \boldsymbol{\Gamma}_{t+1} &= \mathrm{U}_{t+1}(\boldsymbol{\gamma}_t) + r^{(t+1)}\mathbf{s}_{t+1}\mathbf{s}_{t+1}^T \\ \mathbf{s}_{t+1} &= \mathrm{T}_{t+1}(\boldsymbol{\Gamma}_t \mathbf{k}_{t+1}) + \mathbf{e}_{t+1} \end{aligned} \tag{4.7}$$

kjer je $\mathbf{k}_{t+1} = \mathbf{k}_{x_{t+1}}$ in $\mathbf{e}_{t+1}$, ki je enotski vektor dolžine $t+1$. Za lažjo predstavitev smo vpeljali koeficient $\mathbf{s}_{t+1}$ ter operatorja T_{t+1} in U_{t+1}, ki razširjata t-dimenzionalen vektor oziroma matriko na $t+1$-dimenzinalnega, tako da »pripneta« ničle na konec vektorja oziroma v zadnji stolpec in vrstico matrike.

Ker je $\mathbf{e}_{t+1}$ enotski vektor dolžine $t+1$, velikost vektorja $\boldsymbol{\alpha}$ in matrike $\boldsymbol{\Gamma}$ raste z vsakim vhodnim primerom. Zato bomo v nadaljevanju opisali, kako lahko spremenimo postopek, da bomo lahko nadzirali število parametrov.

4.2 Zredčena predstavitev

Število parametrov lahko nadziramo tako, da uvedemo posodobitev modela, ki ne poveča števila parametrov $\boldsymbol{\alpha}$ in $\boldsymbol{\Gamma}$. Ta se izvede kadar napaka aproksimacije ne preseže določene vrednosti. To bi lahko dosegli, če bi za nov vhodni primer držalo

$$C(\mathbf{x}, \mathbf{x}_{t+1}) = \sum_{i=1}^{t} \hat{\mathbf{e}}_{t+1}(i)C(\mathbf{x}, \mathbf{x}_i) \tag{4.8}$$

za vsak $\mathbf{x}_i$. V tem primeru bi imeli posodobitev (4.6) z uporabo le prvih t vhodnih primerov, vendar s posodobljenima parametroma $\hat{\boldsymbol{\alpha}}$ in $\hat{\boldsymbol{\Gamma}}$. Iz (4.7)

opazimo, da bi bila pri tem potrebna le zamenjava vektorja $\mathbf{s}_{t+1}$ z

$$\hat{\mathbf{s}}_{t+1} \approx \mathbf{\Gamma}_t \mathbf{k}_{t+1} + \hat{\mathbf{e}}_{t+1}, \tag{4.9}$$

kjer je $\hat{\mathbf{e}}_{t+1}$ vektor dimenzije t. Na žalost enačba (4.8) ni uporabna za večino primerov, vendar pa lahko, kot aproksimacijo, uporabimo posodobitev (4.9), kjer je $\hat{\mathbf{e}}_{t+1}$ določen z minimizacijo napake

$$\left\| C(\cdot, \mathbf{x}_{t+1}) - \sum_{i=1}^{t} \hat{\mathbf{e}}_{t+1}(i) C(\cdot, \mathbf{x}_i) \right\|^2, \tag{4.10}$$

kjer je $\|\cdot\|$ primerno definirano pravilo v prostoru funkcij vhodov $\mathbf{x}$. Z uporabo pravila (4.10) kot skalarnega produkta reproduciranja jedra Hilbertovega prostora (angl. reproducing kernel Hilbert Space - RKHS) in minimizacijo pridemo do

$$\hat{\mathbf{e}}_{t+1} = \mathbf{K}_t^{-1} \mathbf{k}_{t+1} \tag{4.11}$$

kjer je $\mathbf{K}_t = C(\mathbf{x}_i, \mathbf{x}_{t+1})_{ij,=1,t}$. Podrobnejša razlaga se nahaja v [1].

Približno posodabljanje s (4.9) se bo izvedlo le kadar aproksimacijska napaka, ki jo bomo opisali v nadaljevanju, ne bo presežena. Sicer se bo izvedla »polna« posodobitev, pri kateri se bo povečalo število parametrov in množica z relevantnimi vhodnimi primeri, ki jo bomo imenovali množica baznih vektorjev (angl. basis vector set - $\mathcal{BV}$ set), njene elemente pa bazni vektorji (angl. basis vectors). Z zaporednim izvajanjem bodo tako nekateri vhodni primeri vključeni v množico $\mathcal{BV}$, ostali pa ne, vendar bodo kljub temu vplivali na model.

Tako smo prišli do razredčene predstavitve posteriorne porazdelitve določene z množico $\mathcal{BV}$ in pripadajočima parametroma $\boldsymbol{\alpha}$ in $\boldsymbol{\gamma}$:

$$\mu = \sum_{i \in \mathcal{BV}} C(\mathbf{x}, \mathbf{x}_i) \alpha(i) \tag{4.12}$$

$$C(\mathbf{x}, \mathbf{x}') = C(\mathbf{x}, \mathbf{x}') + \sum_{i,j \in \mathcal{BV}} C(\mathbf{x}, \mathbf{x}_i) \gamma(ij) C(\mathbf{x}_j, \mathbf{x}') \tag{4.13}$$

4.2.1 Aproksimacijska napaka

Kot smo že omenili, potrebujemo pravilo s katerim bomo določali ali bo vhodni primer vključen v množico $\mathcal{BV}$ ali ne. Pravilo, ki ga bomo uporabili, temelji na razliki med napovedano srednjo vrednostjo dobljeno iz modela iz baznih vektorjev in aproksimirano srednjo vrednostjo v naslednjem koraku:

$$\triangle\ \mu_{t+1} = \mu_{t+1} - \hat{\mu}_{t+1} \tag{4.14}$$

kjer je $\hat{\mu}_{t+1}$ srednja vrednost aproksimirane porazdelitve. Sešteveh vseh absolutnih vrednosti razlik za primere v množici $\mathcal{BV}$ in vhodnega primera nas pripelje do:

$$\varepsilon_{t+1} = \sum_{i=1}^{t+1} | \triangle \mu_{i_{t+1}}| = |q^{(t+1)}| \sum_{i=1}^{t+1} |C(\mathbf{x}_i, \mathbf{x}_{t+1}) - \hat{C}(\mathbf{x}_i, \mathbf{x}_{t+1})|, \tag{4.15}$$

ter z uporabo RKHS nadalje do

$$\varepsilon_{t+1} = |q^{(t+1)}|(\mathbf{k}^*_{t+1} - \mathbf{k}^T_{t+1}\mathbf{K}^{-1}_t\mathbf{k}_{t+1}) = |q^{(t+1)}|\nu_{t+1} \tag{4.16}$$

kjer je $\mathbf{k}^*_{t+1} = C(\mathbf{x}_{t+1}, \mathbf{x}_{t+1})$. Iz enačbe (4.16) je razvidno, da je napaka ε_{t+1} izražena kot produkt dveh izrazov. Prvi izraz $q^{(t+1)}$, ki je v primeru, da bo vhodni primer vključen v množico $\mathcal{BV}$, enak koeficientu $\boldsymbol{\alpha}_{t+1}$, imenujemo del odvisen od podobnosti (angl. »likelihood-dependent« part). Drugi izraz

$$\nu_{t+1} = \mathbf{k}^*_{t+1} - \mathbf{k}^T_{t+1}\mathbf{K}^{-1}_t\mathbf{k}_{t+1} \tag{4.17}$$

je geometrijski del (angl. geometrical part) in podaja »novost« (angl. novelty) trenutnega vhodnega primera.

Za izračun geometrijskega dela napake ε_{t+1} moramo izračunati inverz matrike v vsakem koraku. Temu računsko zahtevnemu delu se lahko izognemo s sprotnim posodabljanjem inverzne matrike $\mathbf{Q}_t = \mathbf{K}^{-1}_t$, ki bo koristil pri brisanju baznih vektorjev:

$$\mathbf{Q}_{t+1} = \mathrm{U}_{t+1}(\mathbf{Q}_t) + \nu^{-1}_{t+1}(\mathrm{T}_{t+1}(\hat{\mathbf{e}}_{t+1}) - \mathbf{e}_{t+1})(\mathrm{T}_{t+1}(\hat{\mathbf{e}}_{t+1}) - \mathbf{e}_{t+1})^T. \tag{4.18}$$

kjer sta U_{t+1} in T_{t+1} operatorja za razširitev matrike oziroma vektorja, vpeljana v enačbi (4.7). Izpeljava te enačbe je podrobno opisana v [1].

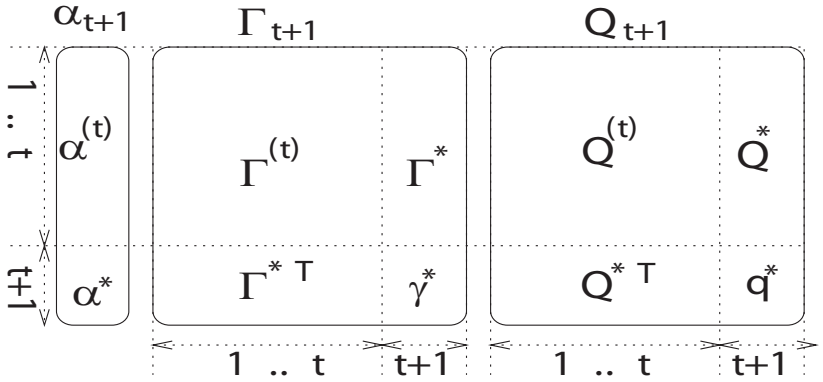
4.2.2 Brisanje baznega vektorja

Za nadziranje velikosti množice $\mathcal{BV}$ oziroma števila parametrov, moramo poleg dodajanja baznih vektorjev, le-te znati tudi brisati. Z vsakim novim vhodnim primerom, ki ga prepoznamo kot pomembnega in ga želimo dodati v množico $\mathcal{BV}$, moramo najprej izbrisati bazni vektor z najmanjšo napako ter ga nadomestiti z novim. Najprej bomo predstavili brisanje baznih vektorjev, nato pa še kriterij, ki določa kateri bazni vektor bo odstranjen iz množice $\mathcal{BV}$.

Predpostavimo, da je bil v množico $\mathcal{BV}$ pravkar dodan vhodni vektor $\mathbf{x}_{t+1}$. To pomeni, da je bil zadnji izvedeni korak posodobitev z $t+1$-im enotskim vektorjem $\mathbf{e}_{t+1}$, in da lahko določimo vrednosti koeficientov $q^{(t+1)}$, $r^{(t+1)}$ in

$\mathbf{s}_{t+1}$ iz enačb (4.7), izračunamo $\hat{\mathbf{e}}_{t+1}$ in uporabimo (4.9) za posodobitev brez vključevanja vhodnega primera v množico $\mathcal{BV}$.

Če predpostavimo, da imamo $t+1$ baznih vektorjev, da ima $\boldsymbol{\alpha}_{t+1}$ $t+1$ elementov, da sta matriki $\boldsymbol{\Gamma}_{t+1}$ in $\mathbf{Q}_{t+1}$ velikosti $(t+1)(t+1)$ ter da želimo zbrisati zadnji dodani bazni vektor, potem je dekompozicija taka, kot je predstavljena na sliki 4.2.



Slika 4.2: Dekompozicija parametrov modela

Izračun parametrov modela v predhodnem koraku in uporaba posodobitve brez razširitve pripelje do enačb za brisanje:

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^{(t)} - \boldsymbol{\alpha}^* \frac{Q^*}{q^*} \tag{4.19}$$

$$\hat{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma}^{(t)} + \boldsymbol{\gamma}^* \frac{\mathbf{Q}^* \mathbf{Q}^{*^T}}{q^{*^2}} - \frac{1}{q^*}[\mathbf{Q}^* \boldsymbol{\Gamma}^{*^T} + \boldsymbol{\Gamma}^* \mathbf{Q}^{*^T}] \tag{4.20}$$

$$\hat{\mathbf{Q}} = \mathbf{Q}^{(t)} - \frac{\mathbf{Q}^* \mathbf{Q}^{*^T}}{q^*} \tag{4.21}$$

kjer so $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\Gamma}}$ in $\hat{\mathbf{Q}}$ parametri po brisanju zadnjega baznega vektorja in $\boldsymbol{\Gamma}^{(t)}$, $\mathbf{Q}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$, $\mathbf{Q}^*$, $\boldsymbol{\Gamma}^*$, q^* ter $\boldsymbol{\gamma}^*$ parametri pred brisanjem. Grafična predstavitev vsakega izmed njih je prikazana na sliki 4.2.

Kot smo že omenili, oceno baznega vektorja, ki ga brišemo iz množice $\mathcal{BV}$ (v tem primeru zadnji dodani primer), določa produkt parametrov $q^{(t+1)}$ in ν_{t+1}. Iz tega sledi ocena

$$\varepsilon_{t+1} = \frac{\boldsymbol{\alpha}^*}{q^*} = \frac{\boldsymbol{\alpha}_{t+1}(t+1)}{\mathbf{Q}_{t+1}(t+1, t+1)} \tag{4.22}$$

Ker pa so bazni vektorji naključno razporejeni, lahko izračunamo oceno za vsak bazni vektor posebej

$$\varepsilon_i = \frac{|\boldsymbol{\alpha}_{t+1}(i)|}{\mathbf{Q}_{t+1}(i,i)}. \tag{4.23}$$

Torej, če je potrebno brisanje baznega vektorja, potem izbrišemo bazni vektor z najmanjšo oceno iz enačbe (4.22).

4.3 Algoritem

Začetno stanje algoritma je določeno z prazno množico baznih vektorjev ($\mathcal{BV}$), maksimalnim številom baznih vektorjev v množici $\mathcal{BV}$ določenim z m, kovariančno funkcijo C in toleranco ε_{tol}, s katerim določamo ali nov primer dodamo v množico $\mathcal{BV}$ ali ne. Parametri $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ ter inverzna matrika $\mathbf{Q}$ so v začetnem stanju nastavljeni kot prazne vrednosti.

Za vsak vhodni primer po vrsti izvedemo naslednje korake:

1. Izračunamo $q^{(t+1)}$, $r^{(t+1)}$, $\mathbf{k}^*_{t+1}$, $\hat{\mathbf{e}}_{t+1}$ in ν_{t+1}.
2. Če $\nu_{t+1} < \varepsilon_{tol}$ izvedemo »zmanjšano« posodobitev - z uporabo vektorja $\hat{\mathbf{e}}_{t+1}$ iz enačbe (4.7) posodobimo parametra $\boldsymbol{\alpha}$ in $\boldsymbol{\Gamma}$, tako da se njuna velikost ne poveča, ter nadaljujemo z naslednjim vhodnim primerom,
3. sicer izvedemo »polno« posodobitev (4.7) z uporabo enotskega vektorja $\mathbf{e}_{t+1}$, ter dodamo vhodni primer v množico $\mathcal{BV}$ in izračunamo inverz nove razširjene matrike z uporabo enačbe (4.18).
4. Če je velikost množice $\mathcal{BV}$ večja od m, izračunamo ocene ε_i za vse bazne vektorje z enačbo (4.23), poiščemo tistega z najmanjšo oceno in ga izbrišemo iz množice $\mathcal{BV}$ z enačbo (4.19).
5. Nadaljujemo z naslednjim vhodnim primerom.

Časovna zahtevnost tega algoritma raste s kvadratom m - določene maksimalne velikosti množice $\mathcal{BV}$. Z iteracijo skozi vse vhodne primere sledi časovna zahtevnost $O(nm^2)$.

Poglavje 5

Primeri

V tem poglavju bomo predstavili rezultate uporabe sprotnega učenja modelov na podlagi Gaussovih procesov. Prvi primer, z manj učnimi primeri (76), je namenjen prikazu delovanja metode, drugi primer, z veliko učnimi primeri (14.520), pa je namenjen predstavitvi učinkovitosti metode.

Pri obeh primerih bomo uporabili regresijski model in Gaussovo kovariačno funkcijo, ker smo ocenili, da gre za gladko in stacionarno funkcijo.

$$C(\mathbf{x}, \mathbf{x}') = v \exp\left[\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right] \tag{5.1}$$

kjer je v vertikalni skalirni faktor (angl. vertical length scale) in l horizontalni skalirni faktor (angl. horizontal length scale). Z uporabo parametrizacije (4.6) v smislu parametrov $\boldsymbol{\alpha}$ in $\boldsymbol{\gamma}$ pridemo do porazdelitve za $\mathbf{y}$ pri vhodu $\mathbf{x}$

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}) = \left(\frac{1}{2\pi l_{\mathbf{x}}^2}\right)^2 \exp\left[-\frac{\|\mathbf{y} - \boldsymbol{\alpha}^T \mathbf{k}_{\mathbf{x}}\|^2}{2\pi l_{\mathbf{x}}^2}\right] \tag{5.2}$$

kjer je $l_x^2 = l_0^2 + \mathbf{k}_x^T C_t \mathbf{k}_x + \mathbf{k}_x^*$. Iz tega sledita koeficienta $q^{(t+1)}$ in $r^{(t+1)}$ v posodobitvenih pravilih (4.7) določena:

$$q^{(t+1)} = (\mathbf{y} - \boldsymbol{\alpha}_t^T \mathbf{k}_x)/l^2 \qquad in \qquad r^{(t+1)} = (-1)/l^2. \tag{5.3}$$

5.1 Predstavitveni primer

Namen tega primera je na enostaven način pokazati delovanje metode sprotnega modeliranja na podlagi GP. Za primer smo modelirali nelinearno funkcijo odvisno od neodvisne spremenljivke x:

$$f(x) = \frac{1}{20}(x+4)(x+2)(x+1)(x-1)(x-3) + 2 + v \tag{5.4}$$

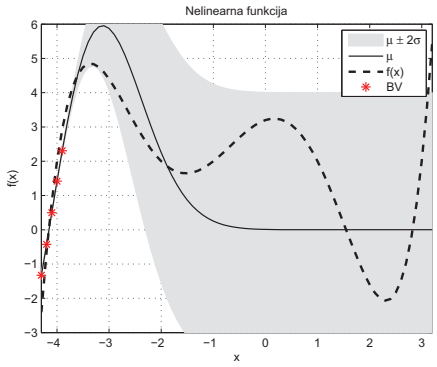
na intervalu od $x \in [-4.3, 3.2]$ s korakom 0.1 - torej skupno 76 točk, pri čemer je varianca Gaussovega šuma v na izhodu enaka $\sigma^2 = 0.01$. Opis modela smo omejili na 10 baznih vektorjev, ter uporabili naslednje vrednosti hiperparametrov $v = 1$ in $w = 2$. Modelirali smo tako zaporedno razporejene točke kot naključno razporejene točke.

Pri obeh načinih, delni rezultati so prikazani na slikah 5.1 in 5.2, je razvidno sprotno učenje metode. S prihodom vsakega primera (točke) posebej se model postopoma izboljšuje. To lahko opazimo tako iz srednje vrednosti (prekinjena črta), ki je vedno bliže nelinearni funkciji (5.4) (modra črta), kot tudi iz variance (siv pas), ki je vedno ožja, kar pomeni vedno večje zaupanje. Rdeče zvezdice pa označujejo bazne vektorje - primere (točke), ki najbolje opisujejo model.

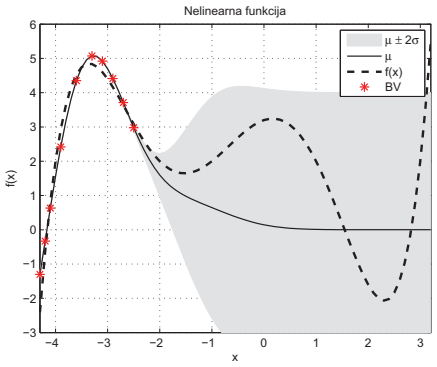
Opazimo lahko tudi odvisnost modela od zaporedja »prihoda« učnih primerov. To je razvidno tako iz slik kot iz vrednosti različnih mer napak napovedi prikazanimi v tabeli 5.1. Na slikah je viden različen razpored baznih vektorjev in posledično tudi nekoliko različne srednje vrednosti. Z vrednostmi napak pri različnih razporeditvah učnih primerov pa smo želeli dodatno okrepiti ugotovitve iz slik. Različne vrednosti mer napak pri različnih razporeditvah to tudi potrjujejo.

	zaporedno	naključno 1	naključno 2
MSE	0,1442	0,1052	0,1115
MAE	0,2513	0,2203	0,2270
LPD	0,3204	0,1910	0,2576
MRSE	0,1429	0,1212	0,1243

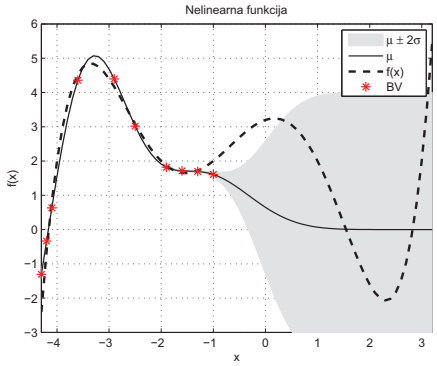
Tabela 5.1: Vrednosti merskih napak napovedanih vrednosti



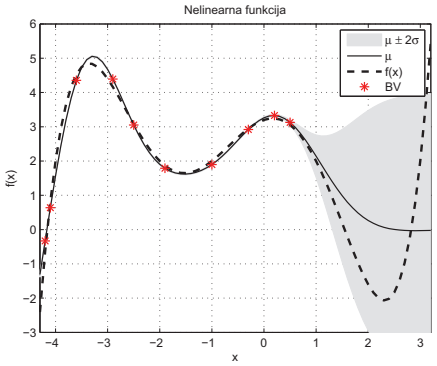
(a) 5. korak



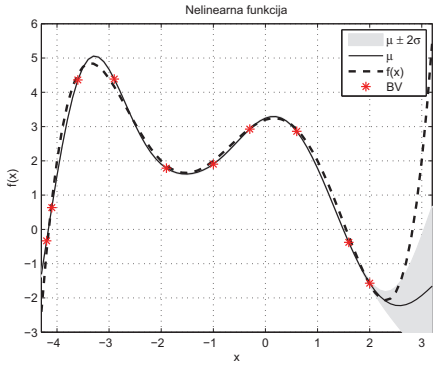
(b) 20. korak



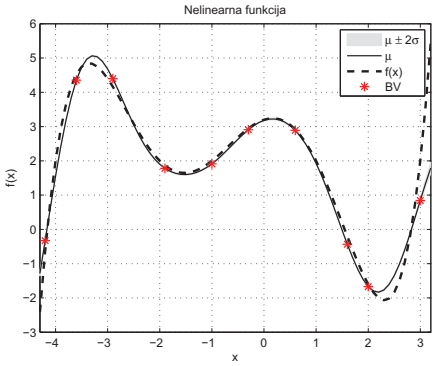
(c) 35. korak



(d) 50. korak

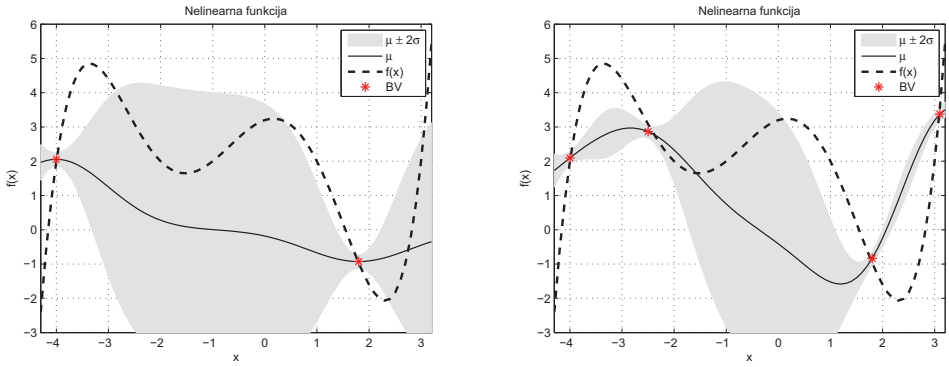


(e) 65. korak



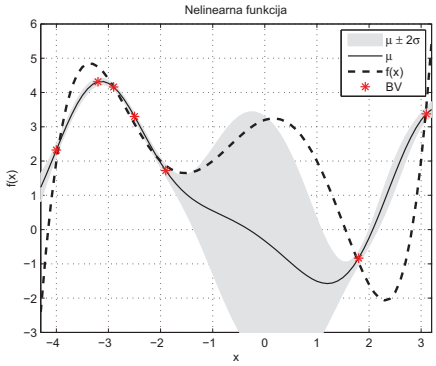
(f) 75. korak

Slika 5.1: Zaporedna razporeditev učnih točk

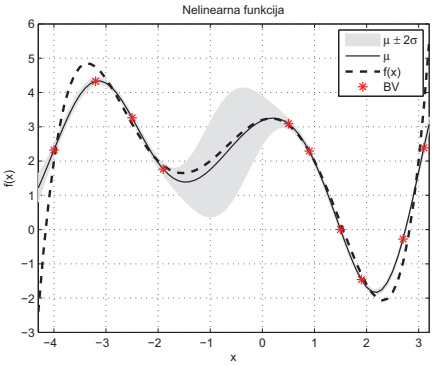


(a) 2. korak

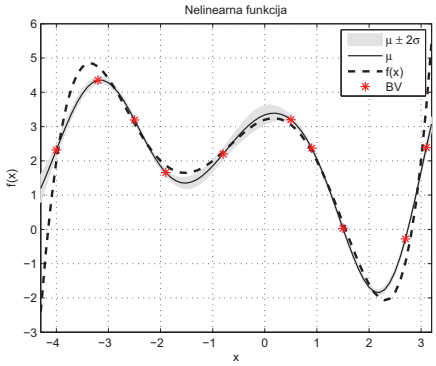
(b) 4. korak



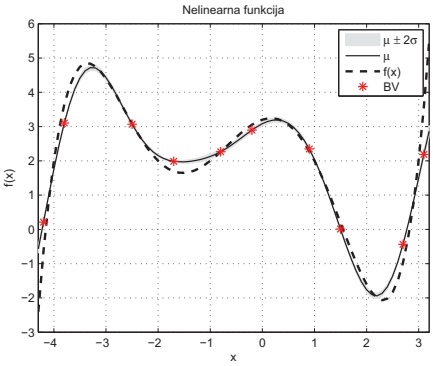
(c) 7. korak



(d) 13. korak



(e) 16. korak



(f) 34. korak

Slika 5.2: Naključna razporeditev učnih točk

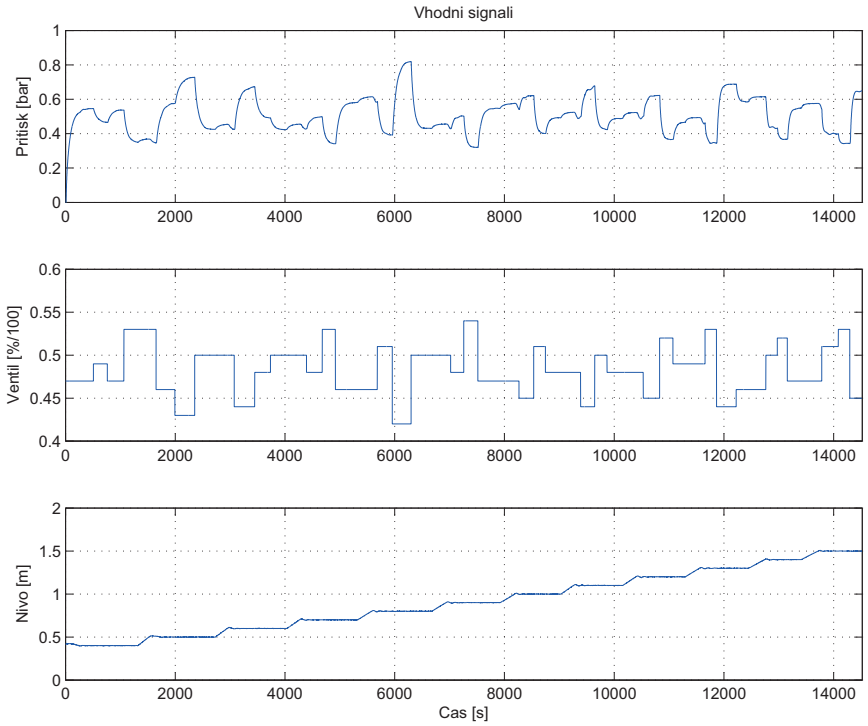
5.2 Primer - ločevalnik

Namen tega primera je pokazati učinkovitost metode sprotnega modeliranja na podlagi GP. Za primer smo modelirali proces priprave plina (podrobneje predstavljenega v dodatku A), imenovan tudi ločevalnik plina in tekočine, ki je multivariabilen nelinearni proces, v katerem kot regularni veličini nastopata tlak p na področju od 0.4 do 0.7 bar in nivo vode h_1 v tlačni posodi na področju od 0.4 do 1.5m. Regulirna signala sta u_1 za odprtost ventila na izhodu plina iz posode in u_2 za odprtost ventila na izhodu tekočine iz posode, oba na področjih od 0 (zaprt ventil) do 1 (odprt ventil). Glavni viri nelinearnosti procesa so nelinearne odvisnosti pretokov skozi ventile od tlačnih razlik na ventilih in nelinearna odvisnost dinamike tlaka plina od nivoja vode v tlačni posodi.

Ta proces je namenjen zajemanju in ohlajanju dimnih plinov, ki vsebujejo CO_2 - pri tem se dimni plini v prirejenem injektorju mešajo s hladilno vodo, ter ločevanju mešanice hladilne vode in dimnih plinov na ponovno uporabljivo vodo in na dimne pline pod tlakom primernim za proces kemijske nevtralizacije bazičnih odplak.

Za vpihovanje v nevtralizacijski reaktor morajo biti dimni plini ohlajeni in pod ustreznim tlakom - približno 0.5 bara. Od tlaka plina je namreč odvisna kvaliteta raztapljanja CO_2 v bazični odplaki in s tem učinkovitost nevtralizacije.

Za modeliranje obnašanja tlaka tega procesa smo uporabili 14.520 učnih primerov, ki so bili izmerjeni med delovanjem procesa v času štirih ur s sekundim intervalom vzorčenja v procesnem laboratoriju Odseka za sisteme in vodenje na Institutu Jožef Stefan. Proces smo najprej modelirali le kot časovno vrsto, torej vrednost signala p (tlak) v odvisnosti od časa, nato pa še simulirali z naivno metodo, pri kateri smo kot vhodne podatke uporabili signal u_1 (odprtost ventila na izhodu plina), signal h_1 (nivo vode) in srednjo vrednost napovedi modela iz prejšnjega vzorčnega koraka. Vrednosti signalov v odvisnosti od časa so prikazana na slikah 5.3.



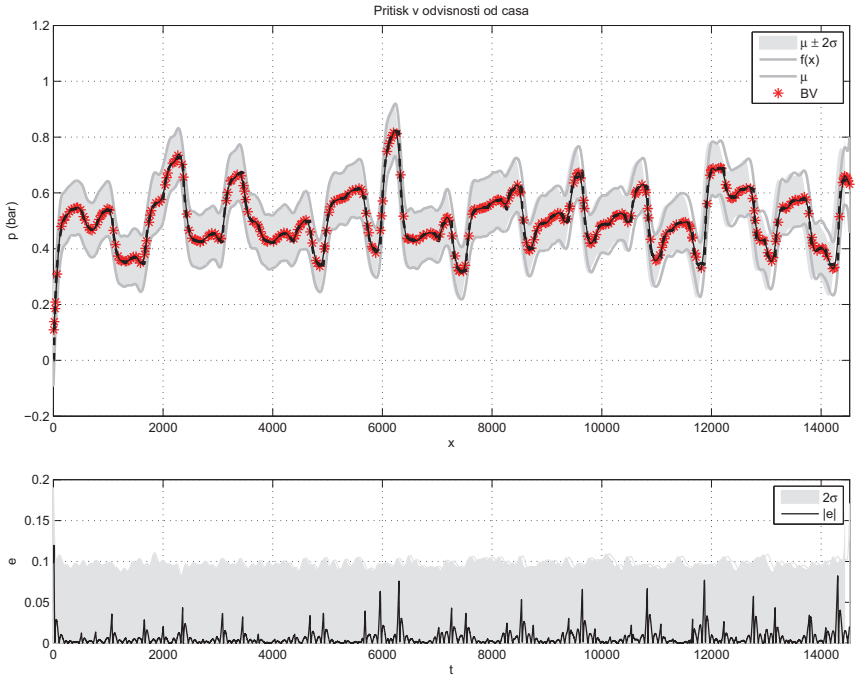
Slika 5.3: Vhodni signali procesa priprave plina

5.2.1 Modeliranje časovne vrste

S tem primerom smo želeli preveriti učinkovitost metode pri problemih z enim regresorjem, zato smo, kot smo že omenili, pri tem primeru modelirali vrednost signala p (tlak) v odvisnosti od časa. Začetne hiperparametre smo nastavili z vrednostmi $v = 0.15$ in $w = 10^{-5}$ ter tolerančni koeficient z vrednostjo $\varepsilon_{tol} = 10^{-11}$. Največjo velikost množice $\mathcal{BV}$ smo sicer nastavili na 1.000, vendar je, glede na nastavitve, za opis problema zadostovalo 351 baznih vektorjev. Rezultat modeliranja je prikazan na sliki 5.4, vrednosti različnih mer napak pa v tabeli 5.2.

MSE	MAE	LPD	MRSE
0,0001461	0,0073	-0,5902	0,1201

Tabela 5.2: Vrednosti merskih napak



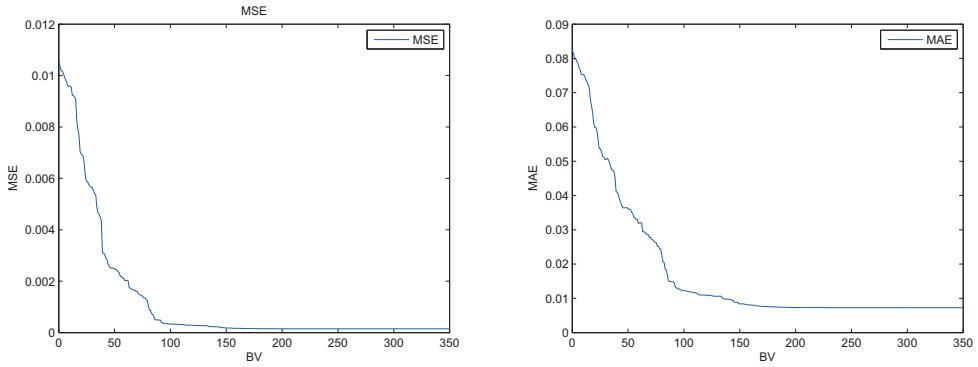
Slika 5.4: Pritisk v odvisnosti od časa

Tako iz slike kot iz tabele vrednosti mer napak lahko sklepamo, da model s 351 baznimi vektorji dobro opiše proces podan sicer s 14.520 učnimi točkami.

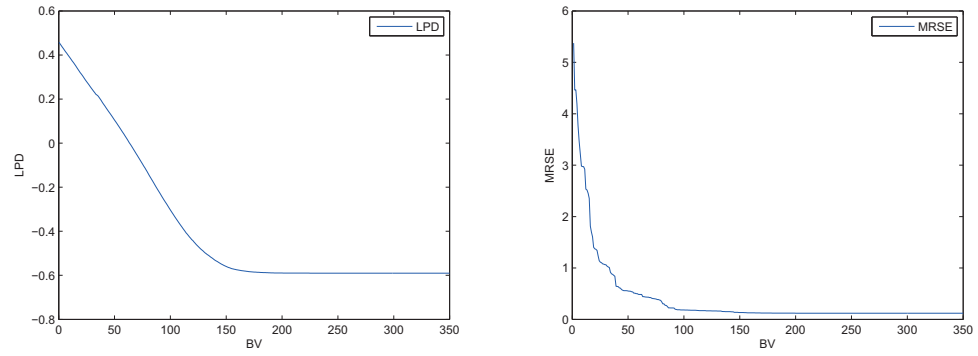
Število baznih vektorjev v odvisnosti od tolerančnega koeficienta

Želeli smo preveriti tudi vrednosti različnih mer napak v odvisnosti od števila baznih vektorjev. Poskus smo izvedli tako, da smo iz obstoječega modela postopoma odstranjevali bazne vektorje z najmanjšo oceno. Rezultat je prikazan na sliki 5.5;

Pri vseh merah napak lahko opazimo umiritev spremembe vrednosti od približno stopetdesetega baznega vektorja naprej. Na podlagi tega lahko sklepamo, da bi že z stopetdesetimi baznimi vektorji dovolj dobro opisali proces podan sicer s 14.520 točkami.



(a) MSE (b) MAE



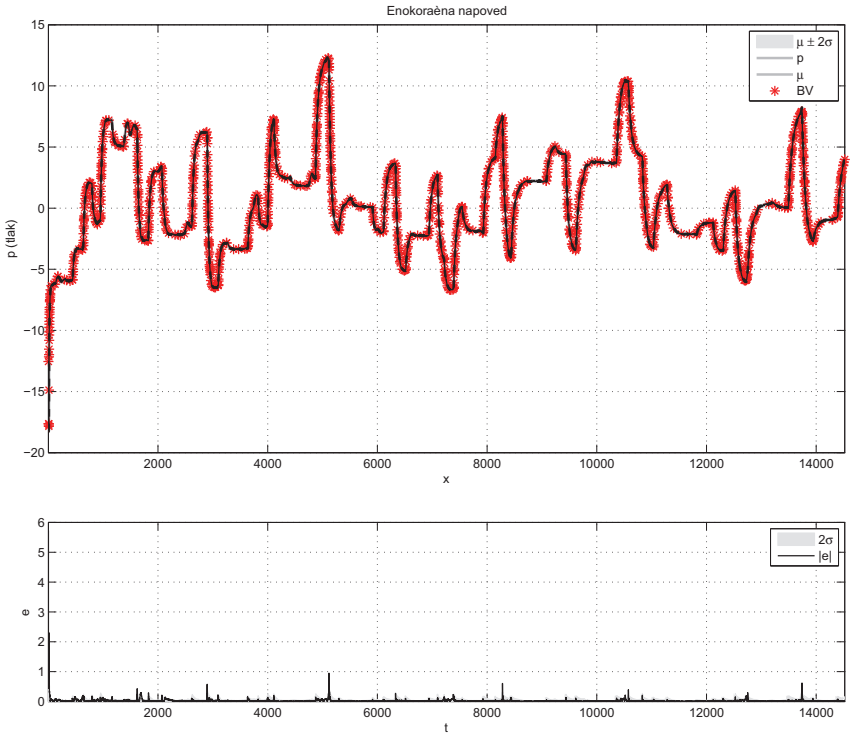
(c) LPD (d) MRSE

Slika 5.5: Vrednosti različnih mer napak v odvisnosti od števila baznih vektorjev

5.2.2 Modeliranje z več vhodnimi regresorji

S tem primerom smo želeli preveriti učinkovitost metode pri več-vhodnih problemih. Modelirali smo proces priprave plina znaslednjimi vhodnimi regresorji: tlak p, nivo vode h_1 v tlačni posodi, kot tretji regresor pa smo uporabili vrednost izhoda prejšnjega koraka vzorčenja, ne iteracije. Zaradi treh vhodov smo nastavili štiri hiperparametre z vrednostmi: $w_1 = 1.1$, $w_2 = 0.04$, $w_3 = 0.4545$ in $v = 0.3687$. Do teh vrednosti smo prišli postopoma - z iterativnim postopkom, tako da smo opazovali napovedi srednje vrednosti in variance modela glede na nastavljene vrednosti. Ko smo prišli do dovolj natančne srednje vrednosti in dovolj majhne variance, smo izvedli še optimizacijo z metodo največje podobnosti. Največje število baznih vektorjev smo sicer omejili na 1.500, ven-

dar je, glede na nastavitve, za opis procesa zadostovalo 1.191 baznih vektorjev. Rezultat enokoračne napovedi je prikazan na sliki 5.6.

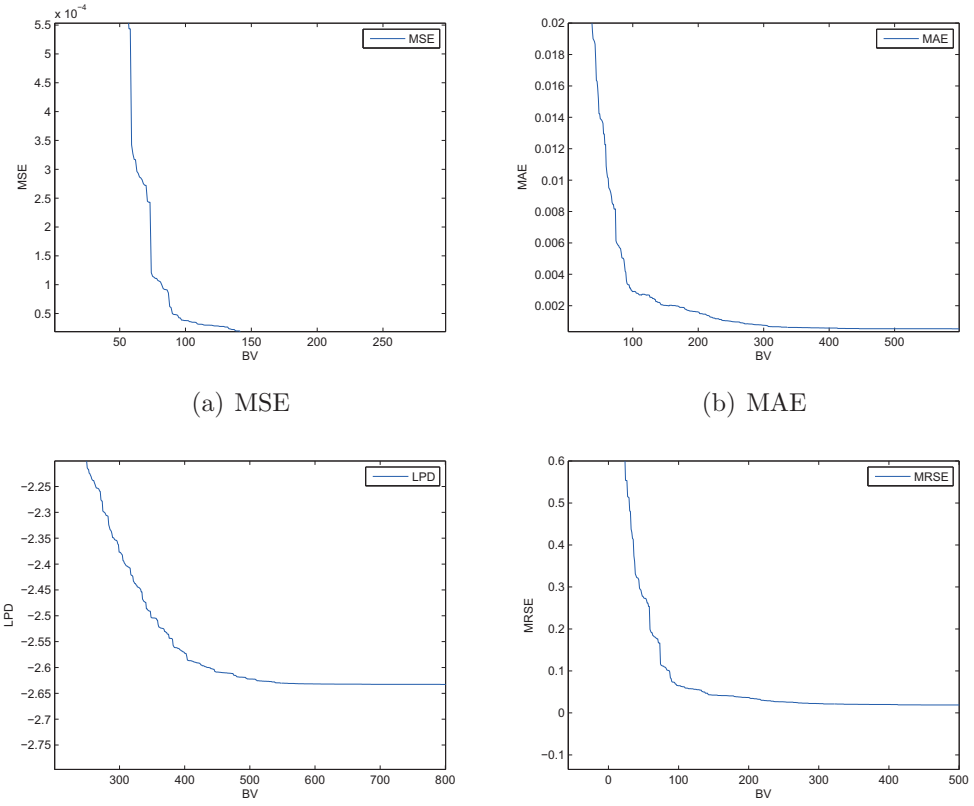


Slika 5.6: Enokoračna napoved

Kot je razvidno iz slike 5.6, predvsem iz spodnjega grafa na katerem je prikazana napaka in varianca, model, zaradi vpeljave izhoda iz prejšnjega koraka kot vhod, seveda zelo dobro opisuje proces. To je razvidno tudi iz slik 5.7, ki prikazujejo vrednosti različnih mer napak v odvisnosti od števila baznih vektorjev.

Izvedli smo tudi vrednotenje modela z »naivno« simulacijo. To pomeni, da smo kot tretji vhod, namesto vrednosti izhoda iz prejšnjega koraka, uporabili srednjo vrednost napovedi na podlagi modela, ki ga vrednotimo [7, 4]. Žal pa rezultati vrednotenja z naivno simulacijo, prikazani na sliki 5.8, izražajo slab opis modela. To je razvidno iz večih odsekov, ki popolnoma zgrešijo napoved.

Iz dobljenih rezultatov lahko sklepamo, da se izbrana metoda dobro obnese

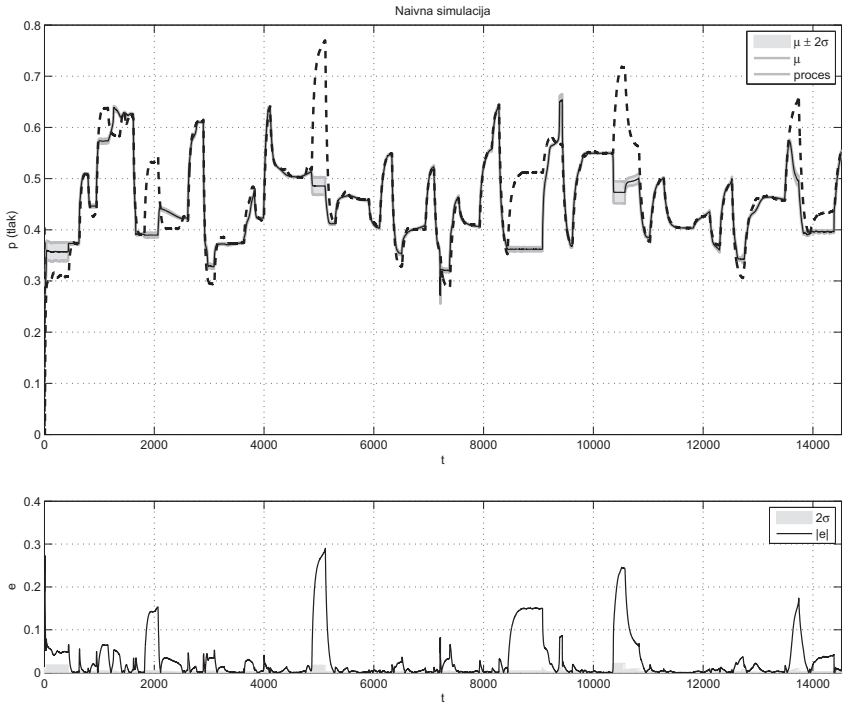


(a) MSE (b) MAE

(c) LPD (d) MRSE

Slika 5.7: Vrednosti različnih mer napak v odvisnosti od števila baznih vektorjev

pri eno-vhodnih problemih, medtem ko ima lahko pri več-vhodnih problemih težave z napovedjo. Za posplošitev te trditve pa smo naredili premalo primerov.



Slika 5.8: Vrednotenje z naivno simulacijo

Poglavje 6

Zaključek

V diplomski nalogi smo obravnavali sprotno učenje modelov na podlagi Gaussovih procesov.

GP model je verjetnostni, neparametrični model, ki napoveduje izhod v obliki Gaussove porazdelitve, kar lahko predstavimo kot najbolj verjetno vrednost napovedi in varianco kot (ne)zaupanje v to napoved. To je ena izmed glavnih prednosti pred ostalimi modeli. Vendar ima tudi slabo lastnost, in sicer veliko računsko zahtevnost, ki raste s tretjo potenco glede na število učnih primerov. Ker je zaradi te eksponentne rasti časovne odvistnosti v osnovi neprimeren za sprotno učenje, smo v diplomskem delu pregledali obstoječe metode za pohitritev tega postopka in izbrali primerno za sprotno učneje. Preučili smo veliko različnih metod, predvsem razredčevalnih, katere zmanjšajo časovno odvisnost na $O(nm^2)$ in s tem zadoščajo kriteriju sprotnega učenja, ki zahteva konstanten čas izvajanja ob vsakem koraku. Vendar le ena metoda, izmed preučenih, ustreza vsem kriterijem sprotnega učenja, torej tudi kriteriju, ki zahteva zmožnost sprotnega posodabljanja. To metodo smo tudi podrobneje opisali, ilustrirali njeno delovanje na enostavnem primeru in preizkusili njeno učinkovitost z modeliranjem procesa priprave plina, za katerega smo imeli na voljo dvakrat po 14.520 meritev. S tem smo tudi izpolnili namen tega diplomskega dela.

Na podlagi izvedenih primerov smo ugotovili, da je metoda primerna za sprotno učenje, saj s pravilno določitvijo začetnih parametrov (maksimalno število baznih vektorjev in tolerančni koeficient) lahko zagotovimo približno konstanten čas posodabljanja modela z vsakim novim vhodnim primerom. Žal pa smo prišli do zaključka, da se metoda v našem primeru bolje obnesla na eno-vhodnih sistemih kot na več-vhodnem sistemu. Poleg tega ima metoda veliko pomankljivost, ki sicer izhaja iz njene narave sprotnega učenja, namreč ne

omogoča sprotnega optimiziranja začetnih vrednosti hiperparametrov. Problem sicer lahko omilimo tako, da na podlagi predznanja ali manjšega modela ocenimo oziroma optimiziramo vrednosti hiperparametrov, vendar na tak način zelo težko pridemo do optimalnih. Zato vidimo v tej pomankljivosti zanimivo nadaljnje raziskovanje.

Dodatek A

Opis procesa priprave plina

Proces priprave plina je ena izmed enot polindustrijskega procesnega laboratorija Odseka za sisteme in vodenje na Institutu Jožef Stefan [7, 8, 19]. Procesni laboratorij je nastal ob podpori evropskega programa TEMPUS ALIAC (angl. Active Learning In Automatic Control) in je opremljen z industrijsko procesno opremo in napravami industrijskih razsežnosti. Predstavlja izvor različnih inženirskih nalog in problemov v zvezi z avtomatskim vodenjem procesov ter poligon za praktično preizkušanje različnih metod s področja procesnega vodenja. Dograjuje se glede na trenutne potrebe, pri čemer se uporablja komercialno dostopna profesionalna industrijska procesna oprema, ki omogoča razvoj in preizkus delovanja sodobnih metod avtomatskega vodenja ob upoštevanju omejitev, neidealnosti in tehničnih posebnosti, na katere naletimo v industriji.

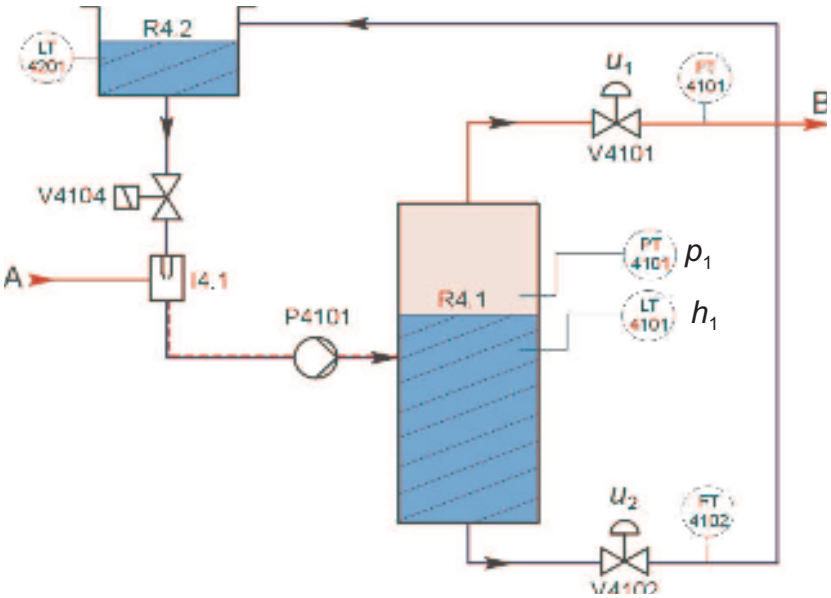
Laboratorij je sestavljen iz dveh tehnoloških sklopov – procesnih enot, ki predstavljata dva tipična industrijska procesa. Delujeta lahko samostojno ali medsebojno povezano, saj je omogočen pretok materiala, energije in informacije med njima. V našrem primeru smo se ukvarjali samo s pripravo plina.

Procesna oprema enote za pripravo plina

Procesna enota za pripravo plina vsebuje naslednjo procesno opremo, slika A.1:

- ločevalnik plina in tekočine (krajše ločevalnik) R4.1, ki je opremljen z:
  - zveznim ventilom za plin V4101,
  - merilnikom pretoka plina iz ločevalnika FT4101,
  - zveznim ventilom za vodo V4102,
  - merilnikom pretoka vode iz ločevalnika FT4102,

  - analognim merilnikom nivoja vode LT4101,
  - merilnikom tlaka plina v lo˘cevalniku PT4101,
- shranjevalna posoda R4.2 je opremljena z:
  - analognim merilnikom nivoja vode LT4201,
  - nivojskim stikalom za indikacijo maksimalnega nivoja LS4201,
- dvopoložajni ventil V4104,
- injektor I4.1,
- črpalka na vodni obroč P4101 s frekvenčnim regulatorjem.



Slika A.1: Shema procesa priprave plina

Delovanje enote za pripravo plina

Priprava plina temelji na odvzemu dimnih plinov iz dimnika peči s frekvenčno regulirano črpalko P4101. Črpalka črpa vodo iz shranjevalne posode R4.2 v tlačno posodo R4.1. Voda v injektorju iz vhoda A v cevovod povleče dimne pline, ta mešanica potuje naprej do tlačne posode, kjer se voda in plin ločita.

Zaradi prisiljenega dotoka mešanice vode in plina v posodo in regulirnih ventilov V4101 in V4102 lahko v tlačni posodi dosežemo nadtlak, ki vodo skozi izpust na dnu ločevalnika in ventil V4102 potisne nazaj v zbiralno posodo, s čimer je tokokrog vode sklenjen. Na izhodu B iz ločevalnika prek ventila V4101 v nadaljnji proces kemijske nevtralizacije pod tlakom izstopa plin. Tlak plina v zgornjem delu ločevalnika mora biti večji od hidrostatičnega tlaka med nivojem vode v shranjevalni posodi in nivojem vode v ločevalniku, da je omogočen odtok vode iz ločevalnika prek regulacijskega ventila V4102 v shranjevalno posodo. Nivo hladilne vode v ločevalniku reguliramo z zveznim ventilom V4102, tlak plina pa z zveznim ventilom V4101 ali s frekvenčno regulirano črpalko P4101. V shranjevalni posodi merimo nivo vode z analognim merilnikom nivoja LT4201. Po potrebi dotočimo vodo prek ročnega ventila, morebitno odvečno vodo pa zaznavamo z nivojskim stikalom LS4201 in jo prek preliva pretočimo v kanalizacijo. Če hladilna voda zelo dolgo kroži po zaključeni poti v sistemu, obstaja nevarnost pregretja vode [19].

Matematični opis procesa v ločevalniku plina in tekočine

Za prikaz poglavitnih relacij v procesu in za ilustracijo nelinearnosti procesa lahko proces priprave plina opišemo z dvema enačbama:

$$\begin{aligned}\frac{dp_1}{dt} &= \frac{1}{S_1(h_{T_i}-h_i)}(p_0(\alpha_0+\alpha_1 p_1+\alpha_2 p_1^2-k_1 R_1^{u_1-1}\sqrt{p_1}))+\\ &\quad +(p_0+p_1)(\Phi_w-k_2 R_2^{u_2-1}\sqrt{p_1+k_w(h_1-h_{T_2})})\\ \frac{dh_1}{dt} &= \frac{1}{S_1}(\Phi_w-k_2 R_2^{u_2-1}\sqrt{p_1+k_w(h_1-h_{T_2})}) \qquad \text{(A.1)}\end{aligned}$$

kjer je $u_i, i=1,2$, vhodni signal za ventil V410i; $h_i, i=1,2$, višina tekočine v posodi R4.i; p_1 relativni pritisk zraka v posodi R4.1; $S_i, i=1,2$, presek posode R4.i; p_0 zračni pritisk; $h_{Ti}, i=1,2$, višina posode R4.i; $R_i, i=1,2$, razmerje pretoka med zaprtim in odprtim ventilom V410i; $k_i, i=1,2$, koeficient pretoka ventila V410i; Φ_w znana konstanta vodnega toka skozi črpalko in $\alpha i, i=1,2,3$ konstante.

Iz enačb lahko sklepamo, da je opisovani nelinerni proces multivariabilen z dvema vhodoma, dvema izhodoma in navzkrižnimi povezavami. Če imamo povratnozančno regulacijo nivoja tekočine h_1, lahko ta sistem predstavimo kot univariabilen sistem z vhodom u_1 in izhodom $p1$. Iz enačb (E.2) lahko tudi razberemo, da je tlak p_1 nelinearno odvisen od nivoja h_1 in vhodnega toka, zaradi česar se proces v različnih območjih obnaša različno.

Slike

Tabele

Literatura

[1] L. Csató, M. Opper, Sparse Online Gaussian Processes, Neural computation, Vol. 14, Št. 3, str. 641-668, 2002

[2] M. N. Gibbs, Bayesian Gaussian processes for regression and classification, doktorska disertacija, Cambridge University, Cambridge, 1997.

[3] M. L. Gredilla, J. Q. Candela, A. F. Vidal, Sparse spectral Sampling, Technical Report, 2007

[4] J. Kocijan, "Identifikacija nelinearnih sistemov z Gaussovimi procesi," v Modeliranje dinamičnih sistemov z umetnimi nevronskimi mrežami in sorodnimi metodami, Univerza v Novi Gorici, 2007, str. 73-86

[5] I. Kononenko, Strojno učenje, Univerza v Ljubljani, 2005

[6] N. Lawrence, M. Seeger, R Herbrich, Fast Sparse Gaussian Process Methods: The Informative Vector Machine, Neural Information Processing Systems, Workshop on Kernel Methods, 2003

[7] B. Likar, Prediktivno vodenje nelinearnih sistemov na osnovi Gaussovih procesov, magistrsko delo, Univerza v Ljubljani, Ljubljana, september 2004.

[8] B. Likar, J. Kocijan, Predictive control of a gas-liquid separation plant based on a Gaussian process model, Computers and Chemical Engineering, Vol. 31, str. 142–152, 2007.

[9] D. J. C. MacKay, Introduction to Gaussian processes, C. M. Bishop, urednik, Neural networks and machine learning, NATO ASI Series – F 168, str. 133–166, Berlin, 1998, Springer-Verlag.

[10] D. J. C. MacKay, Information theory, inference and learning algorithms, poglavje Gaussian Processes, str. 535–548, Cambridge University Press, Cambridge, 2003.

[11] R. M. Neal: Bayesian learning for neural networks, Lecture Notes in Statistics, Vol. 118, Springer-Verlag, New York, 1996

[12] M. Opper, A Bayesian Approach to Online Learning, D. Saad (ur.), Online Learning in Neural Networks, Cambridge University Press, str. 363-378, 1998

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes, C. Cambridge University Press, Second edition, 1992

[14] C. E. Rasmussen: Evaluation of Gaussian processes and other methods for nonlinear regresion, doktorska disertacija, University of Toronto, 1996

[15] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for machine learning, The MIT Press, Cambridge, MA., 2006

[16] M. Seeger, C. Williams, N. Lawrence, Fast Forward Selection to Speed Up Sparse Gaussian Process Regression, Workshop on AI and Statistics 9, 2003.

[17] A. J. Smola, P. L. Bartlett, Sparse Greedy Gaussian Process Regression, Advances in Neural Information Processing Systems 13, str. 619-625, 2001

[18] V. Tanko, Kovariančne funkcije v modelih na podlagi Gaussovih procesov, Diplomsko delo, Univerza v Ljubljani, Ljubljana, 2009

[19] D. Vrančič, Dj. Juričič, J. Petrovčič, Measurements and mathematical modelling of a semi-industrial liquid gas separator for the purpose of a fault diagnosis, Delovno poročilo 7260, Institut Jožef Stefan, Ljubljana, 1995.

[20] G. Wahba, Spline Models for Observational Data, Society for Industrial and Applied Mathematics, Philadelphia, 1990

[21] C. K. I. Williams: Prediction with Gaussian processes: from linear regression and beyond, M. I. Jordan, (ur.), Learning in graphical models, Vol. 89 iz Nato Science Series D, Springer, Berlin, str. 599-621, 1998

[22] C. K. I. Williams, D. Barber, Bayesian classification with Gaussian Processes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, Št. 12, str. 1342-1351

[23] C. K. I. Williams, M. Seeger, Using the Nystr¨om Method to Speed Up Kernel Machines, Advances in Neural Information Processing Systems 13, str. 682-688, 2001

[24] C. K. I. Williams, C. E. Rasmussen, A. Schwaighofer, V. Tresp, Observations on the Nyst¨om Method for Gaussian Process Prediction, Technical Report, University of Edinburgh