UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ALGORITMY NA ODSTRÁNENIE ŠUMU Z DÁT

DIPLOMOVÁ PRÁCA

2018 Bc. Oleksandr KONDRATYEV

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

ALGORITMY NA ODSTRÁNENIE ŠUMU Z DÁT

DIPLOMOVÁ PRÁCA

Študijný program:	Ekonomicko-finančná matematika a modelovanie
Študijný odbor:	9.1.9. Aplikovaná matematika
Školiace pracovisko:	Katedra aplikovanej matematiky a štatistiky
Vedúci práce:	Mgr. Soňa Kilianová, PhD.

Bratislava 2018 **Bc. Oleksandr Kondratyev**

61908489

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Oleksandr Kondratyev
Študijný program:	ekonomicko-finančná matematika a modelovanie (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	aplikovaná matematika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov: Algoritmy na odstránenie šumu z dát.
Algorithms for noise cancellation from data.

Cieľ: Obsahom a cieľom práce bude prehľadne spracovať rôzne algoritmy odstraňovania šumu z dát, využívajúc metódy z rôznych oblastí matematiky (okrem iného napr. aj optimalizácia, diferenciálne rovnice). Niektoré zo študovaných algoritmov budú numericky implementované, preto bude potrebná aj programovacia a numerická zručnosť.

Vedúci:	Mgr. Soňa Kilianová, PhD.
Katedra:	FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry:	prof. RNDr. Daniel Ševčovič, CSc.
Dátum zadania:	25.01.2017

Dátum schválenia: 27.01.2017

prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

....

študent vedúci práce

Poďakovanie Týmto spôsobom by som sa rád poďakoval všetkým, ktorí akokoľvek prispeli k napísaniu danej diplomovej práce. Veľká vďaka patrí vedúcej diplomovej práce Mgr. Soni Kilianovej, PhD. za cenné rady, ochotu, usmerňovanie a odborné vedenie pri napísaní práce. Ďakujem aj mojej rodine za každodennú podporu, inšpiráciu a trpezlivosť.

Abstrakt v štátnom jazyku

KONDRATYEV, Oleksandr: Algoritmy na odstránenie šumu z dát [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: Mgr. Soňa Kilianová, PhD., Bratislava, 2018, 80s.

V súčasnej dobe analýza dát predstavuje podstatu rôznych oblastí podnikania, vedy a spoločenských vied. Napr. finančné dáta majú vysokú frekvenciu, sú charakteristické svojou komplexnou štruktúrou s množstvom skokov kvôli veľkému počtu zmien na trhu. Pri analýze týchto dát veľmi podstatným je sledovať trend, ktorý je výstupom filtrovania od šumu. V predkladanej práci sa venujeme rôznym algoritmom odstránenia šumu z dát použitím vedomostí z rôznych matematických odborov. V tejto práci sme sa venovali rôznym lineárnym a nelineárnym filtrom, ich porovnaniu a taktiež sme uviedli výhody a nevýhody každej z metód. Venovali sme sa metóde regularizačnej aproximácie, ktorá poukazuje na možnosť použitia konvexnej optimalizacie pri rekonštrukcii signálu. Metódy sme naprogramovali použitím softvéru *R* a *MATLAB*, čo nám pomohlo preskúmať a následne aj porovnať vlastností spomínaných metód. Vyššie uvedené softvéry nám umožnili spraviť experiment, v ktorom sme skúmali kvalitu vyhladenia niektorých metód v závislosti od množstva impulzného šumu prítomného v experimentálnych signáloch. Taktiež sme pozorovali hodnoty parametrov daných metód, pri ktorých bolo dosiahnuté optimálne vyhladenie. Pre každý s použitých signálov sme zistili charakter závislosti kvality vyhladenia a hodnôt parametrov použitých metód od množstva impulzov prítomného v dátach.

Kľúčové slová: : impulzný šum, odstránenie šumu, lineárne filtre, nelineárne filtre, regularizačná aproximácia

Abstract

KONDRATYEV, Oleksandr: Algorithms for noise cancellation from data [Master Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: Mgr. Soňa Kilianová, PhD., Bratislava, 2018, 80p.

Nowadays data analysis is fundamental for various areas in business, science and social science. For example, financial data usually have a high frequency, in addition they are characterized by complex structure with a number of leaps due to a large number of changes on stock exchange. It is very important to follow the trend, which is the output of filtering, while analyzing these data. This paper carries out various algorithms for the elimination noise in data using knowledges from several mathematical disciplines. In this work, we have dealt with different linear and nonlinear filters, their comparison. Furthermore, we have presented the advantages and disadvantages for each of the methods. One of the parts is dedicated to the method of regularized approximation which points out to the possibility of using convex optimization in the reconstruction of signals. We have been using the *R* and *MATLAB* software in our reasearch, which has helped us to examine and then to compare the properties of the methods mentioned above. This software allowed us to do the experiment with the aim to study quality of smoothing of some methods according to the amount of impulsive noise that is presented in the experimental signals. We also observed values of the parameters of the methods where the optimal smoothing was achieved. For type of signal, we determined the nature of quality of smoothing and the values of the parameters for methods we used accotding to the amount of impulsive noise presented in the data.

Keywords: impulsive noise, de-noising, linear filters, nonlinear filters, regularization approximation

Obsah

Úvod

Súčasný technický pokrok umožňuje veľkým spoločnostiam a organizáciám okamžite zbierať obrovské množstvo dát, napríklad „tick-by-tick“ dáta z finančných trhov. Finančné dáta majú komplexnú štruktúru s veľkým množstvom skokov spôsobených veľkým počtom okamžitých zmien na trhu, a teda majú vysokú frekvenciu. Pri analýze dát tohto typu je často veľmi vhodné pozrieť sa na trend, ktorý je výstupom filtrovania od šumu. Dôležitou úlohou je teda rozlíšiť trend a stochastické výkyvy.

Neexistuje všeobecná definícia trendu, ktorá by bola vhodná na všetky aplikácie v rôznych oblastiach. Všeobecne sa uznáva, že trend je pomaly sa meniaca zložka, ktorá je hlavnou rušivou silou dlhodobého rozvoja systému. V literatúre sa častokrát stretávame s myšlienkou, že trend je obmedzený na určité nízke frekvencie údajov. Táto koncepcia vylučuje akékoľvek signifikantné vplyvy a výkyvy vyšších frekvencií, avšak nie je uspokojivá pre mnohé finančné dáta, s ktorými sa bežne stretávame v praxi. Čiže dôležitou úlohou je nielen do istej miery odstrániť stochastický šum, ale aj zachovať podstatnú štruktúru dát.

Na riešenie tohto problému existujú rôzne klasické a novovytvorené metódy odstraňovania šumu z dát, takzvané filtre. Dôležitou otázkou je, ktorý algoritmus treba zvoliť na to, aby sme dostali najvhodnejšie riešenie pre konkrétne dáta. Hovoríme o rôznych prístupoch a ich príslušnej klasifikácii. Na základe tejto klasifikácie sa budeme zameriavať nielen na lineárne filtre, ale aj na iné komplexnejšie metódy a skúmať ich výhody a nevýhody.

V prvej kapitole sa budeme konkrétne zaoberať lineárnymi filtrami a ich správaním pri odhadovaní trendu finančných dát. Preberieme vlastnosti a vplyv rôznych parametrov na výstup algoritmov, ktorými sú moving average filtre, kernel metódy a metóda exponenciálneho vyhladenia.

V ďalšej časti tejto práce popíšeme nelineárne filtre a ich výhody oproti lineárnym metódam. Prostredníctvom softvéru *R* na ukážkových signáloch porovnáme správania mean a mediánového filtra (ktoré sú základnými predstaviteľmi týchto dvoch skupín) nielen v súvislosti s gaussovským, ale aj s impulzným šumom. Okrem mediánového filtra preberieme vlastnosti trimmed mean filtrov, L-filtrov, C-filtrov a váženého mediánového filtra.

V tretej kapitole predstavíme základné pojmy teórie regularizačnej aproximácie, ktorá je súčasťou metód konvexnej optimalizácie. Jednou z možných aplikácií danej metódy je jej použitie pri rekonštrukcii zašumených signálov. Konkrétne sa budeme zaoberať spôsobom aplikovania a porovnaním vlastností metódy kvadratického vyhladenia, rekonštrukcie totálnou variáciou a Hodrick-Prescottovho filtra.

V poslednej časti tejto práce popíšeme nami navrhnutý experiment, v ktorom porovnáme správanie niektorých z prebratých metód na experimentálnych signáloch rôzneho typu a rôznej intenzity impulzného šumu. V experimente budeme sledovať optimálnu mieru vyhladenia zašumených signálov použitím rôznych metód. Za mieru vyhladenia budeme považovať strednú kvadratickú odchýlku očisteného signálu od jeho pôvodnej nezašumenej verzie.

Cieľom predkladanej diplomovej práce bolo prehľadne spracovať rôzne algoritmy odstraňovania šumu z dát, využívajúc metódy z rôznych oblastí matematiky a taktiež preskúmať výhody a nevýhody daných metód na rôznych dátach prostredníctvom softvéru *R* a *MATLAB*.

1 Lineárne filtre

Finančné dáta si vieme predstaviť ako časový rad, ktorý je zadefinovaný nasledovne:

$$Y_t = f_t + s_t + X_t,$$

kde f_t je pomaly sa meniaca funkcia, ktora reprezentuje trend, s_t je periodická funkcia, reprezentujúca napríklad sezónnu zložku, X_t je stochastická zložka, o ktorej predpokladáme, že je stacionárna. Pre jednoduchosť sme uvažovali model bez sezónnej zložky, čiže model tvaru:

$$Y_t = f_t + X_t.$$

Pri takejto formulácii za šum považujeme stochastickú zložku X_t, ktorú chceme zredukovať. Existujú rôzne metódy, ktoré umožňujú dáta očistiť od šumu a zároveň zachovať dôležité informácie.

V tejto kapitole sa zaoberáme neparametrickou metódou odstraňovania šumu z dát, a to lineárnymi filtrami. Algoritmy používajúce lineárne filtre sú najznámejšie a najviac používané metódy na odstraňovanie šumu z dát a získanie trendu. Lineárne filtre produkujú hladký trend, ktorý ale nezachováva relevantné ostré hrany, čo je jednou z nevýhod takého filtrovania [1].

Najviac používaná formulácia lineárnych filtrov je nasledovná [1]:

$$\widehat{f}_t = \sum_{i=-h}^{h} w_i Y_{t+i}, \tag{1}$$

kde t je časový bod, pre ktorý sa ráta výstup filtru. Filter naformulovaný pomocou (1) je filtrom dĺžky $(2h + 1)$, lebo pre každú hodnotu Y_t vytvára filtrovaný výstup $\widehat{f}_t$ použitím $(2h + 1)$ dát. Zložky w_i sú váhy, s ktorými pôvodné dáta vstupujú do filtra. Tým pádom pre každú hodnotu Y_t výstup filtrovania $\widehat{f}_t$ je váženým súčtom dát v okolí t.

Váhy w_i sa dajú naformulovať rôznym spôsobom, od čoho bude závisieť správanie a vlastnosti lineárneho filtra. Výberom vhodných váh vieme dostať rôzne typy filtrov, akými sú napríklad [2]:

- *filtre dolnej priepuste (low-pass filter)* - filtre, cez ktoré prechádzajú frekvencie, ktoré sú nižšie než určitá hodnota. Vysoké frekvencie sa pod vplyvom tohto filtra zoslabujú alebo odstraňujú.

- *filtre vysokej priepuste (high-pass filter)* - filtre, cez ktoré prechádzajú frekvencie, ktoré sú vyššie než určitá hodnota.

- *filtre pásmovej priepuste (band-pass filter)* - odstraňuje zložky signálu, ktorých frekvencia je vyššia alebo nižšia než stanovené hranice.

Ďalším dôležitým parametrom, ktorý vstupuje do formulácie lineárneho filtra, je parameter h (*bandwidth*), ktorý predstavuje šírku pásma priepuste. Tento parameter určuje dĺžku intervalu $L = (2h+1)$, čiže rozsah dát, ktoré v každom kroku vstupujú do váženého súčtu pri vytvorení výstupu $\widehat{f}_t$. Zmenou tohto parametra sa dá ovplyvniť charakter výstupu filtrov, čo ukážeme v nasledujúcich častiach tejto práce.

1.1 Moving average

V tejto časti sme sa zaoberali najznámejším lineárnym filtrom, ktorým je *mean filter*, alebo *moving average filter*.

Moving average filter je štandardným nástrojom používaným vo financiách na odstraňovanie šumu z „tick-by-tick" dát. Tento algoritmus patrí do skupiny filtrov dolnej priepuste, čiže odstraňuje krátkodobé výkyvy a ponecháva dlhodobý trend. Ako výsledok vzniká hladký signál.

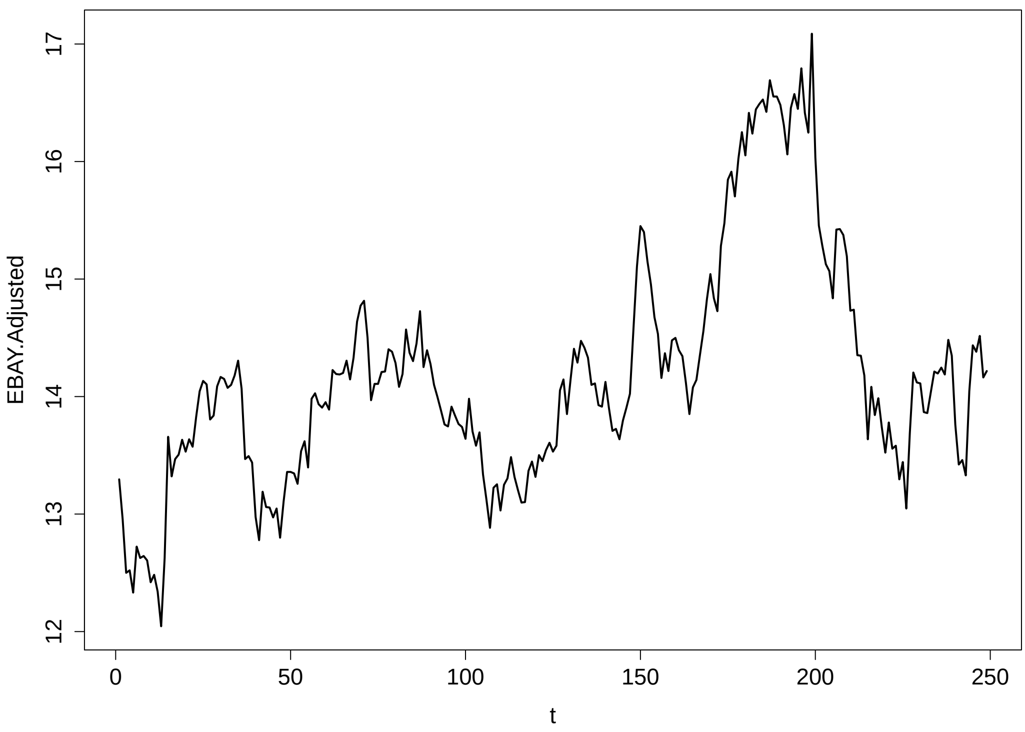
Vychádzajúc z formulácie (1), pri voľbe $w_i = \frac{1}{(2h+1)}$ dostávame vyššie spomenutý mean filter. To znamená, že všetky váhy používané pri filtrovaní sú rovnaké a trendová zložka f_t sa odhaduje nasledovne [2]:

$$\widehat{f}_t = (2h+1)^{-1} \sum_{i=-h}^{h} Y_{t+i}. \tag{2}$$

Zo vzorca (2) je vidieť, že odhadom bude lokálny priemer hodnôt intervalu dĺžky $L = (2h+1)$. S každým krokom sa mení stred intervalu t a počíta sa nová hodnota odhadu $\widehat{f}_t$. Prejdením cez celý časový rad sa získava moving avarage krivka, ktorá je dobrým odhadom trendu daného časového rádu, lebo je očistená od stochastického šumu.

Na detailnejšiu analýzu správania tohto filtra sme použili softvér R a dáta, ktoré sú súčasťou balíka *quantmod*.

Pri analýze sme zvolili časový rad, ktorý pozostáva z denných cien akcií *Ebay.Adjusted* v období 04.01.2007-31.12.2007. Tieto dáta sú zobrazené na Obr.1.



Obr. 1: Dáta *Ebay.Adjusted* v období 04.01.2007-31.12.2007. Zdroj: softvér R, balík *quantmod*.

Z Obr.1 je vidieť, že denné dáta majú stochastické výkyvy, čiže obsahujú zložky s vysokými frekvenciami. Vhodným nástrojom na identifikáciu trendu môže byť mean filter, ktorý sme naprogramovali pomocou softvéru R a následne použili na daný časový rad. Použitý kód vyzerá nasledovne:

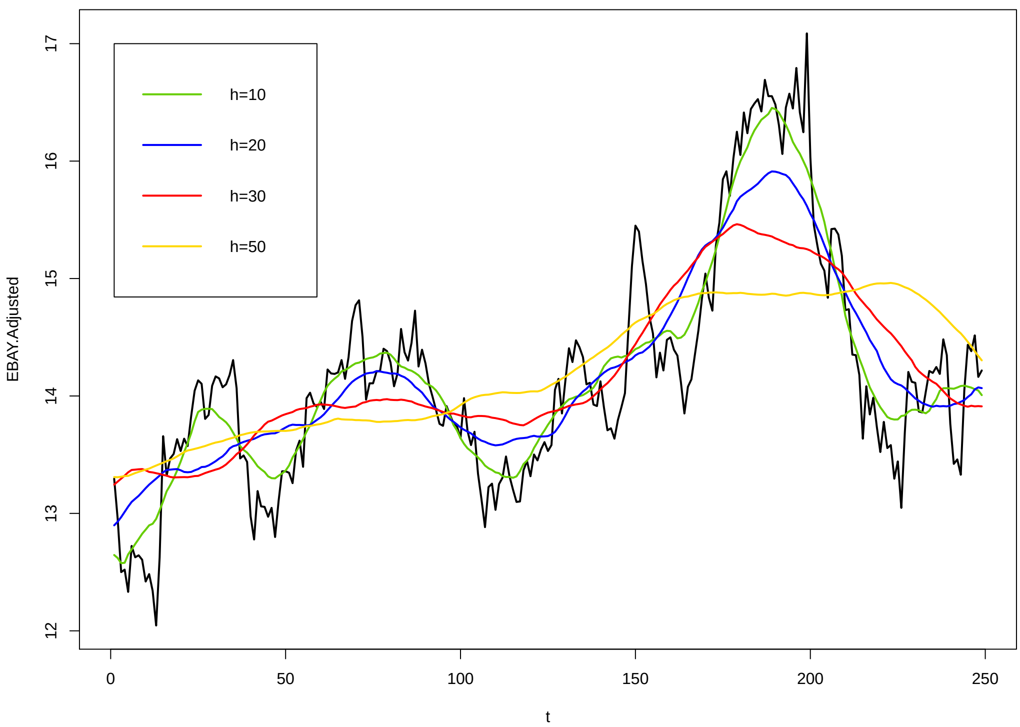
```
h<-20
average<-y
lkraj<-0
for (i in 1:(h+1)){
average[i]<-sum(y[(i-lkraj):(i+h)])/length(y[(i-lkraj):(i+h)])
lkraj<-i }
for(i in (h+1):(length(y)-h-1)) {
average[i]<-sum(y[(i-h):(i+h)])/length(y[(i-h):(i+h)]) }
rkraj<-h
for (i in (length(y)-h):length(y)){
average[i]<-sum(y[(i-h):(i+rkraj)])/length(y[(i-h):(i+rkraj)])
```

```
rkraj<-rkraj-1 }
```

kde h je parameter určujúci dĺžku intervalu použitého na priemerovanie a y je časový rad, ktorého trend chceme zistiť.

Pri aplikovaní moving average filtra nastáva problém na okrajoch časového radu. Podľa dohody sa zvyknú ignorovať dáta, ktoré sú mimo sledovaného časového intervalu. Tak napríklad, hodnota f_2 sa odhaduje pomocou priemeru hodnôt $Y_1, \ldots, Y_{(2+h)}$, čiže zoberie sa viac dát napravo od časového bodu *2*, než naľavo od tohto bodu. Nami použitý kód zahŕňa túto vlastnosť metódy. Tento hraničný efekt je výraznejší, keď trend na hranici je strmý a dĺžka intervalu je veľká, čo vedie k určitým problémom pri odhadovaní trendu na hraniciach časového radu [2].

Ako už bolo spomenuté vyššie, parametrom h vieme meniť správanie lineárneho filtra. Na Obr.2 sme uviedli trendy, ktoré sú výsledkami aplikovania filtra na časový rad cien akcií *Ebay.Adjusted* pri rôznych hodnotách parametra h. Zelená krivka zodpovedá trendu pri



Obr. 2: Trendy odhadnuté pomocou moving average filtra pri rôznych hodnotách parametra h. Zelená krivka zodpovedá trendu pri hodnote parametra h=*10*, modrá pre h=*20*, červená pre h=*30*, žltá pre h=*50*.

hodnote parametra *h=10*, modrá pre *h=20*, červená pre *h=30* a žltá zodpovedá extrémnej hodnote *h=50*.

Z Obr.2 sa dá usúdiť, že pri zvyšovaní hodnoty parametra h dostávame krivku, ktorá je menej ovplyvnená pôvodnými dátami, čiže menej popisuje skutočný trend. Napríklad zelená krivka, ktorá zodpovedá hodnote *h=10*, zahŕňa v sebe informácie o väčšine výkyvov pôvodných dát, kým žltá krivka (*h=50*) je príliš vyhladená a stráca informácie o dôležitých zmenách, ktoré patria do trendu. Vizuálne optimálnou voľbou je v danom prípade hodnota parametra *h=20* (modrá krivka), hoci odhad trendu na okrajoch časového rádu je skreslený, čo je dôsledkom problému, ktorý už bol popísaný vyššie.

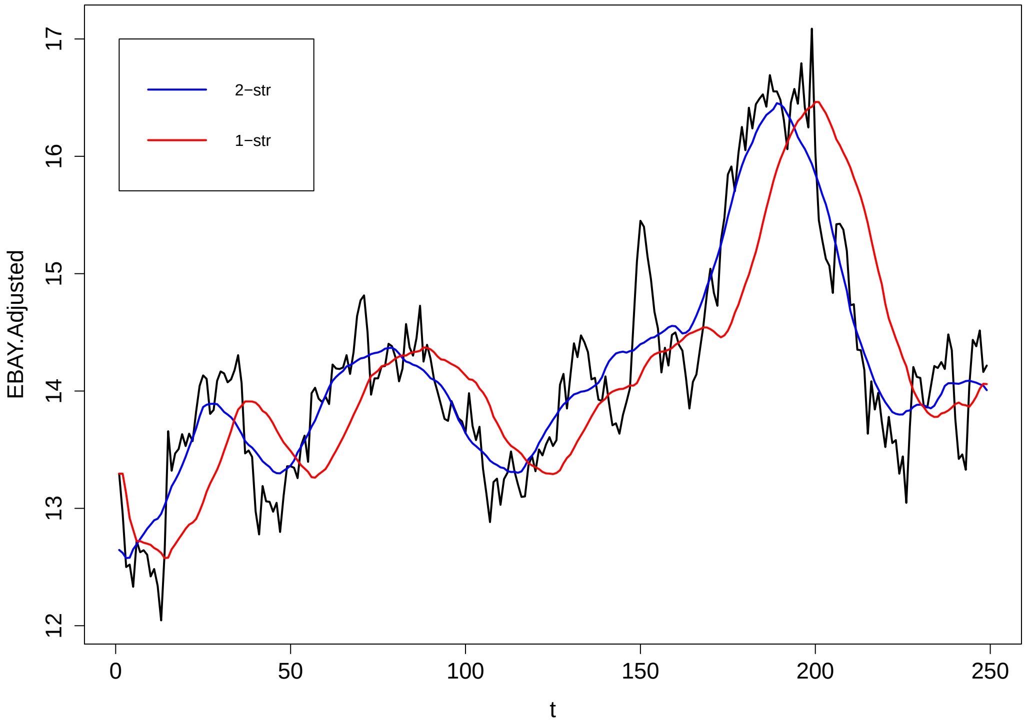
Moving average filter je obojstrannou metódou, lebo pri priemerovaní sa používajú dáta z oboch strán od časového bodu t. Niekedy pri analýze trendu a taktiež na jednoduchú predikciu sa používa takzvaný *jednostranný moving average filter* nasledovného tvaru [2]:

$$\widehat{f}_t = h^{-1} \sum_{i=1}^{h} Y_{t-i}. \tag{3}$$

Filter naformulovaný podľa (3) sme taktiež naprogramovali a dostali sme nasledovný kód:

```
h<-20   #bandwidth
lkraj<-1
jednostranny<-y
for (i in 2:(h+1)){
jednostranny[i]<-sum(y[(i-lkraj):(i-1)])/length(y[(i-lkraj):(i-1)])
lkraj<-i }
for(i in (h+2):length(y)) {
jednostranny[i]<-sum(y[(i-h):(i-1)])/length(y[(i-h):(i-1)]) }
```

Algoritmus sme aplikovali na pôvodné dáta *Ebay.Adjusted* a získaný trend sme porovnali s výstupom obojstranného mean filtra. Porovnanie je uvedené na Obr.3. Modrá krivka zodpovedá vyfiltrovanému trendu použitím obojstrannej metódy s parametrom *h=20*, červená je výsledkom filtrovania pomocou jednostranného moving average algoritmu s parametrom *h=40*. Z obrázku vidíme, že jednostranná metóda vo výsledku dáva trend podobný trendu získaného pomocou obojstrannej metódy, ale posunutý doprava od skutočného. Dôvodom takého skresľovania je to, že v každom kroku je hodnota $\widehat{f}_t$ ovplyvnená dátami iba vľavo



Obr. 3: Výstup obojstranného vs. výstup jednostranného moving average filtra. Modrá krivka zodpovedá vyfiltrovanému trendu použitím obojstrannej metódy s parametrom h=*20*, červená zodpovedá vyfiltrovanému trendu použitím jednostrannej metódy s parametrom h=*40*.

od časového bodu t. Problém môže byť odstranený pomocou lokálneho lineárneho vyhladenia alebo iných metód popísaných v zdrojoch [3] a [4]. Z tohto hľadiska na odhadovanie trendu sa vždy odporúča používať obojstranné metódy.

1.2 Kernel metódy

V predošlej časti sme popísali najznámejšie lineárne filtre, ale všeobecnejší pohľad je daný formuláciou takzvaných *kernel filtrov*. Pri týchto metódach sa váhy môžu formulovať dvomi spôsobmi v závislosti od počtu hodnôt vstupujúcich do filtra pri odhadovaní trendu v danom časovom kroku. Jednou možnosťou je použitie celého časového radu na odhadovanie trendovej zložky v čase $t_0 \in \{1, \ldots, T\}$, kde T je koncový čas radu. Formulácia váh je v tomto prípade nasledovná [2]:

$$w_t = \frac{K(\frac{t-t_0}{h})}{\sum\limits_{t=1}^{T} K(\frac{t-t_0}{h})},$$

kde $K(u)$ je *kernel funkcia.* Potom *kernel filter* nadobúda nasledovný tvar [2]:

$$\widehat{f_{t_0}} = \frac{\sum\limits_{t=1}^{T} K(\frac{t-t_0}{h})Y_t}{\sum\limits_{t=1}^{T} K(\frac{t-t_0}{h})}, \tag{4}$$

kde t_0 je časový bod, pre ktorý sa ráta odhad trendu. Z takto zvolenej formulácie je vidno, že pre odhad $\widehat{f_{t_0}}$ sa používajú dáta z celého časového radu.

Inou možnosťou je použiť prístup podobný ako pri moving average filtri, čiže na odhadovanie trendu v každom kroku sa použije vážený priemer hodnôt intervalu dĺžky $L = (2h+1)$. V tomto prípade sa váhy pridelené hodnotám z tohto intervalu formulujú nasledovne [1]:

$$w_i = \frac{K(\frac{i}{h})}{\sum\limits_{j=-h}^{h} K(\frac{j}{h})}, i = -h, ..., h,$$

čiže kernel funkcia $K(u)$ má v tomto prípade definičný obor $[-1, 1]$. Uvažujúc túto formuláciu, kernel filter nadobúda nasledovný tvar:

$$\widehat{f_t} = \frac{\sum\limits_{i=-h}^{h} K(\frac{i}{h})Y_{t+i}}{\sum\limits_{j=-h}^{h} K(\frac{j}{h})}. \tag{5}$$

Takto naformulované filtre sú vylepšenou verziou moving average filtra, spomínaného v predošlej časti a ktorého váhy boli rovnomerne rozdelené medzi všetkými dátami. Kernel filtre umožňujú prideliť dátam, ktoré sú blízko sledovaného časového bodu, väčšiu váhu. Váhy sú závislé od voľby kernel funkcie K. Bežne sa používajú funkcie ako napríklad *Epanechnikovov kernel* [1]:

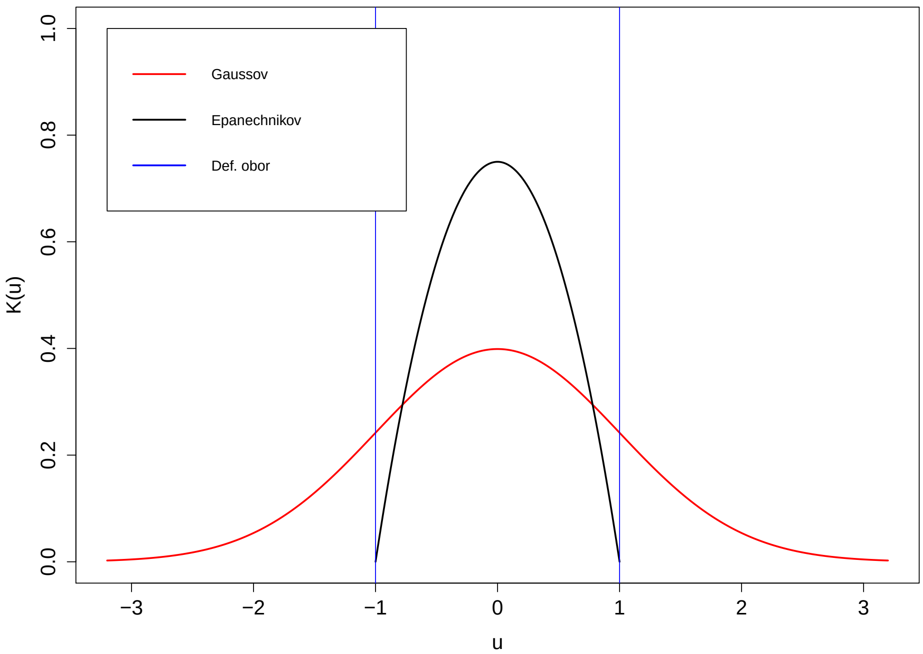
$$K^E(u) = \tfrac{3}{4}(1-u^2)^+,$$

ktorý sa používa vo filtri (5), alebo *Gaussov kernel* [1]:

$$K^G(u) = \tfrac{1}{\sqrt{2\pi}} exp(\tfrac{-u^2}{2}),$$

ktorý sa dá vhodne použiť pri filtri naformulovanom ako (4).

Na Obr.4 sú uvedené grafy funkcií K^E a K^G. Červená krivka zobrazuje Gaussov kernel, čierna Epanechnikovov kernel. Modré zvislé čiary zodpovedajú definičnému oboru $[-1, 1]$, štandardne používanému pri niektorých metódach.



Obr. 4: Grafy Epanechnikovovej a Gaussovej kernel funkcie spolu so štandardným definičným oborom. Červená krivka zobrazuje funkciu $K^G(u)$, čierna $K^E(u)$.

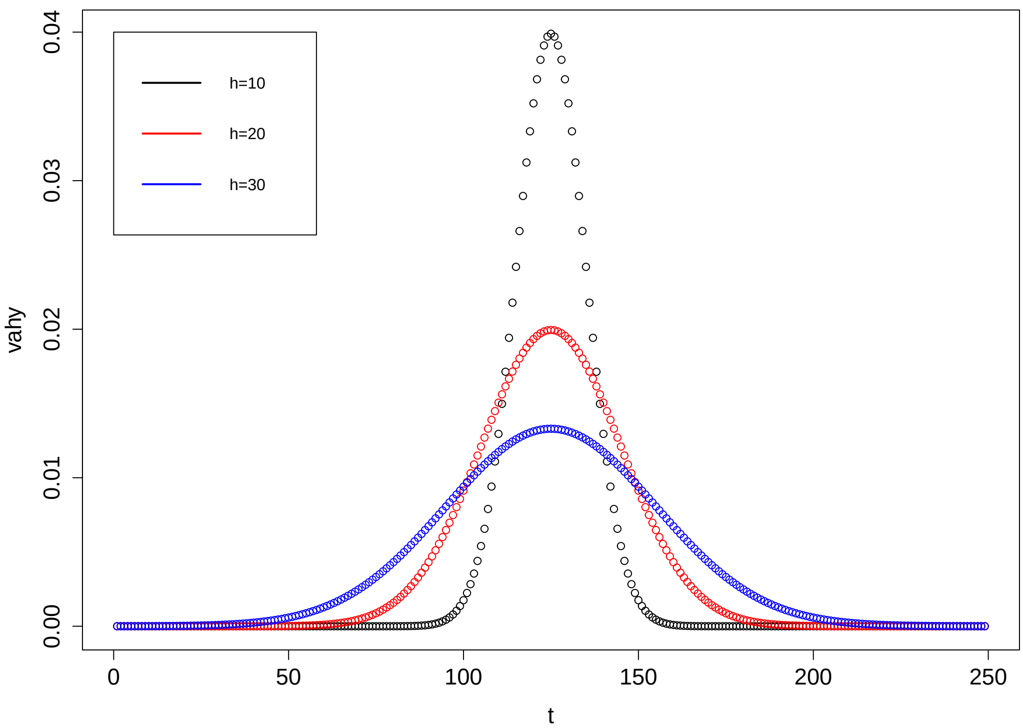
Z dôvodu, že pri Gaussovom filtri sa v každom kroku používa celý časový rad, parameter h bude predstavovať dĺžku *intervalu signifikantných váh*. Táto vlastnosť sa dá lepšie pochopiť z grafického zobrazenia váh. Na Obr.5 sme uviedli váhy w_t, pri voľbe $t_0 = 125$, čo je stred časového intervalu $[0, 250]$. Čierne krúžky zodpovedajú váham pri voľbe parametra $h = 10$, červené $h = 20$, modré $h = 30$. Čím je tento parameter väčší, tým viac dát dostane signifikantné váhy pri výpočte odhadu $\widehat{f_{t_0}}$. Tak napríklad, pri voľbe $h = 10$ váhy pridelené bodom za hranicami intervalu $[100, 150]$, budú skoro nulové, ale pri iných uvažovaných hodnotách h je interval signifikantnosti zjavne širší.

Iným parametrom, ktorým sa dá ovplyvniť správanie Gaussovho filtra, je štandardná odchýlka rozdelenia $N(0, \sigma)$, ktorá tento filter definuje (K^G je definované so $\sigma = 1$). Pomocou tohto parametra sa škáluje príslušná kernel funkcia.

Ak sa zoberie do úvahy aj tento parameter, tak formulácia Gaussovho filtra bude nasledovná:

$$K_\sigma^G(u) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-u^2}{2\sigma^2}).$$

Obr.6 zobrazuje váhy zodpovedajúce parametru $h = 10$ a rôznym hodnotám parametra



Obr. 5: Závislosť váh vypočítaných pomocou Gaussovho kernelu od voľby parametra h. Čierne krúžky zobrazujú váhy pri voľbe parametra $h = 10$, červené $h = 20$, modré $h = 30$.

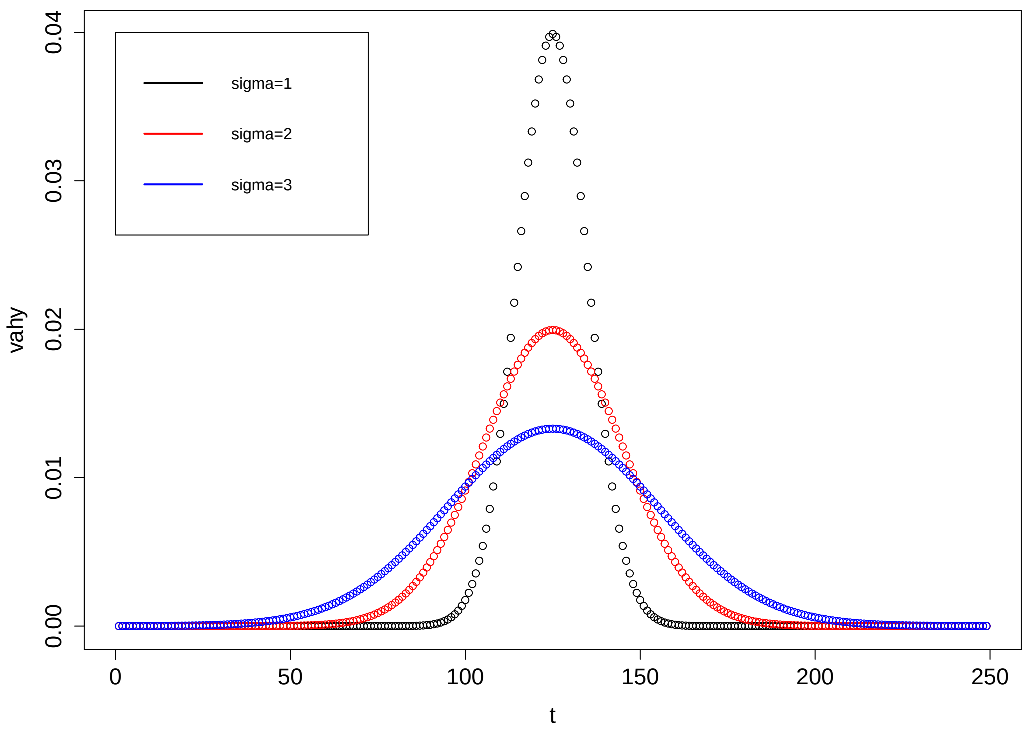
σ. Váhy vypočítane pomocou Gaussovho kernelu s voľbou $\sigma = 1$ sú znázornené čiernymi krúžkami, červené krúžky zodpovedajú váham pri voľbe $\sigma = 2$, modré $\sigma = 3$. Z Obr.6 sa dá usúdiť, že pri zvyšovaní parametra škálovania σ sa zväčšuje aj interval signifikantnosti váh, čiže to má podobný účinok ako zvyšovanie parametra h.

Podobné úvahy o parametri škálovania sa dajú spraviť aj pre Epanechnikovov kernel, hoci v literatúre sme takéto úvahy nenašli. Zavedieme preto parameter škálovania a, pričom, daný kernel potom nadobúda nasledovný tvar:

$$K_a^E(u) = \tfrac{3}{4}(1 - au^2)^+.$$

Zmenou tohto parametra sa mení tvar výslednej paraboly. V limitnom prípade, keď sa hodnota a blíži k 0, dostávame konštantnú funkciu, čiže:

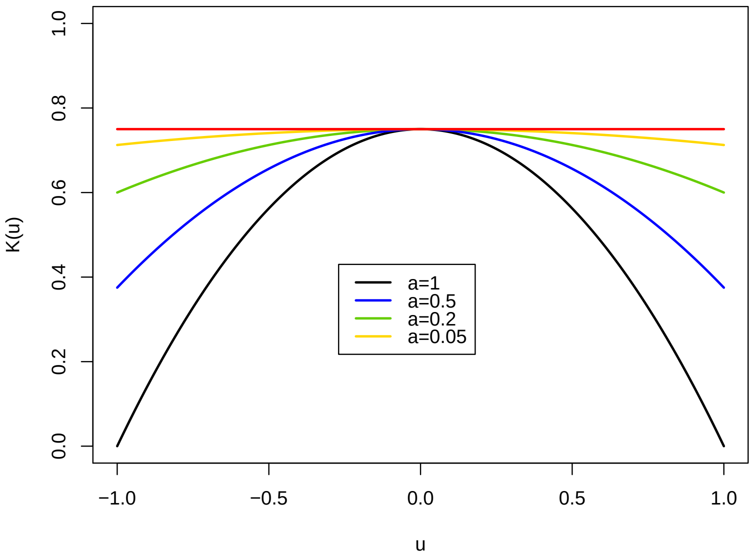
$$\lim_{a \to 0} \frac{3}{4}(1 - au^2)^+ = \frac{3}{4}.$$



Obr. 6: Závislosť váh vypočítaných pomocou Gaussovho kernelu od voľby parametra σ. Čierne krúžky zobrazujú váhy vypočítane pomocou Gaussovho kernelu s voľbou $\sigma = 1$, červené krúžky $\sigma = 2$, modré $\sigma = 3$.

Na Obr.7 sme uviedli grafy kernelov $K_a^E(u)$ pre tri rôzne hodnoty parametra škálovania a taktiež aj limitný prípad, ku ktorému sa dané paraboly blížia pri zmenšovaní hodnoty a. Červená krivka na Obr.7 zodpovedá limitnej funkcii $K(u) = \frac{3}{4}$, čierna parabola je pôvodný Epanechnikovov kernel, čiže parameter a je v tomto prípade rovný 1, modrá parabola zodpovedá hodnote $a = 0.5$, zelená $a = 0.2$, žltá $a = 0.05$.

Ukázali sme, že zmenou parametra škálovania sa mení tvar kernel funcie, čiže sa budú meniť aj váhy priradené dátam pri používaní Epanechnikovho filtra a to tak, že pri zmenšovaní parametra škálovania sa váhy budú približovať k váham používaným pri moving average filtri. Túto vlastnosť sme naformulovali do vety, ktorá znie nasledovne:



Obr. 7: Grafy Epanechnikovových kernelov pre rôzne hodnoty parametra škálovania a limitný prípad. Červeným je zobrazená limitná funkcia $K(u) = \frac{3}{4}$, čierna parabola zobrazuje Epanechnikovov kernel modrá parabola zodpovedá hodnote $a = 0.5$, zelená $a = 0.2$, žlta $a = 0.05$.

Veta 1.1 (O prechode od Epanechnikovho kernelu k moving average filtru)**.** *Nech* $w_i = \frac{K(\frac{i}{h})}{\sum\limits_{j=-h}^{h} K(\frac{j}{h})}$, $i = -h, \dots, h$ *sú váhy, ktoré sa priraďujú dátam vstupujúcim do Epanechnikovho filtra v danom (ľubovoľnom) časovom kroku.*
a) Ak $K(u) = \frac{3}{4}(1 - au^2)^+$ *je Epanechnikovov kernel s parametrom škálovania a, potom* $\lim_{a\to 0} w_i = \frac{1}{2h+1}$, *čo zodpovedá váham používaným pri moving average filtri.*
b) Ak $K(u) = \frac{3}{4}(a - u^2)^+$ *je Epanechnikovov kernel s parametrom škálovania a, potom* $\lim_{a\to\infty} w_i = \frac{1}{2h+1}$, *čo zodpovedá váham používaným pri moving average filtri.*

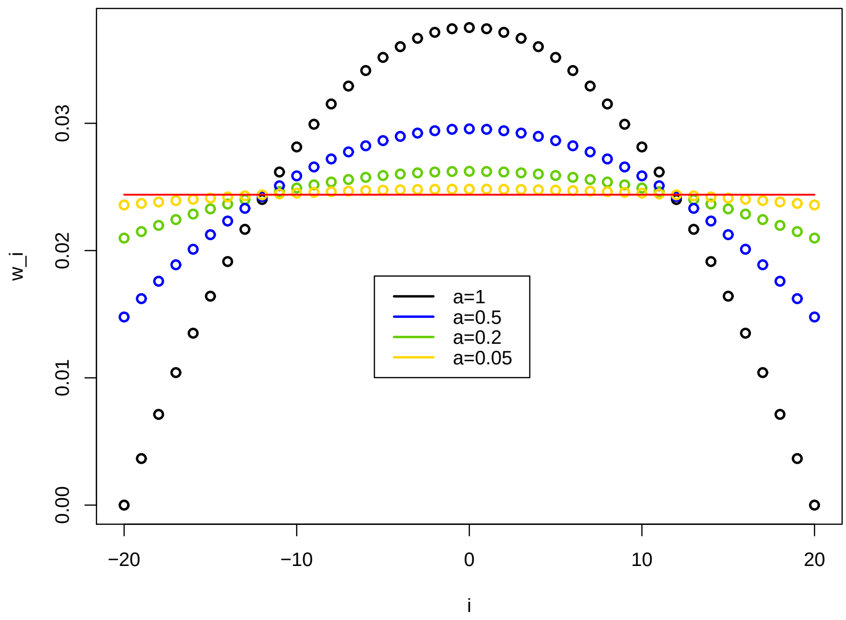
Dôkaz. Priamym výpočtom limity váh dostávame:

$$a)\lim_{a\to 0} w_i = \lim_{a\to 0} \frac{K(\frac{i}{h})}{\sum\limits_{j=-h}^{h} K(\frac{j}{h})} = \lim_{a\to 0} \frac{1 - a(\frac{i}{h})^2}{\sum\limits_{j=-h}^{h} (1 - a(\frac{j}{h})^2)} = \frac{1}{\sum\limits_{j=-h}^{h} 1} = \frac{1}{(2h+1)}$$

$$b)\lim_{a\to\infty} w_i = \lim_{a\to\infty} \frac{K(\frac{i}{h})}{\sum\limits_{j=-h}^{h} K(\frac{j}{h})} = \lim_{a\to\infty} \frac{a - (\frac{i}{h})^2}{\sum\limits_{j=-h}^{h} (a - (\frac{j}{h})^2)} = \lim_{a\to\infty} \frac{1 - \frac{i^2}{h^2 a}}{\sum\limits_{j=-h}^{h} 1 - \frac{j^2}{h^2 a}} = \frac{1}{\sum\limits_{j=-h}^{h} 1} = \frac{1}{(2h+1)}$$

$\square$

Na lepšie pochopenie Vety 1.1 a) môže slúžiť grafické znázornenie váh vypočítaných pre rôzne hodnoty parametra a, ako ilustrujeme na Obr. 8. Zvolili sme parameter $h = 20$, čiže váhy sa prideľujú $L = (2h + 1) = 41$ hodnotám. Červená čiara na danom obrázku



Obr. 8: Váhy vypočítané pomocou Epanechnikov kernelu pre $h = 20$ a rôzne hodnoty parametra škálovania a. Červená čiara zodpovedá limitným konštantným váham $\frac{1}{(2h+1)} = \frac{1}{41} \approx 0.0244$, čierne krúžky zodpovedajú váham pri hodnote parametra $a = 1$, modré pri $a = 0.5$, zelené $a = 0.2$, žlté $a = 0.05$.

zodpovedá limitným konštantným váham, ktoré sa v tomto prípade rovnajú $\frac{1}{(2h+1)} = \frac{1}{41} \approx 0.0244$, čierne krúžky sú váhy vypočítané pri hodnote parametra $a = 1$, modré pri $a = 0.5$, zelené $a = 0.2$, žlté $a = 0.05$. Môžeme vidieť, že čím menšia je hodnota parametra a, tým rovnomernejšie je rozdelenie váh medzi dátami vo vybranom intervale, čo v limitnom prípade vedie k situácii, keď všetky váhy sú rovnaké.

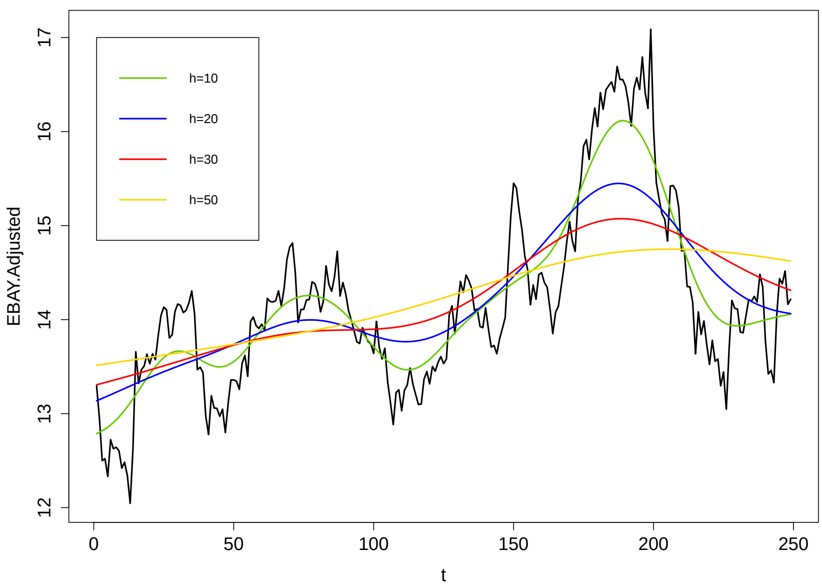
Fungovanie a následné porovnanie kernel filtrov s algoritmami popísanými v predošlých častiach sme spravili na dátach *Ebay.Adjusted*. Použitím kódu, ktorý sme naprogramovali v softvéri R, je možné vyfiltrovať trend pomocou Gaussovho kernelu. Tento kód vyzerá nasledovne:

```
h<-10
```

```
sigma<-1
normal<-y
for (i in 1:length(y)){
u<-(t-i)/h
vahy<-exp((-u^2)/(2*sigma^2))
vahy<-vahy/sum(vahy)
normal[i]<-sum(vahy*y)}
```

kde parametre h a σ sa dajú podľa potreby meniť, čiže kernel funkcia sa dá ľubovoľne škálovať.

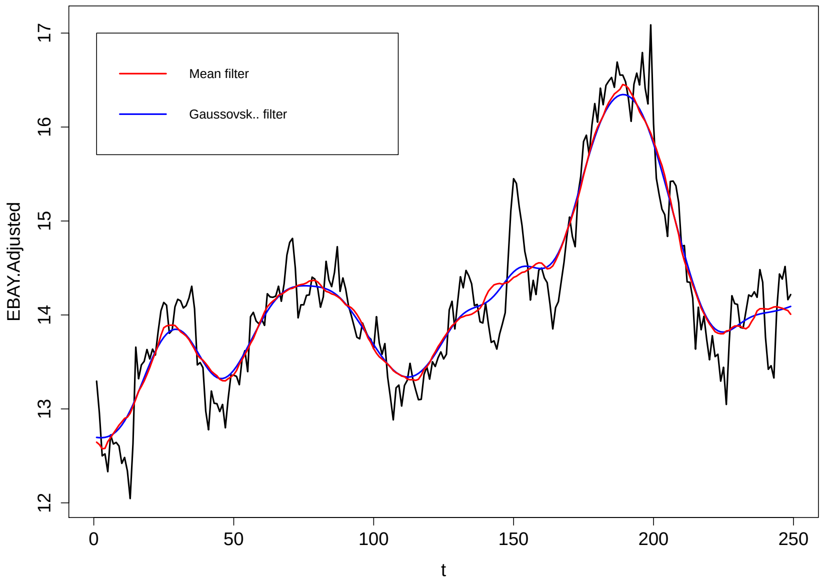
Vyššie uvedený kód sme aplikovali na naše dáta na porovnanie výsledkov filtrovania pri meniacom sa parametri h a fixovanom parametri škálovania $\sigma = 1$. Ako výsledok sme dostali rôzne trendy, ktoré sme uviedli na Obr.9. Zelená krivka na Obr.9 zodpovedá



Obr. 9: Trendy odhadnuté pomocou Gaussovho filtra pri rôznych hodnotach parametra h. Zelená krivka zodpovedá trendu vypočítanému pri voľbe hodnoty $h = 10$, modrá krivka zodpovedá hodnote $h = 20$, červená $h = 30$ a žltá $h = 50$.

trendu vypočítanému pri voľbe hodnoty $h = 10$, modrá krivka zodpovedá hodnote $h = 20$, červená $h = 30$ a žltá $h = 50$. Je vidieť, že zmenou parametra h pri kernel metódach sa taktiež mení hladkosť získaného trendu, podobne ako pri moving average filtroch. Čím je hodnota tohto parametra menšia, tým viac frekvencií a signifikantných výkyvov dát bude zachytených vo vyfiltrovanom trende. Rovnaký vplyv na výsledný trend má aj parameter škálovania σ, čo sa dá usúdiť z analýzy váh, ktorá bola popísaná vyššie bez toho, aby sme zobrazovali výsledky filtrovania.

Ako bolo spomínané vyššie, kernel filtre sú vylepšenou verziou moving average filtrov, čiže tieto dve metódy vo výsledku dávajú rôzne trendy. Na Obr.10 sme uviedli porovnanie trendov získaných pomocou moving average filtra s parametrom $h = 10$ a Gaussovho filtra s parametrami $h = 10$ a $\sigma = 0.68$. Parameter škálovania σ bol zvolený tak, aby sa získaný trend čo najviac podobal trendu moving average filtra.



Obr. 10: Porovnanie Gaussov filter vs. mean filter.

Modrá krivka popisuje trend získaný pomocou Gaussovho filtra, červená zodpovedá výstupu moving average. Z Obr.10 môžeme usúdiť, že filtrovaním dát Gaussovým kernelom sa dá získať trend, ktorý bude vyhladený viac než trend získaný pomocou moving average filtra.

1.3 Metóda exponenciálneho vyhladenia

Metóda exponenciálneho vyhladenia je taktiež jedna z často používaných metód na získanie vyfiltrovaného trendu, alebo vyhladeného časového radu. Táto metóda je jednostranná a na rozdiel od jednostranného moving average filtra umožňuje prideliť starším dátam menšiu váhu [5]. Váhy používané pri tejto metóde klesajú exponenciálne s časom. Pri exponenciálnom vyhladení je potrebné zvoliť hodnotu parametrov vyhladenia a od tejto voľby bude závisieť štruktúra získaných váh, čiže aj tvar vyfiltrovaného trendu.

Existujú tri typy exponenciálnych filtrov: *jednoduchý (single)*, *dvojitý (double)* a *trojitý (triple) exponenciálny filter*, ktoré sa štandardne používajú na predikciu časových radov. Na samotné vyhladenie a odstránenie šumu stačí použiť jednoduchý exponenciálny filter, ktorému sa venujeme v tejto časti. Exponenciálnym filtrom sa okrem iných venovala aj práca [6] a v nej sú okrem jednoduchého spracované aj ďalšie typy týchto filtrov.

Podľa dohody sa pri použití tohto lineárneho filtra za odhad trendovej zložky $\widehat{f_2}$ zvolí hodnota Y_1, čiže prvá hodnota uvažovaného časového radu. Hodnota $\widehat{f_3}$ sa spočíta pomocou vzorca $\widehat{f_3} = \alpha Y_2 + (1-\alpha)\widehat{f_2}$. Podobne sa spočíta odhad trendovej zložky pre každý čas t, čiže všeobecná formulácia je daná rekurziou a vyzerá nasledovne [7]:

$$\widehat{f_t} = \alpha Y_{t-1} + (1-\alpha)\widehat{f_{t-1}} \qquad 0 < \alpha \leqslant 1, \qquad t \geqslant 3, \tag{6}$$

kde α je *konštanta vyhladenia*, ktorá určuje efektivitu veľkosť filtrovania. Existuje aj iná formulácia, pri ktorej sa v tejto rovnici namiesto hodnoty Y_{t-1} používa hodnota Y_t, ktorú sme však v tejto práci nepoužívali. Taktiež sa dajú zvoliť aj rôzne možnosti nastavovania prvej hodnoty $\widehat{f_2}$. Napríklad sa dá použiť priemer prvých štyroch alebo piatich dát. Čím menšia je hodnota parametra α, tým je dôležitejšia správna voľba hodnoty $\widehat{f_2}$ [7].

Uvedený filter sa nazýva exponenciálny kvôli tomu, že váhy zodpovedajúce dátam klesajú exponenciálne s časom. Túto vlastnosť ilustrujeme na nasledujúcom príklade. Pri hľadaní odhadu trendovej zložky v čase t sa postupuje podľa vzorca (6). Podobným činom sa da vyjadriť aj hodnota $\widehat{f_{t-1}}$. Po dosadení tejto hodnoty do vzorca (6) sa získava nasledovné vyjadrenie pre $\widehat{f_t}$:

$$\widehat{f_t} = \alpha Y_{t-1} + (1-\alpha)(\alpha Y_{t-2} + (1-\alpha)\widehat{f_{t-2}}) = \alpha Y_{t-1} + \alpha(1-\alpha)Y_{t-2} + (1-\alpha)^2\widehat{f_{t-2}}.$$

Vzorec pre $\widehat{f_t}$ sa dá prepísať postupným nahradením hodnôt $\widehat{f_{t-2}}$, $\widehat{f_{t-3}}$ a tak ďalej až po hodnotu $\widehat{f_2}$ (rovnej Y_1) na vzorec tvaru [7]:

$$\widehat{f_t} = \alpha \sum_{i=1}^{t-2} (1-\alpha)^{i-1} Y_{t-i} + (1-\alpha)^{t-2}\widehat{f_2}, \quad t \geqslant 2. \tag{7}$$

Tak napríklad hodnota $\widehat{f_5}$, ktorá je odhadom trendu v čase $t = 5$, sa dá vyjadriť podľa (7) v nasledovnom tvare:

$$\widehat{f_5} = \alpha[(1-\alpha)^0 Y_4 + (1-\alpha)^1 Y_3 + (1-\alpha)^2 Y_2] + (1-\alpha)^3 \widehat{f_2}.$$

Ako je vidieť z uvedenej formulácie, váhy sú vyjadrené v tvare $\alpha(1-\alpha)^t$, čiže predstavujú geometricky klesajúcu postupnosť. Podľa vlastnosti geometrickej postupnosti sa dá nájsť súčet prvých t členov tejto postupnosti, a to nasledovným spôsobom [7]:

$$\alpha \sum_{i=0}^{t-1} (1-\alpha)^i = \alpha \left[\frac{1-(1-\alpha)^t}{1-(1-\alpha)} \right] = 1 - (1-\alpha)^t.$$

Vychádzajúc z formulácie exponenciálneho filtra, predpokladáme, že konštanta vyhladenia α je z intervalu $(0, 1]$. Keď $\alpha = 1$, dochádza k špeciálnemu prípadu, keď $\widehat{f_t} = Y_{t-1}$, čiže odhad trendu v čase t sa rovná hodnote merania v čase $t-1$. Ak parameter α budeme uvažovať z intervalu $(0, 1)$, tak potom súčet váh limitne sa rovná 1. To znamená, že [7]:

$$\lim_{t\to\infty} 1 - (1-\alpha)^t = 1,$$

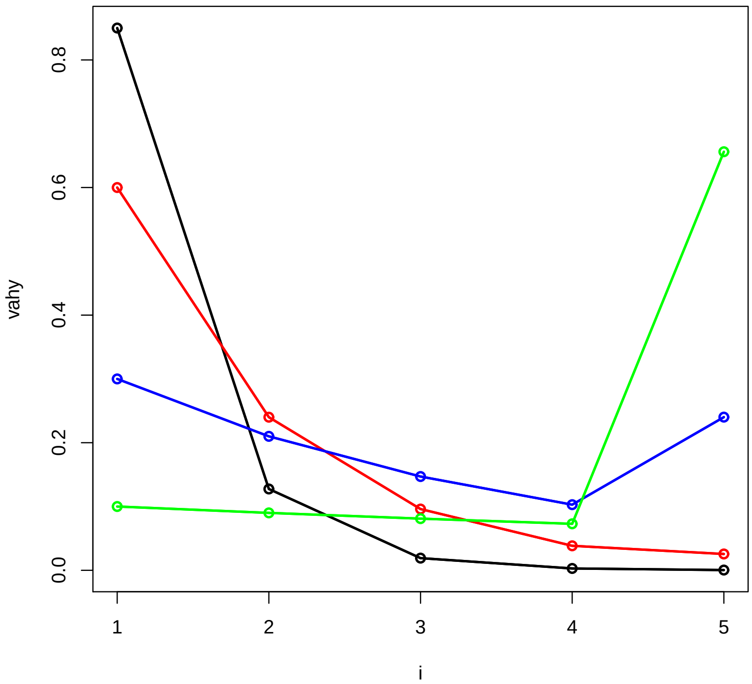
lebo $(1-\alpha)$ je číslo z intervalu $(0, 1)$, čiže pri umocňovaní na nekonečne veľké číslo ide do 0. Vo filtroch, ktoré už boli spomenuté v tejto práci, sa táto vlastnosť dosahovala tým, že sa váhy predeľovali súčtom všetkých váh používaných pri odhadovaní trendu v danom časovom bode, čo pri exponenciálnom filtri nie je potrebné.

Rýchlosť, s ktorou sa vyhladzujú staršie dáta, je funkciou konštanty vyhladenia α. Keď α je blízko k 1, vyhladenie je rýchle a keď α je blízko 0, vyhladenie je pomalé [7]. Charakter závislosti výstupu exponenciálneho filtra od konštanty vyhladenia sme ukázali na príklade odhadovania hodnoty $\widehat{f_6}$, čo je odhadom trendu v bode $t = 6$. Podľa algoritmu popísaného vyššie sme vyrátali váhy, ktoré sa prideľujú dátam $Y_1, Y_2, \ldots, Y_5$ pre štyri rôzne hodnoty α. Výsledky sme uviedli do Tab.1.

Tabuľka 1: Váhy priradené dátam v závislosti od hodnoty α.

	Y_5	Y_4	Y_3	Y_2	Y_1
$\alpha = 0.85$	0.85	0.1275	0.0191	0.0029	0.0004
$\alpha = 0.6$	0.6	0.24	0.096	0.0384	0.0256
$\alpha = 0.3$	0.3	0.21	0.147	0.1029	0.2401
$\alpha = 0.1$	0.1	0.09	0.081	0.0729	0.6561

Obr.11 zobrazuje váhy pre rôzne hodnoty α. Čierne spojené čiarami krúžky zodpove-



Obr. 11: Grafické zobrazenie váh, vypočítaných pre rôzne hodnoty α.

dajú váham, ktoré sa prideľujú dátam Y_{6-i} pri hodnote $\alpha = 0.85$, červené zodpovedajú hodnote $\alpha = 0.6$, modré $\alpha = 0.3$, zelené $\alpha = 0.1$. Ako už bolo spomínané vyššie, váhy priradené starším dátam klesajú exponenciálne. Keď sa zvolí hodnota $\alpha < 0.5$, tak váha priradená hodnote $\widehat{f_2}\,(Y_1)$ bude väčšia než váha zodpovedajúca predošlej hodnote. Taktiež je vidieť, že čím menšia je hodnota parametra α, tým viac dát dostane signifikantné váhy pri výpočte odhadu trendu v danom časovom bode.

2 Nelineárne filtre

Pri napísaní danej kapitoly sme vychádzali hlavne z [8].

Ako sme videli, lineárne filtre majú tendenciu zhladzovať okraje a ďalšie detaily, aj keď tvoria základnú súčasť trendu časových radov. Aby sa tomu zabránilo, vyvinula sa široká škála nelineárnych filtrov, ktoré na jednej strane zachovávajú tieto detaily, zatiaľ čo na druhej strane sa snažia vyhladiť čo najviac šumu. Nelineárne filtre sa používajú nielen v časových radoch, ale častokrát sa používajú špeciálne pre dvojrozmerné signály, konkrétne obrázky, kde pôvodný obrázok, pravdepodobne poškodený šumom pri prenose dát, pozostáva prevažne z hrán, ktoré tvoria obrázok. Analýzu fungovania nelineárnych filtrov a ich porovnanie z lineárnymi sme robili na jednorozmerných dátach, čiže rovnako ako v predošlej kapitole.

Na lepšie pochopenie vlastností a výhod nelineárnych filtrov oproti lineárnym sme spravili porovnanie dvoch základných filtrov týchto dvoch tried, konkretne mean filter, ktorý sme popisovali v prvej kapitole, a *mediánový filter*, ktorý patrí do kategórie nelineárnych filtrov. Pre dané účely budeme predpokladať, že poškodený šumom signál má nasledovný tvar:

$$Y_t = f_t + X_t,$$

kde f_t predstavuje pôvodný nepoškodený signál a X_t je šum, ktorý je potrebné odstrániť. Inými slovami, úlohou je získať signál f_t, ak máme jeho poškodenú verziu Y_t. Samozrejme, v mnohých prípadoch nie je možne nájsť presné riešenie pre takúto úlohu, ale po zavedení zjednodušujúcich predpokladov alebo pri poznaní charakteristík pôvodného signálu sa problém dá vyriešiť existujúcimi metódami dostatočne presne. Napríklad lineárne filtre fungujú optimálne, ak predpokladáme, že šum je gaussovský a odchýlka od presného riešenia sa počíta ako stredná kvadratická odchýlka. V reálnom svete je však charakter šumu častokrát ďaleko od gaussovského rozdelenia. Keď nie sú splnené predpoklady klasickej teórie lineárnych filtrov, použitie nelineárnych filtrov môže viesť k lepšiemu riešeniu.

2.1 Odstraňovanie gaussovského šumu mean a mediánovým filtrami

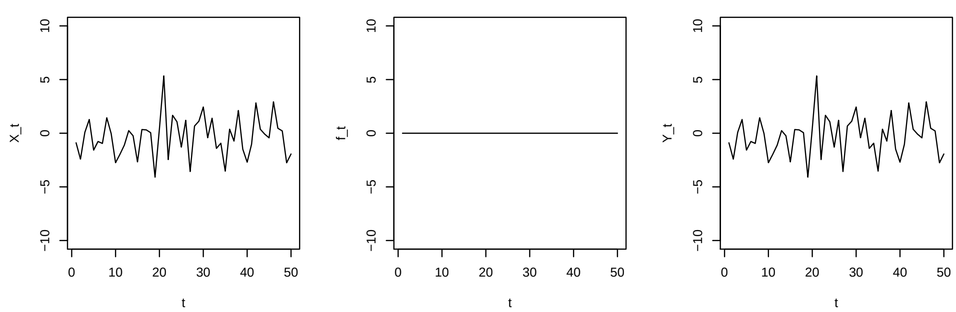
Porovnanie mean a median filtrov sme spravili na jednoduchých príkladoch, ktoré poukazujú na hlavné problémy, s ktorými sa stretávame pri odstraňovaní šumu z dát. Uvažovali sme tri základné tvary signálov, ktoré sú: *konštantný signál*, *lineárny sklon* a *krokový signál*. Ich formulácia je nasledovná:

- $f_t = a$, $t = 1, 2, \ldots, 50$ (konštantný signál);
- $f_t = 0.2t$, $t = 1, 2, \ldots, 50$ (lineárny sklon);
- $f_t = \begin{cases} 0, & i = 1, 2, \ldots, 25 \\ 10, & i = 26, 27, \ldots, 50 \end{cases}$ (krokový signál).

Predpokladáme, že X_t, $t = 1, 2, \ldots, 50$ sú nezávislé náhodné premenné z gaussovského rozdelenia $N(0, 4)$, ktoré predstavujú šum. Premenné X_t sme generovali pomocou príkazu:

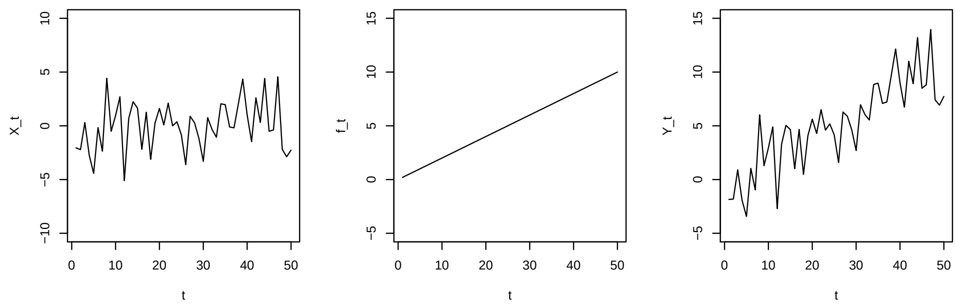
```
rnorm(50, mean = 0, sd = 2),
```

čiže v každom časovom bode pôvodný signál je zašumený náhodne vygenerovanou zložkou. Dane tri typy signálov spolu s ich zašumenou verziou sme uviedli na Obr.12, Obr.13 a Obr.14.

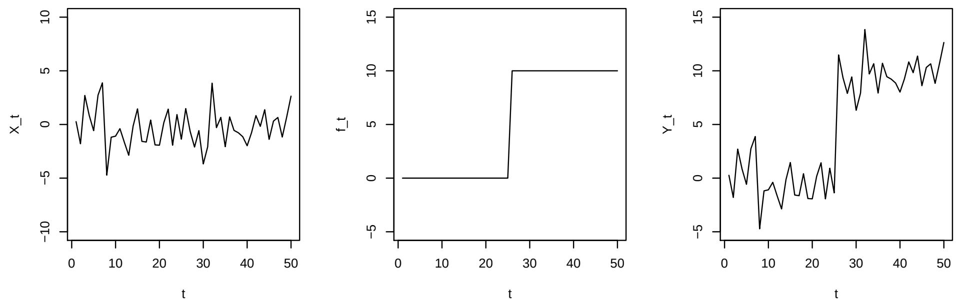


Obr. 12: Náhodne vygenerovaný šum (vľavo); konštantný signál (uprostred); zašumená verzia signálu (vpravo).

Na Obr.12 je uvedený konštantný signál f_t (uprostred), náhodne vygenerovaný šum X_t (vľavo) a zašumená verzia daného signálu Y_t (vpravo), ktorá bola získaná pomocou súčtu



Obr. 13: Náhodne vygenerovaný šum (vľavo); lineárny sklon (uprostred); zašumená verzia signálu (vpravo).



Obr. 14: Náhodne vygenerovaný šum (vľavo); krokový signál (uprostred); zašumená verzia signálu (vpravo).

$Y_t = f_t + X_t$. Na Obr.13 sú uvedené podobné grafy pre signál, ktorý predstavuje lineárny sklon a Obr.14 zodpovedá grafom pre krokový signál.

Dôležitým pozorovaním je to, že odstraňovanie šumu zo signálu pomocou lineárnych filtrov je *lineárnou operáciou*, ktorej definícia má nasledovné znenie [8]:

Definícia 2.1. *Nech N, M sú vektorové priestory nad telesom K. Zobrazenie $\phi : N \to M$ sa nazýva lineárne zobrazenie, ak je splnené nasledovné:*

- $\phi(x + y) = \phi(x) + \phi(y)$
- $\phi(\alpha x) = \alpha\phi(x)$

Daný fakt dáva možnosť pozorovať účinok lineárnych filtrov na poškodený signál tak, že filter sa aplikuje zvlášť na pôvodný signál a na šum. Uvažovali sme mean filter s dĺžkou okna $L = 7$, čiže hodnota parametra, zodpovedajúceho za šírku pásma priepuste, je $h = 3$.

Potom výsledok filtrovania signálu v čase t, pomocou moving average filtra, sa dá rozpísať nasledovne:

$$\widehat{f}_t = \frac{1}{7}\sum_{i=-3}^{3} Y_{t+i} = \frac{1}{7}\sum_{i=-3}^{3} f_{t+i} + \frac{1}{7}\sum_{i=-3}^{3} X_{t+i} = (\textit{filtrovaný pôvodný signál}) + (\textit{filtrovaný šum}).$$

Zo štatistiky je známe, že variancia priemeru siedmich nezávislých náhodných premenných, z ktorých každá má varianciu rovnú 1, je $\frac{1}{7}$. Čiže po vyhladení pomocou mean filtra je šum X_t zoslabený na 86%, ale celkové vyhladenie poškodeného signálu Y_t je skreslené filtrovaním pôvodného signálu bez šumu f_t, čo bude vidieť pri zobrazovaní výsledkov analýzy.

Rovnaké ukážkové signály sme filtrovali aj takzvaným mediánovým filtrom, ktorý je jedným z často používaných nelineárnych filtrov a má nasledovnú formuláciu [8]:

$$\widehat{f}_t = MED\{Y_{t-h}, Y_{t-h+1}, \ldots, Y_{t+h-1}, Y_{t+h}\} \tag{8}$$

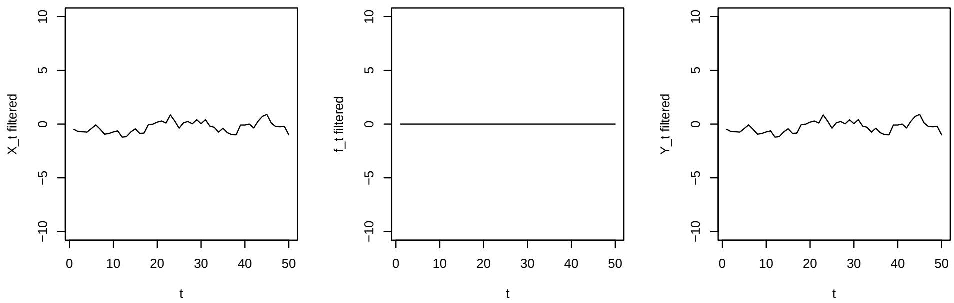
kde MED je operátor, ktorý po zoradení prvkov množiny vyberie centrálny prvok, napríklad:

$$MED\{1,3,5,2,7,6,4\} = MED\{1,2,3,4,5,6,7\} = 4$$

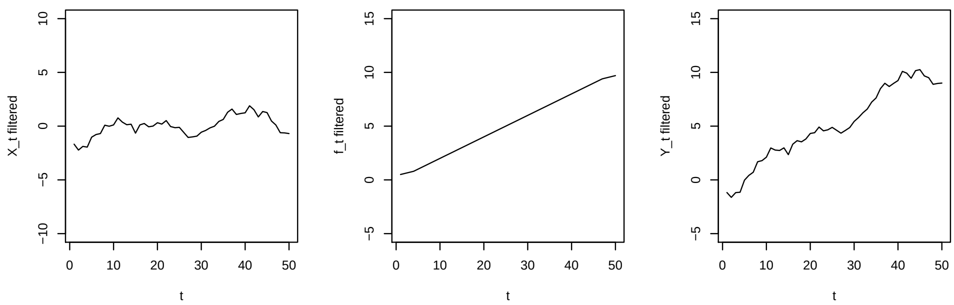
a parameter h je definovaný rovnako ako pri moving average filtri, čiže zodpovedá za dĺžku okna $L = (2h+1)$ dát vstupujúcich do filtra pri odhadovaní hodnoty $\widehat{f}_t$. Nelineárne filtre nespĺňajú predpoklady lineárneho operátora, čiže ich nie je možné aplikovať zvlášť na šum a signál bez šumu. Čiže analyzovať a pozorovať fungovanie daných filtrov je možné iba na výslednom očistenom signáli.

Na Obr.15, Obr.16 a Obr.17 sme uviedli výsledky filtrovania experimentálnych signálov pomocou mean filtra. Z toho dôvodu, že mean filter je lineárnym operátorom, na daných obrázkoch uprostred je ukázaný výstup filtrovania pôvodných nezašumených signálov f_t, vľavo je uvedený filtrovaný šum X_t a vpravo sa nachádza súčet týchto dvoch signálov, ktorý je výstupom filtrovania poškodeného signálu Y_t.
Na Obr.18, Obr.19 a Obr.20, sme ukázali zašumený experimentálny signál Y_t (vľavo) a výsledok aplikovania na daný signál mediánového filtra (vpravo).

Z Obr.15 je vidieť, že konštantný signál sa nezmenil pri filtrovaní mean filtrom, čiže výsledok filtrovania poškodeného konštantného signálu je ovplyvnený len pôsobením filtra na šum. Pôsobením mediánového filtru na daný signál sa odstráni šum do rovnakej miery, čo vidíme z Obr.18.



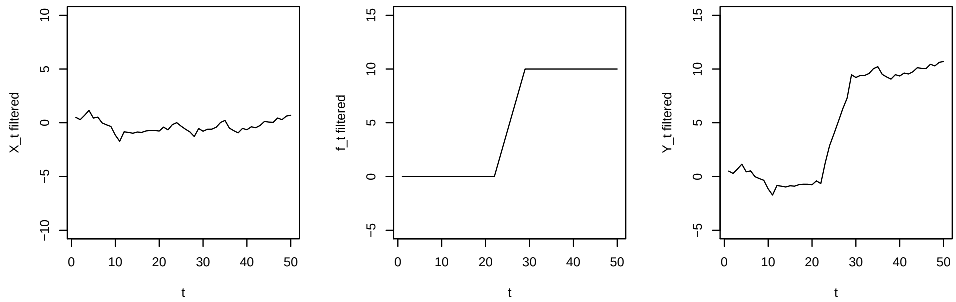
Obr. 15: Filtrovaný šum (vľavo); filtrovaný konštantný signál (uprostred); výstup filtrovania poškodeného signálu (vpravo).



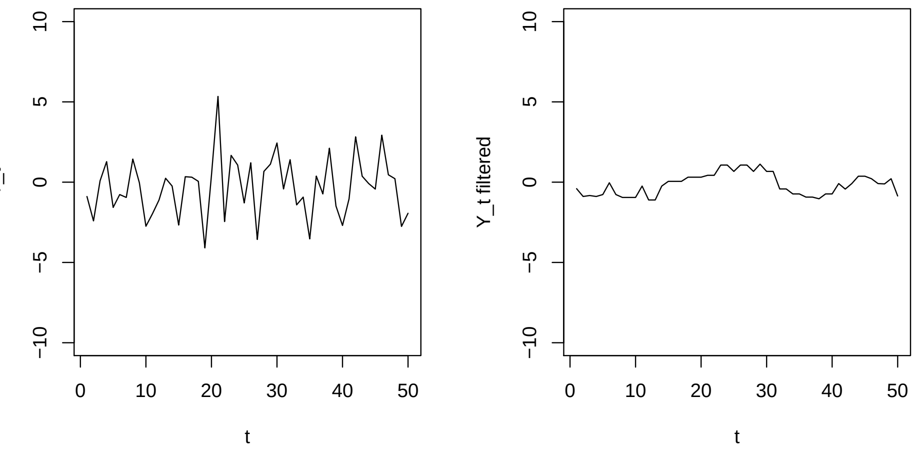
Obr. 16: Filtrovaný šum (vľavo); filtrovaný lineárny sklon (uprostred); výstup filtrovania poškodeného signálu (vpravo).

Na Obr.16 vidíme, že lineárny sklon sa nezmenil po aplikovaní mean filtra, okrem oblasti na začiatku a na konci daného signálu, kde pri filtrovaní v čase t sa zoberie viac dát z jednej strany od daného časového bodu. Podobne je to aj v prípade mediánového filtra. Daný problém sme už spomínali v predošlej kapitole pri popisovaní vlastnosti lineárnych filtrov. Pri odstránení šumu pomocou mean filtra sa zachováva trend lineárneho sklonu, čo nie je až tak dobre vidieť na Obr.19, ktorý zobrazuje výsledok používania na daný signál mediánového filtra. Schodovitý tvar očisteného signálu je spôsobený tým, že pri dostatočne strmom sklone šum nezmení poradie prvkov množiny dát, vstupujúcich do filtra pri odhadovaní hodnoty očisteného signálu v čase t, čiže šum sa neodstráni do dostatočnej miery.

Krokový signál na Obr.17 je skreslený pôsobením mean filtra. V porovnaní s pôvodným signálom zobrazeným na Obr.14, kde skok nastal okamžite, je vidieť, že v tomto prípade skok sa predĺžil. Čiže aj filtrovaná verzia zašumeného krokového signálu bude z tohto



Obr. 17: Filtrovaný šum (vľavo); filtrovaný krokový signál (uprostred); výstup filtrovania poškodeného signálu (vpravo).



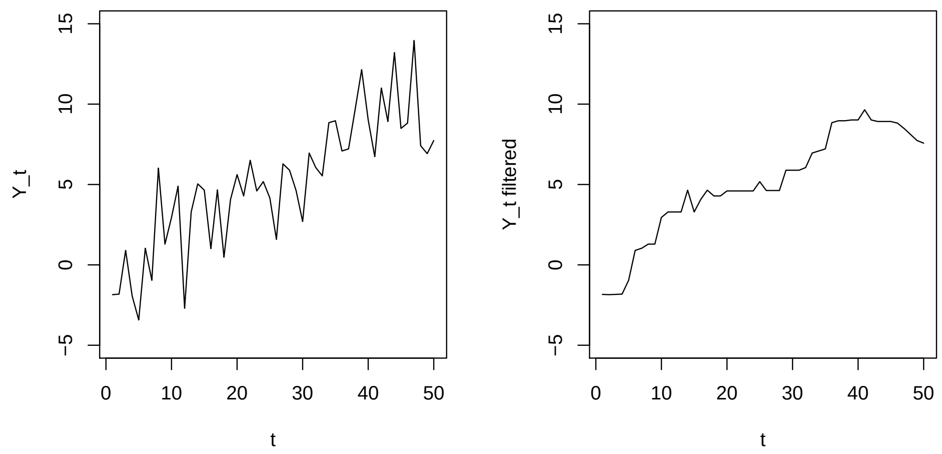
Obr. 18: Zašumený konštantný signál (vľavo); výstup filtrovania mediánovým filtrom (vpravo).

dôvodu skreslená. V prípade mediánového filtra dochádza k inému problému, dolná časť kroku je zaoblená, čo vidíme na Obr.20. Je to spôsobené tým, že pri počítaní filtrovanej hodnoty v dolnej časti kroku sa zoberú tri hodnoty z jeho hornej časti, čiže výsledkom filtrovania bude maximálna hodnota štyroch dolných dát.

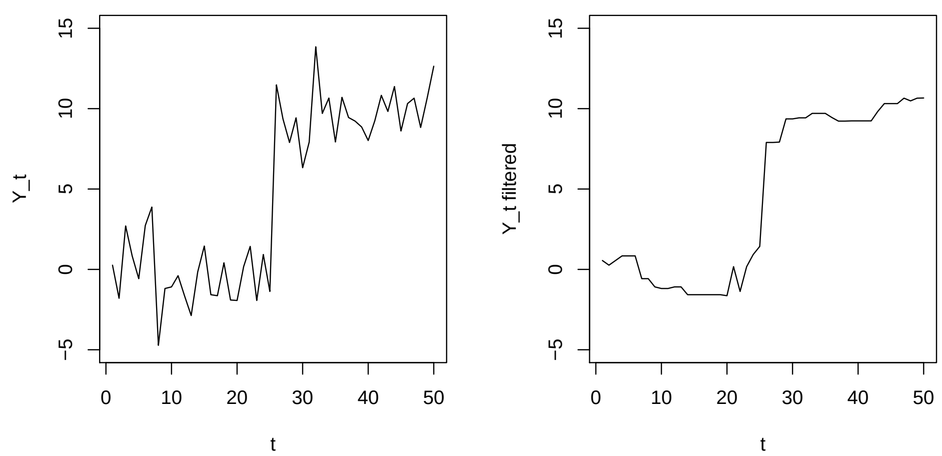
Veľkosť nepresnosti filtrovania šumu zo signálu sa dá oceniť nielen z grafického zobrazenia pôvodného a očisteného signálu. Jednou z možných metód vyhodnotenia nepresnosti je *stredná kvadratická odchýlka* (MSE). V našom prípade sme rátali odchýlku očisteného signálu od pôvodného, čiže vzorec pre MSE signálu je nasledovný:

$$MSE = \frac{1}{50}\sum_{t=1}^{50}(\widehat{f_t} - f_t)^2.$$

Do Tab.2 sme uviedli hodnoty MSE pre tri experimentálne signály a dva skúmané typy filtrov.



Obr. 19: Zašumený lineárny sklon (vľavo); výstup filtrovania mediánovým filtrom (vpravo).



Obr. 20: Zašumený krokový signál (vľavo); výstup filtrovania mediánovým filtrom (vpravo).

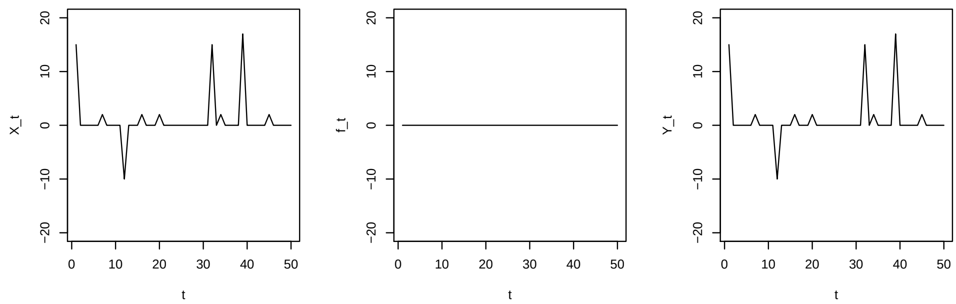
Z Tab.2 je vidieť, že skúmané filtre sa najlepšie preukázali na konštantnom signáli. Hodnoty MSE sú v tomto prípade podobné. Z porovnania obrázkov sme usúdili, že pri lineárnom sklone mean filter funguje lepšie než mediánový filter. Daný fakt sa potvrdil aj hodnotami strednej kvadratickej odchýlky. Pri lineárnom sklone je hodnota MSE zodpovedajúca mean filtru je menšia. Taktiež vidíme, že rozšírenie skoku, spôsobené aplikovaním mean filtra na krokový signál, sa prejavilo v hodnote MSE. Stredná kvadratická odchýlka zodpovedajúca mean filtru je výrazne vyššia, než hodnota zrátaná pre mediánový filter. Čiže krokový signál sa lepšie očisťuje použitím nelineárneho filtra.

Tabuľka 2: Hodnoty MSE pre tri experimentálne signály a skúmane typy filtrov.

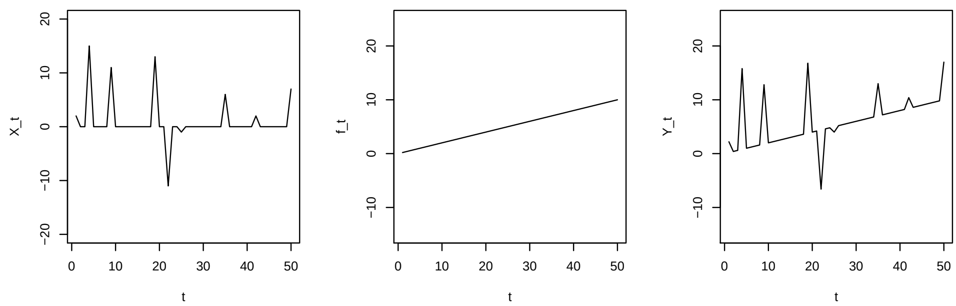
	Konštantný signál	Lineárny sklon	Krokový signál
Mean filter	0.3494	0.842	1.8044
Median filter	0.4763	1.2961	1.0566

2.2 Impulzný šum

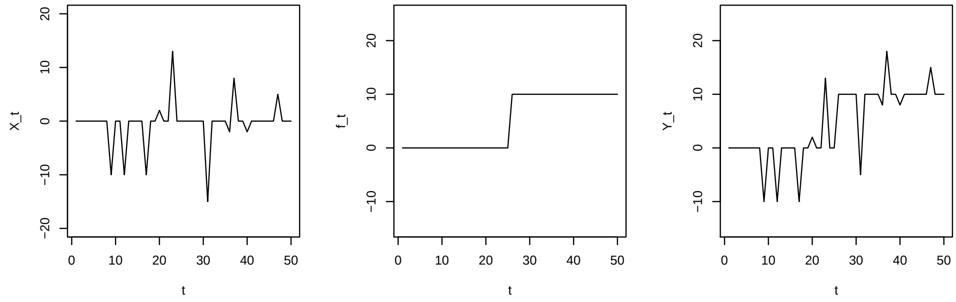
V predošlej časti sme analyzovali správanie jednotlivých filtrov na signáloch, ktoré boli poškodené gaussovským šumom. V tejto časti budeme skúmať mean a mediánový filter v súvislosti s odstraňovaním *impulzného šumu*. Za impulzný šum sa považuje signál, ktorý v niektorých časových okamihoch obsahuje okamžité skoky rôznej veľkosti. Na Obr.21, Obr.22 a Obr.23 uvádzame príklady impulzného šumu (vľavo), experimentálne signály (uprostred), ktoré sme používali pri analýze v predošlej časti a taktiež ich zašumenú verziu (vpravo).



Obr. 21: Impulzný šum (vľavo); konštantný signál (uprostred); zašumená verzia signálu (vpravo).

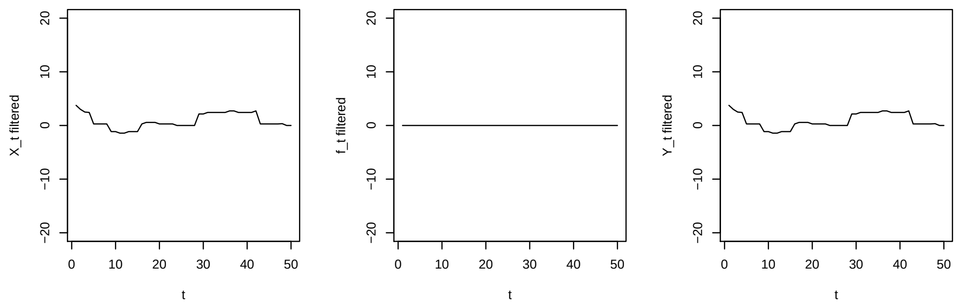


Obr. 22: Impulzný šum (vľavo); lineárny sklon (uprostred); zašumená verzia signálu (vpravo).



Obr. 23: Impulzný šum (vľavo); krokový signál (uprostred); zašumená verzia signálu (vpravo).

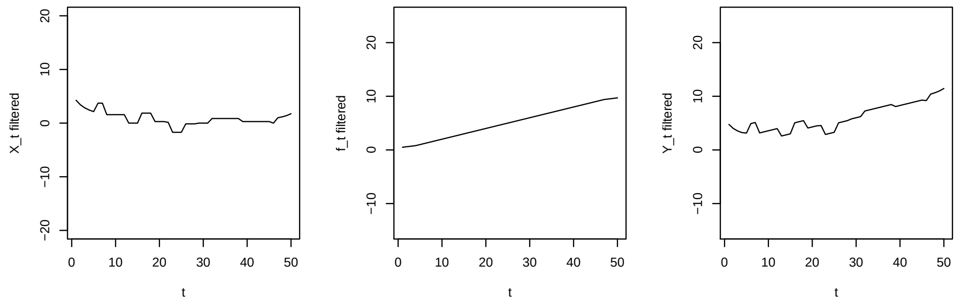
Z obrázkov je vidieť, že impulzný šum pridáva okamžité a výrazné zmeny do skutočného signálu, čiže aj skúmané filtre budú fungovať inak než pri gaussovskom šume. Rovnakým spôsobom, ako v predošlej časti sme aplikovali mean filter s parametrom $h = 3$ zvlášť na očistený signál a na impulzný šum. Tak na Obr.24, Obr.25 a Obr.26 sú uvedené výsledky filtrovania signálov použitím daného filtra.



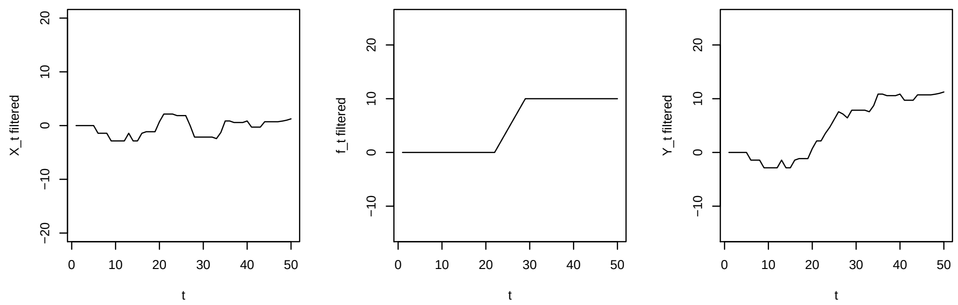
Obr. 24: Filtrovaný impulzný šum (vľavo); filtrovaný konštantný signál (uprostred); výstup filtrovania poškodeného signálu (vpravo).

Na Obr.27, Obr.28 a Obr.29 sme ukázali filtrovanie zašumených experimentálnych signálov pomocou mediánového filtra.

Z uvedených obrázkov je hneď vidieť niektoré rozdiely v správaní mean a mediánových filtrov v súvislosti s impulzným šumom. Mean filter rozširuje impulz, ale aj zmenšuje jeho amplitúdu. Mediánový filter, na rozdiel od lineárneho mean filtra, umožňuje kompletne zredukovať pôsobenie impulzného šumu, čo je jednou z fundamentálnych vlastností a výhod nelineárnych metód. Dokonca mediánový filter, ktorý má dĺžku okna rovnú $L = 2h + 1$, úplne odstraňuje impulz dĺžky $h + 1$. Táto vlastnosť sa dá lepšie pochopiť na príklade signálu zašumeného niekoľkými impulzami rôznej dĺžky. Na Obr.30 sme uviedli



Obr. 25: Filtrovaný impulzný šum (vľavo); filtrovaný lineárny sklon (uprostred); výstup filtrovania poškodeného signálu (vpravo).

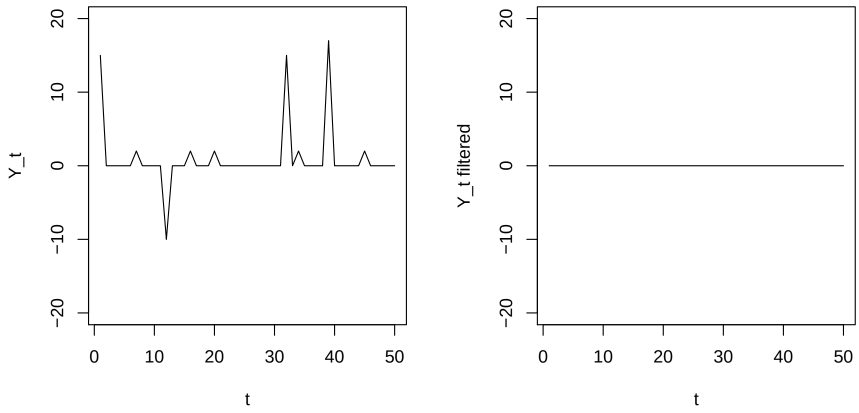


Obr. 26: Filtrovaný impulzný šum (vľavo); filtrovaný krokový signál (uprostred); výstup filtrovania poškodeného signálu (vpravo).

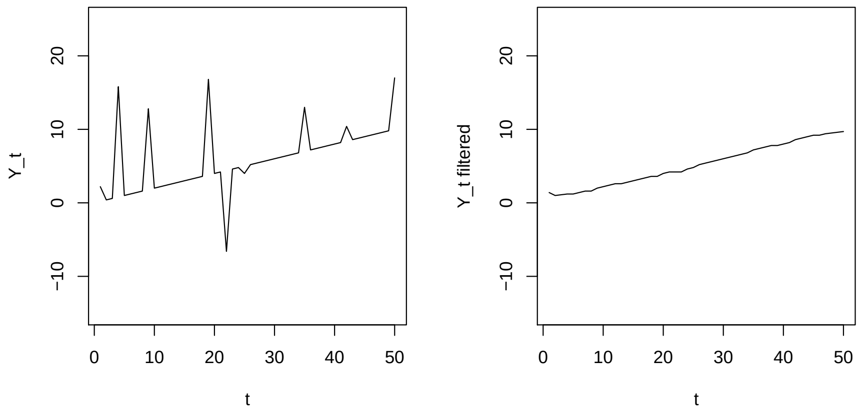
konštantný signál zašumený impulzami dĺžky 1, 2, 3 a 4. Na Obr.31 je filtrovaná verzia daného signálu, ktorú sme získali pomocou mediánového filtra s hodnotou parametra $h = 3$.

Z Obr.31 je vidieť, že nami zvolený filter odstránil všetky impulzy okrem impulzu dĺžky 4. Je dôležité si uvedomiť, že také ideálne správanie mediánového filtra v prípade impulzného šumu sa dá pozorovať len na konštantných signáloch. Napríklad, impulz, ktorý sa nachádza v blízkosti hrany alebo kroku, môže byť odstránený použitím mediánového filtra, ale súčasne bude spôsobený posun hrany v smere odstráneného impulzu, čo skreslí výsledok. Mnohé literárne zdroje daný problém menujú ako *edge jitter* [8]. Na lepšie pochopenie tohto problému sme použili krokový signál zobrazený na Obr.32, na ktorý sme následne aplikovali mediánový filter s hodnotou $h = 5$. Výsledok filtrovania je uvedený na Obr.33.

Vidíme, že zvolený filter odstránil impulzný šum, ale zároveň krok, ktorý sa začínal v



Obr. 27: Zašumený konštantný signál (vľavo); výstup filtrovania mediánovým filtrom (vpravo).



Obr. 28: Zašumený lineárny sklon (vľavo); výstup filtrovania mediánovym filtrom (vpravo).

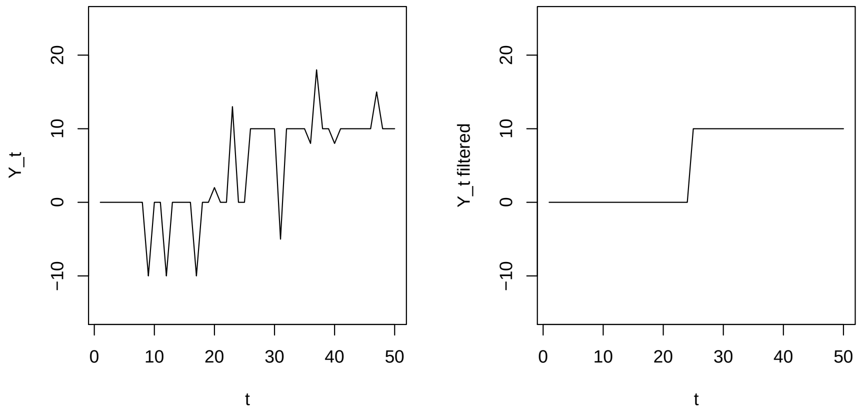
časovom bode $t = 11$, sa posunul do bodu $t = 10$. Filtrovaná hodnota v čase $t = 11$ sa spočítala nasledovne:

$$\widehat{f_{11}} = MED\{\mathbf{1}, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1\} = MED\{0, 0, 0, 0, 0, \mathbf{1}, 1, 1, 1, 1, 1\} = 1,$$

čiže hodnota impulzného šumu, ktorá bola zachytená oknom filtra, spôsobila chybnú zmenu hodnoty signálu v danom časovom bode. Tento problém sa bude prejavovať vo veľkosti nepresnosti očistenia impulzného šumu mediánovým filtrom.

Podobným spôsobom ako v predošlej časti sme spočítali hodnoty strednej kvadratickej odchýlky pre tri typy signálov a dva skúmané filtre. Tieto hodnoty uvádzame v Tab.3.

Z Tab.3 je vidieť, že pri konštantnom signáli a pri lineárnom sklone sa mean filter preukázal zjavne horšie než mediánový filter, ktorý v tomto prípade fungoval ideálne. Taktiež hodnoty MSE spočítané pre mean filter v prípade impulzného šumu sú výrazne



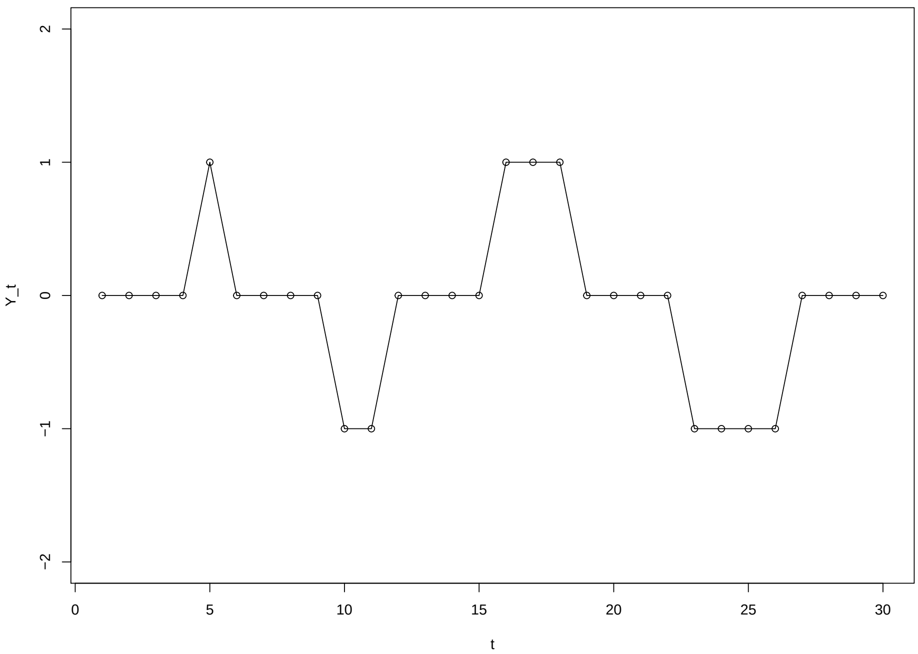
Obr. 29: Zašumený krokový signál (vľavo); výstup filtrovania mediánovým filtrom (vpravo).

Tabuľka 3: Hodnoty MSE pre tri experimentálne signály a skúmane typy filtrov.

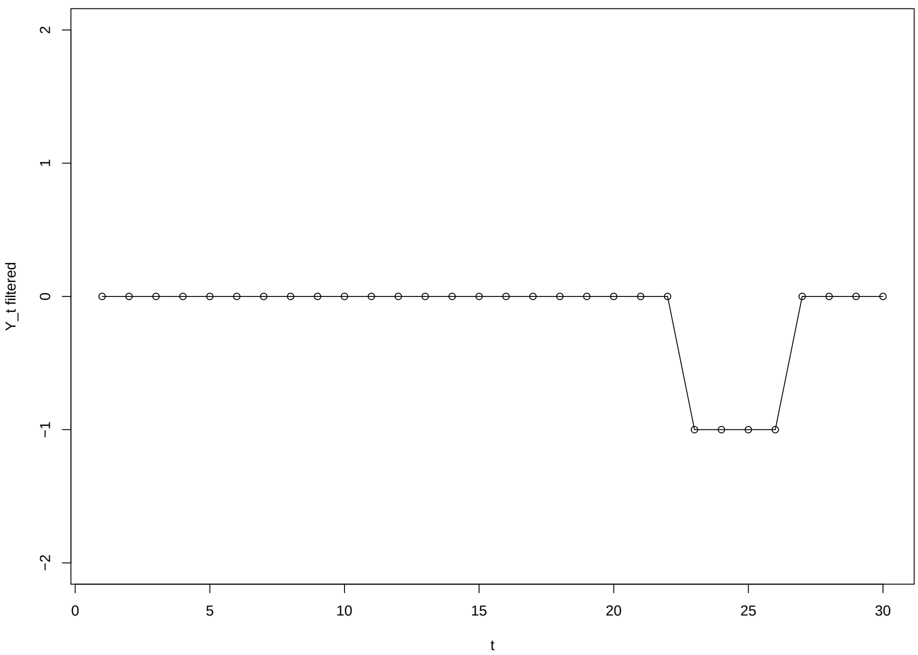
	Konštantný signál	Lineárny sklon	Krokový signál
Mean filter	2.6485	2.4716	4.1394
Median filter	0	0.0654	2

vyššie od hodnôt zodpovedajúcich Gassovskému šumu, ktoré sme videli v Tab.2. V prípade krokového signálu veľkosť nepresnosti očistenia mediánovým filtrom bola spôsobená posunom hrany, ale hodnota strednej kvadratickej odchýlky je zároveň výrazne menšia než hodnota zodpovedajúca mean filtru.

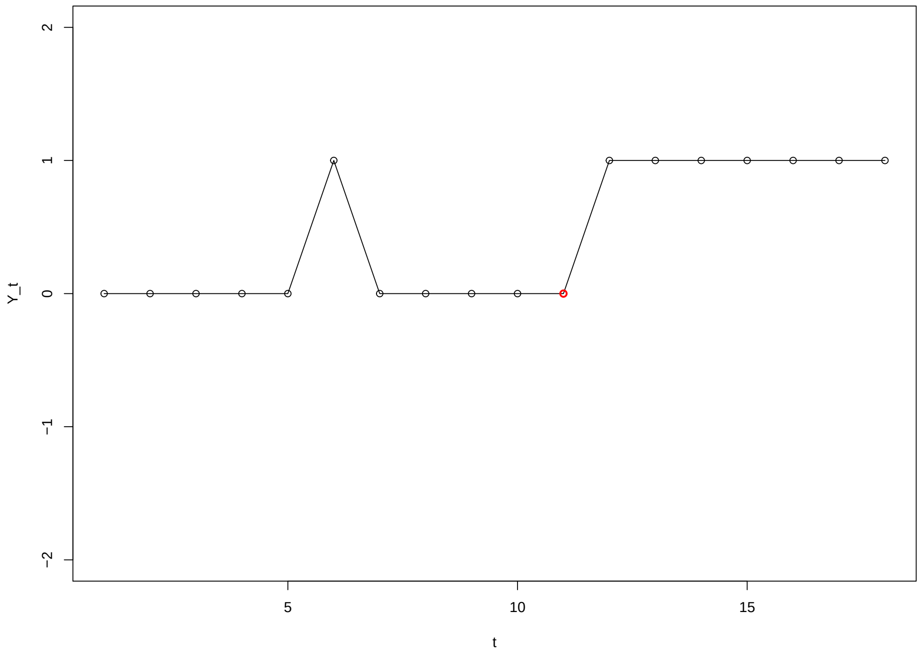
Prebraté príklady poukázali na výhody a nevýhody aplikovania mediánového filtra, ktorý patrí do triedy nelineárnych filtrov. Filter sa veľmi dobre ukázal pri probléme odstraňovania impulzného šumu, ale očistenie signálu od gausovského šumu bolo pre túto metódu problematické. Čiže v situáciách, keď sa v signáli nachádza šum obidvoch typov, daná metóda nie je najlepšou voľbou. Preto boli vyvinuté iné typy filtrov, ktoré predstavujú pevnú kombináciu metodík fungovania mean a mediánových filtrov. Jednou z takých metód je napríklad *trimmed mean filter*, popísaný v nasledujúcej časti.



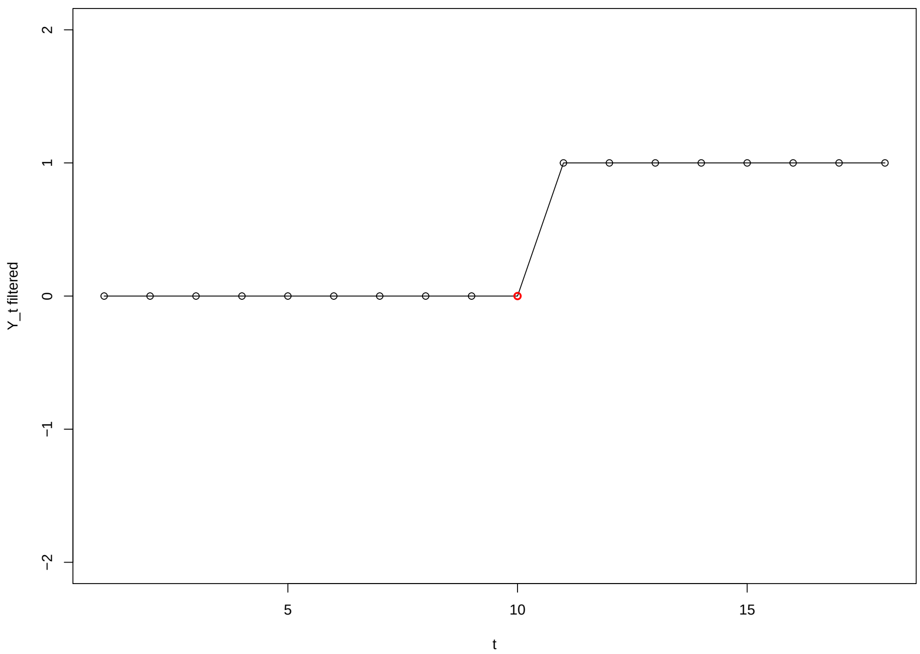
Obr. 30: Ukážkový konštantný signál zašumený impulzami dĺžky 1, 2, 3 a 4.



Obr. 31: Výsledok foltrovania zašumeného signálu pomocou mediánového filtra s parametrom $h = 3$.



Obr. 32: Ukážkový krokový signál. Červeným kružkom je znazornený začiatok kroku v čase $t = 11$.



Obr. 33: Výsledok foltrovania zašumeného signálu pomocou mediánového filtra s parametrom $h = 5$. Červeným kružkom je znazornený začiatok kroku v čase $t = 10$.

2.3 Trimmed mean filtre

Prvý článok, v ktorom sa spomína metóda *osekávania*, pochádza ešte z roku 1821. Autor daného článku nie je známy. Myšlienka tejto metódy spočíva v tom, že sa zamietnu hodnoty, ktoré sa môžu byť považované za outlierov, to sú najväčšie a najmenšie hodnoty vstupujúce do okna filtrovania. Po odseknutí časti dát sa ostatné hodnoty spriemerujú a vytvoria výstup filtra v danom časovom bode [8].

Trimmed mean filtre [9], [10] tvoria skupinu filtrov, ktoré sa líšia spôsobom výberu hodnôt, ktoré sa odstraňujú z priemeru. Tak napríklad, *(r, s) – fold trimmed mean filter* sa formuluje nasledujúcim spôsobom [8]:

$$TrMean(Y_1, Y_2, \ldots, Y_N; r, s) = \frac{1}{N - r - s} \sum_{i=r+1}^{N-s} Y_{(i)}, \tag{9}$$

kde $Y_{(i)}$ predstavujú hodnoty zoradené podľa veľkosti a $N = 2k + 1$ je veľkosť okna daného filtra (analógia s $L = 2h + 1$), čiže počet dát vstupujúcich do priemeru v danom časovom bode. Z formulácie (9) je vidieť, že daný filter funguje tak, že sa z hodnôt N zoradených podľa veľkosti vstupujúcich do priemeru v danom časovom bode odstráni $r+s$ dát. Presnejšie, vynechá sa r najmenších a s najväčších hodnôt, čiže priemer sa vypočíta z $N - r - s$ dát.

Existuje aj modifikovaná verzia tohto filtra, ktorá sa nazýva *(r, s) – fold winsorized mean filter*. Táto metóda sa od predošlej líši tým, že sa r najmenších hodnôt vymení za hodnotu $Y_{(r+1)}$ a s najväčších za $Y_{(N-s)}$. Potom formulácia danej metódy vyzerá nasledovne [8]:

$$WinMean(Y_1, Y_2, \ldots, Y_N; r, s) = \frac{1}{N}(rY_{(r+1)} + \sum_{i=r+1}^{N-s} Y_{(i)} + sY_{(N-s)}). \tag{10}$$

Ak je počet hodnôt odstránených z oboch strán rovnaký, čiže $r = s$, potom sa tento počet zvykne označovať ako časť z celkového počtu hodnôt N, čiže $\alpha = \frac{j}{N}$, kde $0 \leq j \leq N/2$ je celé číslo. Z oboch strán sa potom vynechá αN hodnôt. Vyššie uvedené filtre sa označujú ako *α-trimmed mean filter* a *α-Winsorized mean filter* a majú nasledovnú formuláciu [8]:

$$TrMean(Y_1, Y_2, \ldots, Y_N; \alpha) = \frac{1}{N - 2\alpha N} \sum_{i=\alpha N+1}^{N-\alpha N} Y_{(i)}; \tag{11}$$

$$WinMean(Y_1, Y_2, \ldots, Y_N; \alpha) = \frac{1}{N}(\alpha N \cdot Y_{(\alpha N+1)} + \sum_{i=\alpha N+1}^{N-\alpha N} Y_{(i)} + \alpha N \cdot Y_{(N-\alpha N)}). \quad (12)$$

Zo vzorcov (9), (10) vyplývajú nasledujúce vzťahy:

1. Ak sa zvolia hodnoty $r = 0$ a $s = 0$, tak (r, s) – fold trimmed mean filte a (r, s) – fold winsorized mean filter sa pretransformujú na obyčajný mean filter s rovnákou dĺžkou okna (v tomto prípade rovnou N).

2. Pri voľbe $r = k$ a $s = k$, kde k je parameter určujúci dĺžku okna filtra $N = 2k + 1$, tak (r, s) – fold trimmed mean filter a (r, s) – fold winsorized mean filter sa pretransformujú na mediánový filter.

Podobne, z formulácii (11) a (12) vyplývajú vzťahy:

1. Voľbou parametra $\alpha = 0$ vieme pretransformovať α-trimmed mean filter a α-winsorized mean filter na mean filter.

2. Pri voľbe $\alpha = 0.5$ α-trimmed mean filter a α-winsorized mean filter sa zredukujú na mediánový filter.

Z daných vzťahov sa dá usúdiť, že čím viac hodnôt bude odseknutých pri priemerovaní, tým sa bude správanie daných trimmed mean filtrov viac podobať na správanie mediánového filtra. Naopak, zmenšenie počtu vynechaných hodnôt približuje výstup trimmed mean filtrov k výstupu mean filtra. Čiže hodnoty parametrov r, s a α zodpovedajú za *trade-off* v správaní trimmed mean filtrov, čo znamená, že sa pomocou týchto parametrov dá vhodne kombinovať charakter pôsobenia mean a mediánového filtrov na zašumený signál.

Predpokladajme, že v pevnom časovom bode vstupy filtra majú nasledovný tvar: $y = \{2, 7, 9, 20, 5\}$. Po usporiadaní daných hodnôt dostávame postupnosť $Y_{(1)} = 2$, $Y_{(2)} = 5$, $Y_{(3)} = 7$, $Y_{(4)} = 9$, $Y_{(5)} = 20$. Potom použitím vyššie spomenutých trimmed mean filtrov získavame nasledujúce výsledky:

- $(2, 1)$-fold trimmed mean filter: $(7 + 9)/2 = 8$;

- $(1, 2)$-winsorized mean filter: $2 \cdot 5 + 3 \cdot 7)/5 = 6.2$;

- 0.2-trimmed mean filter: $(5 + 7 + 9)/3 = 7$;

- 0.2-winsorized mean: $(2 \cdot 5 + 7 + 2 \cdot 9)/5 = 7$.

Do skupiny trimmed mean filtrov patri aj tak zvaný *modifikovaný trimmed mean filter* (MTM) [10]. Daný filter je založený na voľbe konštanty $q \in R$. Po zadefinovaní konštanty q a zoradení hodnôt vstupujúcich do okna filtrovania, elementy, ktoré patria do intervalu $[Y_{(k+1)} - q, Y_{(k+1)} + q]$, sa spriemerujú. Daný priemer potom predstavuje výstup MTM filtra v danom časovom bode. Formulácia tohto filtra vyzerá nasledovne [8]:

$$MTM(Y_1, Y_2, \ldots, Y_N; q) = \frac{\sum_{i=1}^{N} a_i Y_i}{\sum_{i=1}^{N} a_i},$$

kde

$$a_i = \begin{cases} 1, & ak|Y_i - Y_{(k+1)}| \leq q, \\ 0, & inak. \end{cases}$$

Zovšeobecnením trimmed mean filtrov sú tak zvané *L-filtre*, ktoré sú popísané v nasledujúcej časti.

2.4 L-filtre

Na zlepšenie fungovania a odstránenie istých problémov obyčajného mediánového filtra boli vyvinuté *L-filtre*. Tieto filtre sú založené na váženom priemere dát vstupujúcich do okna filtra, ale na rozdiel od kernel metód, popísaných v prvej kapitole, sa váhy priradia dátam až po ich zoradení od najmenších po najväčšie. Množina váh určuje vlastnosti a charakteristiky príslušného L-filtra. Ak si označíme vektor váh L-filtra ako $w = (w_1, w_2, \ldots, w_N)$, vektor vstupov ako $y = (Y_1, Y_2, \ldots, Y_N)$, tak daný filter má nasledovnú formuláciu [8]:

$$L(Y_1, Y_2, \ldots, Y_N; w) = \sum_{i=1}^{N} w_i Y_{(i)}. \tag{13}$$

Na príklade dát $y = \{2, 7, 9, 20, 5\}$ sme ukázali výstup L-filtra pre rôzne vektory váh:

- $w = (0.5, 0, 0, 0, 0.5)$; $L(2, 7, 9, 20, 5; w) = 0.5 \cdot 2 + 0.5 \cdot 20 = 11$;

- $w = (0, 0.1, 0.8, 0.1, 0)$; $L(2, 7, 9, 20, 5; w) = 0.1 \cdot 5 + 0.8 \cdot 7 + 0.1 \cdot 9 = 7$;

- $w = (-0.1, 0, 1.2, 0, -0.1)$; $L(2, 7, 9, 20, 5; w) = -0.1 \cdot 2 + 1.2 \cdot 7 - 0.1 \cdot 20 = 6.2$.

Je vidieť, že druhý vektor váh zodpovedá $(1, 1)$ – fold trimmed mean filtru, lebo prvá a posledná váha je nulová. Dá sa ukázať, že pri vhodne zvolenom vektore váh sa L-filter transformuje na mediánový, mean, (r, s) – fold trimmed mean, α-trimmed mean, alebo winsorized mean filter.

L-filtre, ktoré spĺňajú vzťah:

$$\sum_{i=1}^{N} w_i = 1,$$

sa taktiež nazývajú *vyhladzujúce (smoothing) L-filtre* [8]. Uvedená podmienka musí byť splnená na to, aby sme pri počítaní váženého priemeru signálu, ktorý má konštantnú hodnotu, vo výsledku dostali signál s rovnakou hodnotou.

Veľkou výhodou týchto filtrov je to, že pri známom pravdepodobnostnom rozdelení šumu sa dajú zvoliť váhy tak, že daný L-filter bude optimálnym v zmysle strednej kvadratickej odchýlky.

2.5 C-filtre

Mediánovy filter a taktiež aj L-filtre, prebraté v predošlej časti, nezohľadňovali pôvodnú polohu dát. Výstup filtrovania je v tomto prípade ovplyvnený len umiestnením dát vstupujúcich do okna filtra po zoradení od najmenších po najväčšie. Takéto správanie filtrov môže viesť k určitým problémom pri zachovaní dôležitých detailov pôvodného signálu, hlavne pri zvyšovaní dĺžky okna filtrovania. Uvažujme napríklad dva signály nasledujúceho tvaru:

$$y_1 = (10, 17, 20, 17, 10, 0, -10, -17, -20, -17, -10, 0, 10);$$
$$y_2 = (-20, -17, -17, -10, -10, 0, 0, 10, 10, 10, 17, 17, 20).$$

Signál y_1 má tvar sínusoidy a y_2 predstavuje rastúci signál schodovitého tvaru. Hoci dané signály sú tvarovo kompletne odlišní, zodpovedajúce zoradené vektory sú identické. Vektor zoradených hodnôt pre obidva signály vyzerá nasledovne:

$$y = (-20, -20, -17, -17, -10, -10, 0, 0, 10, 10, 10, 17, 17, 20).$$

Rovnaký vektor zoradených hodnôt dostaneme aj pre hocijakú inú permutáciu vektorov y_1 alebo y_2. Potom aj výstupy filtrov, ktoré nezohľadňujú pôvodnú polohu dát, budú rovnaké. Jednou z metód, ktorá dáva dôraz nielen na zoradenie dát podľa veľkosti, ale aj na ich poradie v čase, je tak zvaný *C-filter* [11], alebo kombinovaný filter.

Daný filter sa dá chápať dvoma rôznymi spôsobmi:

- C-filter je L-filter, ktorého váhy priradené hodnotám $Y_{(i)}$ sú závislé na umiestnení príslušných hodnôt v čase;

- C-filter je vážený priemer hodnôt Y_i, vstupujúcich do okna filtrovania, kde váhy sú závislé na umiestnení daných hodnôt po zoradení.

Výstup C-filtra má potom nasledovnú formuláciu [8]:

$$C(Y_1, Y_2, \ldots, Y_N; C) = \sum_{i=1}^{N} c(R(Y_i), i) Y_i, \tag{14}$$

kde $C = [c(i, j)]$ je $N \times N$ matica váh a $R(Y_i)$ je poradové číslo hodnoty Y_i v zoradenej postupnosti vstupných hodnôt. Je vidieť, že váhy priradené vstupným hodnotám sú závislé na zoradení dát. Pre každú permutáciu vstupných hodnôt sa zvolia váhy z matice C a vážený súčet dát vygeneruje výstup filtra v danom časovom bode. Veľmi dôležitou úlohou pri používaní daného filtra je rozumná voľba matice koeficientov. Z formulácie (14) taktiež vidno, že súčet váh priradených dátam pri počítaní výstupu filtra v konkrétnom časovom bode sa nemusí rovnať 1. Tento problém sa dá vyriešiť tým, že v každom kroku filtrovania sa spočíta súčet váh, ktorým sa potom znormalizuje výstup filtra. Potom formulácia *normalizovaného C-filtra* vyzerá nasledovne [8]:

$$NormC(Y_1, Y_2, \ldots, Y_N; C) = \frac{\sum_{i=1}^{N} c(R(Y_i), i) Y_i}{\sum_{i=1}^{N} c(R(Y_i), i)}. \tag{15}$$

Na príklade jednoduchého signálu $y = (2, 7, 9, 20, 5)$ a matice koeficientov C, ktorá má nasledovný tvar:

$$\begin{pmatrix} \mathbf{0.00} & 0.05 & 0.05 & 0.05 & 0.00 \\ 0.10 & 0.15 & 0.15 & 0.15 & \mathbf{0.10} \\ 0.50 & \mathbf{0.65} & 0.80 & 0.65 & 0.50 \\ 0.10 & 0.15 & \mathbf{0.15} & 0.15 & 0.10 \\ 0.00 & 0.05 & 0.05 & \mathbf{0.05} & 0.00 \end{pmatrix}$$

ukážeme princíp fungovania C-filtra.

V prvom rade sa vytvorí vektor rankov zodpovedajúci danému vektoru y, ktorý sme označili ako r. Čiže $r = (1, 3, 4, 5, 2)$, čo znamená že prvá hodnota vektora y je najmenšia, druhá hodnota bude na treťom meste po zoradení dát patriacich do y a tak ďalej. Hodnoty vektora r ukazujú na riadok matice C, z ktorého sa priradí váha príslušnej hodnote vektora y. Umiestnenie danej hodnoty vo vektore y potom určuje stĺpec koeficientov matice C.

V našom prípade sa výstup C-filtra zráta nasledovným spôsobom:

$$C(2, 7, 9, 20, 5; C) = 2 \cdot c(1, 1) + 7 \cdot c(3, 2) + 9 \cdot c(4, 2) + 20 \cdot c(5, 4) + 5 \cdot c(2, 5) =$$
$$0 \cdot 2 + 0.65 \cdot 7 + 0.15 \cdot 9 + 0.05 \cdot 20 + 0.10 \cdot 5 = 7.4.$$

Po predelení danej hodnoty súčtom použitých váh dostaneme výstup normalizovaného C-filtra. Súčet váh sa rovná:

$$0 + 0.65 + 0.15 + 0.05 + 0.10 = 0.95.$$

Potom dostávame:

$$NormC(2, 7, 9, 20, 5; C) = \tfrac{7.4}{0.95} \approx 7.79.$$

Ďalšiu vlastnosť C-filtra uvadzáme vo Vete 2.2, ktorú sme naformulovali z príkladu zo zdroja [8] (str.254). Nami spravené riešenie je potom dôkazom tejto vety.

Veta 2.2 (O nenulovosti váh)**.** *Nech C je $N \times N$ matica koeficientov. Potom množina váh zodpovedajúca ľubovoľnému vstupnému vektoru bude obsahovať aspoň jednu nenulovú hodnotu za podmienky, že daná matica C obsahuje podmaticu $n \times m$ s nenulovými koeficientami, kde $n + m = N + 1$.*

Dôkaz. V každom kroku filtrovania sa používa iba jeden koeficient z každého riadku a stĺpca matice C. Predpokladáme, že matica C obsahuje nulové elementy všade, okrem nenulovej podmatice rozmeru $n \times m$. Čiže počet nulových riadkov je $N - n$, a počet nulových stĺpcov sa rovná $N - m$. Potom najväčší možný počet nulových váh priradených dátam zodpovedá počtu nulových riadkov a stĺpcov spolu:

$$(N - n) + (N - m) = 2N - (n + m)$$

Použitím podmienky $n + m = N + 1$ dostávame:

$$2N - (n + m) = 2N - (N + 1) = N - 1.$$

Celkový počet dát vstupujúcich do filtra je N, čiže aspoň jedna váha bude určite nenulová.

$\square$

Matice tohto typu sa často používajú pri odstraňovaní šumu z dát pomocou C-filtrov.

V tejto časti sme popísali filter, ktorého výhodou oproti L-filtrom a ostatným metódam používajúcich zoradenie dát bolo to, že sa brala do úvahy informácia o pôvodnej polohe dát v čase. S podobným problémom sa stretávame aj pri obyčajnom mediánovom filtri, modifikovanou verziou ktorého je takzvaný *vážený mediánový filter*, popísaný v ďalšej časti.

2.6 Vážený mediánový filter

Pri používaní obyčajného mediánového filtra sa častokrát narážame na situácie, keď sa relevantné detaily pôvodného signálu strácajú. Je to dôsledkom toho, že všetky vstupné dáta majú na výstup filtra v danom kroku rovnaký vplyv. Použitím *váženého mediánového filtra* [12] sa dá zdôrazniť vplyv dát, ktoré sa nachádzajú na určitých miestach okna filtrovania. Obvykle sa dáva väčší dôraz centrálnej hodnote okna, čiže hodnote, pre ktorú sa v danom časovom bode ráta výstup filtra, alebo aj vedľajším dátam. Inými slovami idea, na ktorej je založený daný filter, je priradiť niektorým dátam väčšiu váhu pri hľadaní mediánu z hodnôt okna. V prípade tohto filtra sa to dá dosiahnuť tak, že hodnoty, ktoré sú na určitých pozíciách v okne filtrovania, sa zopakujú niekoľkokrát a následne sa spočíta medián. Počet opakovaní je závislý od váh w_i, ktoré sú kladné celé čísla.

Operáciu opakovania sme označili ako $\Diamond$. Čiže, $r\Diamond y = y, \ldots, y$ predstavuje postupnosť hodnôt y, kde hodnota y sa zopakovala r-krát. Napríklad postupnosť čísel $\{1, 1, 1, 2, 3, 3\}$ sa dá prepísať ako $\{3\Diamond 1, 2, 2\Diamond 3\}$.

Ak váhy označíme ako vektor $w = (w_1, w_2, \ldots, w_N)$, potom formulácia váženého mediánového filtra je nasledovná [8]:

$$WeightMed(Y_1, Y_2, \ldots, Y_N; w) = MED\{w_1\Diamond Y_1, w_2\Diamond Y_2, \ldots, w_N\Diamond Y_N\}. \tag{16}$$

Daný filter sa dá naformulovať aj ako optimalizačná úloha. V tomto prípade má vážený mediánový filter nasledovný tvar [8]:

$$WeightMed(Y_1, Y_2, \ldots, Y_N; w) = \arg\min_{\beta} \sum_{i=1}^{N} w_i |Y_i - \beta|, \tag{17}$$

inými slovami váženým mediánom hodnôt $(Y_1, Y_2, \ldots, Y_N)$ je hodnota β, ktorá minimalizuje súčet

$$\sum_{i=1}^{N} w_i |Y_i - \beta|.$$

Na to, aby sa ukázala ekvivalencia formulácií (16) a (17), je potrebné dokázať, že riešením optimalizačnej úlohy (17) je medián M množiny $\{w_1 \Diamond Y_1, w_2 \Diamond Y_2, \ldots, w_N \Diamond Y_N\}$. Počet hodnôt v danej množine sme označili ako n. Zrejme platí, že tento počet spĺňa nerovnosť $n \geq N$, čiže pri dôkaze budeme uvažovať množinu dát $\{Y_1, Y_2, \ldots, Y_n\}$. Pre jednoduchosť budeme predpokladať, že dané hodnoty sú usporiadané nasledovne:

$$Y_1 \leq Y_2 \leq \ldots \leq Y_n.$$

Najprv uvažujme prípad, keď n je nepárne číslo, čiže $n = 2l + 1$, pre nejaké $l \in N$. Pri dôkaze sa použije trojuholníková nerovnosť, z ktorej vyplýva, že pre ľubovoľné tri čísla a, b, c platí:

$$|a - b| + |a - c| \geq |c - b|,$$

pričom rovnosť sa nadobúda práve vtedy, keď $b \leq a \leq c$. Pre ľubovoľné číslo β potom platí:

$$\sum_{i=1}^{2l+1} |\beta - Y_i| = |\beta - Y_1| + |\beta - Y_2| + \ldots + |\beta - Y_{2l+1}| =$$
$$(|\beta - Y_1| + |\beta - Y_{2l+1}|) + (|\beta - Y_2| + |\beta - Y_{2l}|) + \ldots + (|\beta - Y_l| + |\beta - Y_{l+2}|) + |\beta - Y_{l+1}| \geq$$
$$|Y_1 - Y_{2l+1}| + |Y_2 - Y_{2l}| + \ldots + |Y_l - Y_{l+2}| + |\beta - Y_{l+1}|.$$

Vychádzajúc z trojuholníkovej nerovnosti vieme povedať, že rovnosť sa nadobudne vtedy, keď za β dosadíme medián $M = Y_{l+1}$. Daná hodnota sa nachádza medzi každou z dvojíc Y_1 a Y_{2l+1}, ..., Y_l a Y_{l+2}. Čiže minimum funkcie $\sum_{i=1}^{N} w_i |Y_i - \beta|$ sa nadobúda pri $\beta = M = Y_{l+1}$, čiže mediáne hodnôt $\{Y_1, Y_2, \ldots, Y_n\}$.

V prípade, keď n je párne číslo, $n = 2l$, sa analogickým spôsobom sa ukáže, že riešením optimalizáčnej úlohy je ľubovoľné číslo medzi Y_l a Y_{l+1} vrátane mediánu $M = (Y_l + Y_{l+1})/2$.

Na hlbšie pochopenie charakteru váženého mediánového filtra a preukázanie niektorých vlastností danej metódy sme taktiež vyriešili príklady z [8] (str. 254).

Príklad. Nech $h(.)$ je monotónna funkcia. Ukážte, že platí:

$$h(WeightMed(Y_1, Y_2, \ldots, Y_N; w)) = WeightMed(h(Y_1), h(Y_2), \ldots, h(Y_N); w).$$

Riešenie.

Podľa definície monotónne rastúcej funkcie platí nasledujúca implikácia:

$$x_1 < x_2 \Rightarrow f(x_1) < f(x_2).$$

Podobne pre monotónne klesajúcu funkciu platí:

$$x_1 < x_2 \Rightarrow f(x_1) > f(x_2).$$

Aplikovaním váh w a operátora $\Diamond$ dostávame postupnosť hodnôt $Y_1, Y_2, \ldots, Y_n$, $n \geq N$, z ktorých sa následne podľa formulácie (16) zráta medián. Predpokladáme, že hodnoty $Y_1, Y_2, \ldots, Y_n$ sú v nasledovnom vzťahu:

$$Y_1 \leq Y_2 \leq \ldots \leq Y_{med} \leq \ldots \leq Y_n.$$

Aplikovaním na dané hodnoty monotónne rastúcej funkcie a použitím definície dostávame:

$$h(Y_1) \leq h(Y_2) \leq \ldots \leq h(Y_{med}) \leq \ldots \leq (Y_n),$$

čiže poradie sa nezmení. Potom vieme zapísať nasledujúcu rovnosť:

$$h(WeightMed(Y_1, Y_2, \ldots, Y_N; w)) = h(MED\{Y_1, \ldots, Y_{med}, \ldots, Y_n\}) = h(Y_{med}) =$$
$$MED\{h(Y_1), \ldots, h(Y_{med}), \ldots, h(Y_n)\} = WeightMed(h(Y_1), h(Y_2), \ldots, h(Y_N); w).$$

V prípade, že funkcia $h(.)$ je monotónne klesajúca, dostaneme rovnaký výsledok. Z definície monotónne klesajúcej funkcie vyplýva:

$$Y_1 \leq Y_2 \leq \ldots \leq Y_{med} \leq \ldots \leq Y_n \Rightarrow h(Y_1) \geq h(Y_2) \geq \ldots \geq h(Y_{med}) \geq \ldots \geq (Y_n),$$

čiže po aplikovaní funkcie $h(.)$ sa poradie hodnôt preklopí, ale centrálna hodnota, ktorá je mediánom danej postupnosti, sa nezmení.

3 Regularizácia a Hodrick-Prescottov filter

Užitočným nástrojom pri odstraňovaní šumu z dát môže byť aj optimalizácia. Jednou z takých metód, ktorá sa dá aplikovať na vyčistenie signálov, je takzvaná *regularizačná aproximácia*, ktorá je úlohou konvexnej optimalizácie. Na popis tejto metódy je potrebné zaviesť niektoré pojmy súvisiace s danou problematikou.

Vo všeobecnosti má úloha regularizačnej aproximácie nasledovný tvar [13]:

$$min\ (\|Ax - b\|, \|x\|). \tag{18}$$

Vidíme, že v tejto úlohe sa minimalizujú naraz dve účelové funkcie v tvare vektorových noriem. Normy sa dajú voliť ľubovoľne. Daná úloha je určitým rozšírením takzvanej *úlohy normovanej aproximácie*, čiže úlohy tvaru [13]:

$$min\ \|Ax - b\|, \tag{19}$$

kde $A \in R^{m\times n}$ a $b \in R^m$ sú známe dáta, $x \in R^n$ je premenná a $\|.\|$ je norma na priestore R^m. Riešenie optimalizačnej úlohy (19) sa niekedy nazýva približným riešením $Ax \approx b$ v norme $\|.\|$. Vektor odchýlok (alebo rezíduí) sa označuje nasledovne:

$$r = Ax - b.$$

Optimalizačná úloha (19) je úlohou konvexnej optimalizácie a je riešiteľná [13]. Táto úloha vždy má aspoň jedno optimálne riešenie. Hodnota účelovej funkcie sa rovná nule iba vtedy, keď $b \in S(A)$, kde $S(A)$ označuje stĺpcový priestor matice A. Daný problém je ale zaujímavejší a užitočnejší, keď $b \notin S(A)$. Obvykle sa predpokladá, že stĺpce matice A sú nezávislé a platí, že $m \geq n$. Keď $m = n$, tak optimálne riešenie úlohy (19) má analytické vyjadrenie v tvare $A^{-1}b$ a môžeme predpokladať, že $m > n$.

Pri uvedenom probléme sa obvykle stretávame s l_1 a l_2 normami. V prípade l_1 normy, ktorá je definovaná ako $\|y\|_1 = |y_1| + \ldots + |y_n|$, kde $y \in R^n$, optimalizačná úloha (19) nadobúda nasledovný tvar:

$$min\ \|Ax - b\|_1 = |r_1| + \ldots + |r_m|.$$

Daná úloha sa dá naformulovať ako úloha lineárneho programovania v nasledovnej forme [13]:

$$min\ \mathbf{1}^T t$$
$$-t \leq Ax - b \leq t,$$

kde $x \in R^n$ a $t \in R^m$ sú premenné.

Použitím l_2 normy v úlohe (19), ktorá je známa aj ako Euklidova norma a je definovaná ako: $\|y\|_2 = \sqrt{y_1^2 + \ldots + y_n^n}$, kde $y \in R^n$, sa dostávame k optimalizačnému problému nasledovného tvaru:

$$min\ \|Ax - b\|_2^2 = r_1^2 + r_2^2 + \ldots + r_m^2.$$

Táto úloha predstavuje súčet štvorcov rezíduí. Daný problém sa dá vyriešiť analyticky. Po vyjadrení účelovej funkcie ako konvexnej kvadratickej funkcie tvaru [13]

$$f(x) = x^T A^T A x - 2b^T A x + b^T b$$

sa hľadá minimum danej funkcie. Bod x minimalizuje f, ak je splnená nasledujúca rovnosť:

$$\bigtriangledown f(x) = 2A^T A x - 2A^T b = 0.$$

Potom x musí spĺňať:

$$A^T A x = A^T b.$$

Uvedená rovnosť vždy má riešenie. Ak navyše platí, že matica A má nezávislé stĺpce, problém má jediné riešenie $x = (A^T A)^{-1} A^T b$ [13].

Úloha regularizácie (18) sa dá naformulovať ako optimalizačná úloha, ktorá predstavuje minimalizáciu váženého súčtu dvoch účelových funkcií predstavených v (18). Príslušná úloha vyzerá nasledovne [13]:

$$min\ \|Ax - b\| + \gamma \|x\|, \tag{20}$$

kde $\gamma > 0$ je váha priradená norme vektora x. Pri použití Euklidovej l_2 normy sa obvykle používa iná forma regularizačnej úlohy, ktorou je minimalizácia váženého súčtu štvorcov noriem:

$$min\ \|Ax - b\|_2^2 + \gamma \|x\|_2^2, \tag{21}$$

Postupnou zmenou parametra γ v intervale $(0, \infty)$ a nasledovným výpočtom optimalizačnej úlohy (20) sa dá nakresliť trade-off krivka predstavujúca optimálne hodnoty

$\|Ax - b\|$ a $\|x\|$ v závislosti od hodnoty parametra γ. Daná krivka potom ukazuje na to, aká veľká má byť jedna z účelových funkcií na to, aby druhá bola malá.

Idea regularizácie v tvare (20) má aj určité rozšírenia. Tak napríklad, pri takzvanej *regularizácii vyhladenia* sa norma vektora x, ktorá vyjadruje veľkosť riešenia optimalizačnej úlohy, nahradí funkciou $\|Dx\|$. Potom dostávame úlohu tvaru [13]:

$$min\ \|Ax - b\| + \gamma\|Dx\|, \tag{22}$$

kde matica D predstavuje operátor prvých alebo druhých diferencií optimálneho riešenia x. Čiže v optimalizačnej úlohe (22) určuje $\|Dx\|$ mieru variancie alebo mieru hladkosti optimálneho riešenia x.

Predstavme si, že vektor $x \in R^n$ predstavuje hodnoty nejakej fyzikálnej veličiny, napríklad teploty, pozdĺž intervalu $[0, 1]$. Hodnota x_i je teplota v bode i/n. Aproximácia gradientu alebo prvej derivácie parametra v okolí bodu i/n sa dá vyjadriť v tvare $n(x_{i+1} - x_i)$, a aproximácia druhej derivácie je vyjadrená v tvare druhých diferencií:

$$n(n(x_{i+1} - x_i) - n(x_i - x_{i-1})) = n^2(x_{i+1} - 2x_i + x_{i-1})$$

Matica D v optimalizačnej úlohe (22) je teda trojdiagonálna matica nasledovného tvaru:

$$D = n^2 \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \end{pmatrix} \in R^{(n-2)\times n}.$$

Potom Dx reprezentuje aproximáciu druhej derivácie parametra a $\|Dx\|_2^2$ určuje stredné kvadratické zakrivenie parametra x pozdĺž intervalu $[0, 1]$.

S problémom regularizačnej aproximácie sa vieme detailnejšie oboznámiť v bakalárskej práce [14].

3.1 Aplikácia regularizácie na odstraňovanie šumu z dát

Pri rekonštrukcii zašumeného signálu sa opäť vraciame k pôvodnej formulácii:

$$Y_t = f_t + X_t,$$

kde f_t zodpovedá očistenému signálu a X_t je šum. Vo všeobecnosti sa predpokladá, že signál f_t sa nemení príliš rýchlo, čo znamená, že $f_t \approx f_{t+1}$. Po zavedení tohto predpokladu je úlohou odstrániť šum a zároveň zachovať do istej miery hladkosť pôvodného signálu. Potom v zmysle základnej formulácie úloh regularizačnej aproximácie (18) dostávame nasledujúci tvar optimalizačnej úlohy pre daný problém [13]:

$$min\ (\|\widehat{f} - Y\|_2, \|\widehat{f}\|), \tag{23}$$

kde $\widehat{f}$ je premenná reprezentujúca filtrovanú verziu signálu a Y je vopred daný zašumený signál. Výsledkom optimalizačnej úlohy rekonštrukcie signálu (23) je signál $\widehat{f}$, ktorý je blízko zašumenej verzie Y v zmysle l_2 normy a zároveň je dostatočne vyhladený, čiže hodnota $\|\widehat{f}\|$ je malá. Úloha (23) je úlohou konvexnej optimalizácie, čo znamená že na jej riešenie sa dajú použiť metódy, súvisiace s danou problematikou.

Základnou metódou rekonštrukcie signálu je takzvané *kvadratické vyhladenie*. Pri uvedenej metóde sa používa kvadratická funkcia vyhladenia, ktorá vyzerá nasledovne [13]:

$$\phi_{quad}(\widehat{f}) = \sum_{i=1}^{N-1} (\widehat{f_{i+1}} - \widehat{f}_i)^2 = \|D\widehat{f}\|_2^2,$$

kde $D \in R^{(N-1)\times N}$ je dvoj-diagonálna matica:

$$D = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix}.$$

Vychádzajúc z formulácie (20) dostávame optimalizačnú úlohu pre odstraňovanie šumu z dát v nasledovnom tvare:

$$min\ \|\widehat{f} - Y\|_2^2 + \gamma\|D\widehat{f}\|_2^2, \tag{24}$$

kde parameter $\gamma > 0$ určuje trade-off medzi hladkosťou filtrovaného výstupu $\widehat{f}$ a jeho blízkosti k zašumenému signálu z hľadiska l_2 normy. Riešenie danej optimalizačnej úlohy sa dá vyjadriť analyticky v tvare [13]:

$$\widehat{f} = (I + \gamma D^T D)^{-1} Y.$$

Dané analytické riešenie sa dá spočítať veľmi efektívne, pokým matica $I + \gamma D^T D$ je trojdiagonálna, čo v tomto prípade platí [13].

Metóda kvadratického vyhladenia funguje dobre pri rekonštrukcii signálov v prípade, že pôvodný signál bez šumu je dostatočne hladký, čiže neobsahuje výrazné okamžité skoky. Akékoľvek rýchle zmeny pôvodného signálu budú zrejme zoslabené alebo odstránené kvadratickým vyhladením.

V prípade krokových alebo aj iných signálov obsahujúcich okamžité rýchle zmeny vhodnou je metóda *rekonštrukcie totálnou variáciou.* Táto metóda je založená na funkcii vyhladenia nasledovného tvaru [13]:

$$\phi_{tv} = \sum_{i=1}^{N-1} |\widehat{f}_{i+1} - \widehat{f}_i| = \|D\widehat{f}\|_1.$$

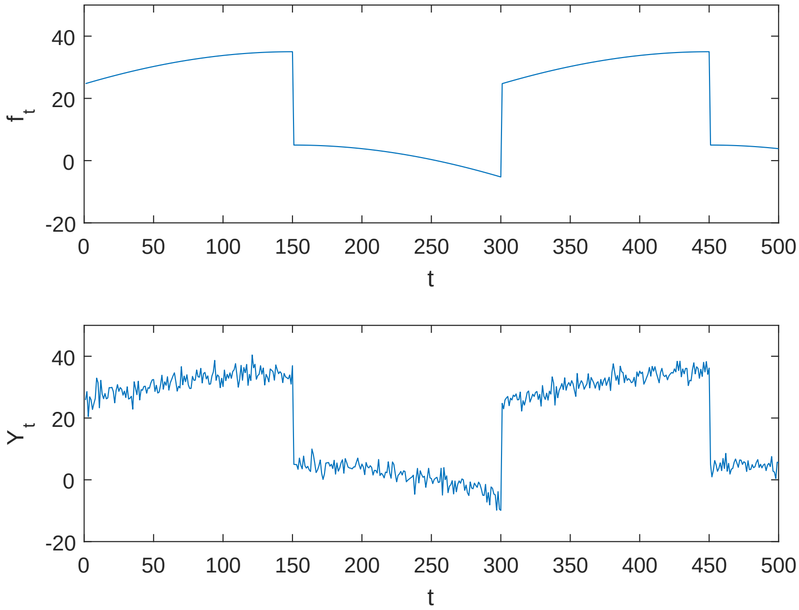
Daná funkcia je *totálnou variáciou* vektora $\widehat{f} \in R^N$ a podobne ako funkcia ϕ_{quad} zdôrazňuje veľké okamžité zmeny vo výstupnom signáli. Odlišné je to, že funkcia ϕ_{tv} pri optimalizácii priradí výrazne menšie penalty hodnotám $|\widehat{f}_{i+1} - \widehat{f}_i| > 1$ než ϕ_{quad}, čo znamená, že niektoré dôležité detaily, ako sú napríklad skoky alebo kroky, môžu byť vo vyfiltrovanom signáli zachované. Optimalizačná úloha zodpovedajúca danej funkcii vyhladenia má potom nasledovný tvar:

$$min\ \|\widehat{f} - Y\|_2^2 + \gamma\|D\widehat{f}\|_1. \tag{25}$$

Na lepšie pochopenie uvedenej metodiky porovnáme tieto dva prístupy na príklade zašumeného krokového signálu Y_t, ktorý sme uviedli na Obr.34 spolu s jeho pôvodnou očistenou verziou f_t.

Najprv sa pozrieme na riešenie daného problému metódou kvadratického vyhladenia. Riešením optimalizačnej úlohy (24) sa pre rôzne hodnoty $\gamma \in [0, \infty)$ zostrojí trade-off krivka, popisujúca závislosť veľkosti $\|\widehat{f} - Y\|_2^2$ od $\|D\widehat{f}\|_2^2$. Ako už bolo spomenuté vyššie, optimalizačná úloha (24) je úlohou konvexnej optimalizácie, čiže na jej riešenie sa dá použiť modelovací systém *CVX* [15], [16], ktorý je nadstavbou softvéru *MATLAB* a je určený na riešenie úloh konvexnej optimalizácie. Použitím nižšie uvedeného kódu:

```
cvx_begin
    variable x(n)
    minimize( norm(x - Y) + gamma*norm(D*x) )
cvx_end
```

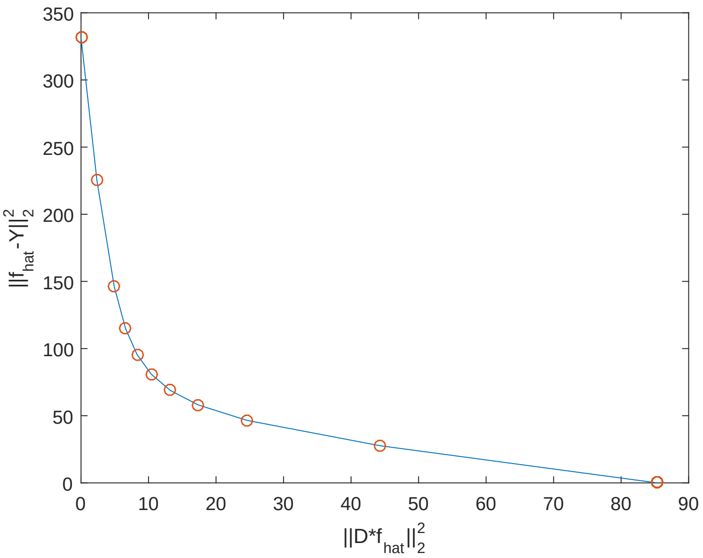


Obr. 34: Krokový signál f_t a jeho zašumená verzia Y_t.

vyriešime optimalizačnú úlohu (24). Premenná x je v tomto prípade výstupom filtrovania, čiže v zmysle formulácie (24) táto premenná zodpovedá optimálnemu filtrovanému signálu $\widehat{f}$. Na vykreslenie trade-off krivky sme použili 20 hodnôt parametra γ, presnejšie to boli hodnoty *logspace(-2, 2, 20)*. Na Obr.35 je uvedená trade-off krivka zodpovedajúca týmto hodnotám a signálu zobrazenému na Obr.34. Extrémna hodnota na pravej strane trade-off krivky zodpovedá výstupu $\widehat{f} = Y$ a hodnota účelovej funkcie je potom rovná $\|D\widehat{f}\|_2^2 = 85,3$. Extrémna hodnota na ľavej strane krivky zodpovedá výstupu $\widehat{f} = 0$, pre ktorý platí $\|\widehat{f} - Y\|_2^2 = \|Y\|_2^2 = 331,5$.

Na Obr.36 sme pre porovnanie uviedli výstupy filtrovania pre tri rôzne hodnoty $\|\widehat{f} - Y\|_2^2$, čiže aj pre 3 rôzne hodnoty parametra γ. Horný obrázok zodpovedá signálu vypočítanému pri $\gamma = 2,07$ a hodnota miery podobnosti zašumenému signálu sa v tomto prípade rovná $\|\widehat{f} - Y\|_2^2 = 58,09$. Obrázok v strede zodpovedá hodnotám $\gamma = 3,36$, $\|\widehat{f} - Y\|_2^2 = 68,82$. Dolný obrázok je pre hodnotu $\gamma = 8,86$ a $\|\widehat{f} - Y\|_2^2 = 95,50$.

Na obrázku je vidieť, že v porovnaní s pôvodným signálom znázorneným na Obr.34 sa najlepšie vyhladenie dosiahlo pri hodnote $\|\widehat{f} - Y\|_2^2 = 68,82$, ktorá zodpovedá zlomu na trade-off krivke. Výsledok filtrovania získaný pri väčšej veľkosti je príliš vyhladený, čo znamená že sa stratili dôležité detaily v tvare krokov. Naopak na hornom obrázku vidíme, že sa dostatočne veľa šumu neodstránilo pri aplikovaní kvadratického vyhladenia s takto



Obr. 35: Trade-off krivka zodpovedajúca krokovému signálu. Červené krúžky zodpovedajú výsledkom získaným pre rôzne hodnoty $\gamma = logspace(-2, 2, 20)$.

zvolenými hodnotami parametrov.

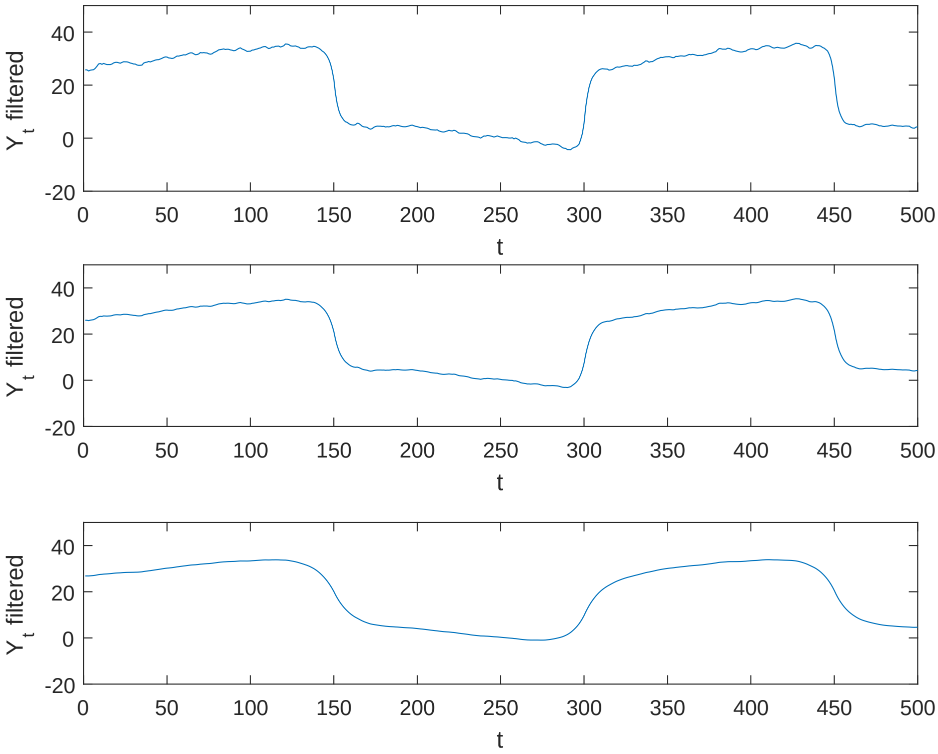
Podobným spôsobom sme ukázali fungovanie metódy rekonštrukcie signálu totálnou variáciou. Pre analýzu sme zvolili signál zobrazený na Obr.34. Na vyriešenie úlohy konvexnej optimalizácie (25) sme použili modelovací systém CVX a kód uvedený nižšie:

```
cvx_begin
    variable x(n)
    minimize( norm(x - Y) + gamma*norm(D*x, 1) )
cvx_end
```

Na Obr.37 je uvedená trade-off krivka, ktorá bola spravená rovnakým spôsobom ako pri kvadratickom vyhladení.

Obr.38 zodpovedá trom rôznym výstupom filtrovania zašumeného signálu metódou totálnej variácie, konkrétne sú to signály zodpovedajúce hodnotám $\|\widehat{f} - Y\|_2^2 = 25,45$, $\|\widehat{f} - Y\|_2^2 = 41,12$ a $\|\widehat{f} - Y\|_2^2 = 86,15$.

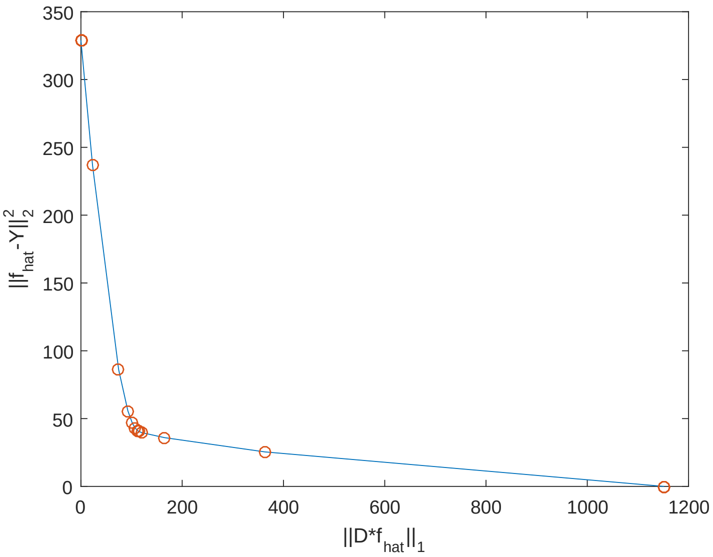
Horný signál na Obr.38 obsahuje veľké množstvo neodstráneného šumu. Dolný signál je príliš vyhladený, čiže neobsahuje pomaly sa meniace časti pôvodného signálu. Signál v strede Obr.38 zodpovedá hodnotám na zlome trade-off krivky. Dôležitou vlastnosťou metódy totálnej variácie je to, že na rozdiel od kvadratického vyhladenia zachováva



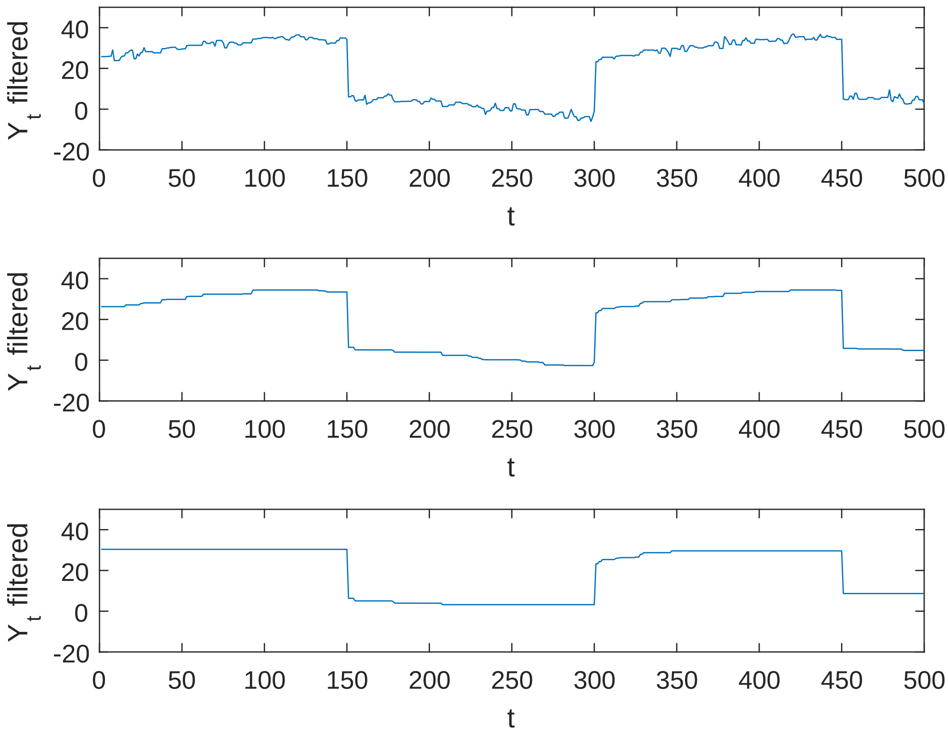
Obr. 36: Výsledky filtrovania krokového signálu metódou kvadratického vyhladenia. Horný obrázok zodpovedá signálu vypočítanému pri $\gamma = 2,07$ a $\|\widehat{f} - Y\|_2^2 = 58,09$. Obrázok v strede $\gamma = 3,36$, $\|\widehat{f} - Y\|_2^2 = 68,82$. Dolný obrázok $\gamma = 8,86$ a $\|\widehat{f} - Y\|_2^2 = 95,50$.

okamžité zmeny pôvodného signálu ako napríklad kroky.

Inou často používanou metódou na vytvorenie trendu časových radov a odstraňovanie šumu z dát je *Hodrick-Prescottov filter*, ktorý opíšeme v ďalšej časti tejto práce.



Obr. 37: Trade-off krivka zodpovedajúca krokovému signálu. Červené krúžky zodpovedajú výsledkom získaným pre rôzne hodnoty $\gamma = logspace(-2, 2, 20)$.



Obr. 38: Výsledky filtrovania krokového signálu metódou totálnej variácie. Horný obrázok zodpovedá signálu vypočítanému pri $\gamma = 0,04$ a $\|\widehat{f} - Y\|_2^2 = 25,45$. Obrázok v strede $\gamma = 0,30$, $\|\widehat{f} - Y\|_2^2 = 41,12$. Dolný obrázok $\gamma = 2,07$ a $\|\widehat{f} - Y\|_2^2 = 85,15$.

3.2 Hodrick-Prescottov filter

Hodrick-Prescottov filter, ktorý sa obvykle používa na generovanie trendu časových radov a vyhladenie dát, je taktiež regularizačnou metódou. Na rozdiel od kvadratického vyhladenia a vyhladenia pomocou totálnej variácie sa v danej metóde miera vyhladenia dát počíta pomocou druhých diferencií. Pri danom filtri odhad očisteného signálu $\widehat{f_t}$ je riešením optimalizačnej úlohy nasledovného tvaru [17]:

$$min \sum_{t=1}^{N}(Y_t - \widehat{f_t})^2 + \gamma \sum_{t=2}^{N-1}(\widehat{f_{t-1}} - 2\widehat{f_t} + \widehat{f_{t+1}})^2,$$

kde $\gamma \geq 0$ je parameter regularizácie, ktorý zodpovedá za trade-off medzi hladkosťou očisteného signálu a odchýlkou od zašumenej verzie Y_t. Všimneme si, že argument, zodpovedajúci druhému súčiniteľu danej minimalizačnej úlohy, $\widehat{f_{t-1}} - 2\widehat{f_t} + \widehat{f_{t+1}}$, je nulový iba vtedy, keď body $\widehat{f_{t-1}}, \widehat{f_t}, \widehat{f_{t+1}}$ ležia na priamke. To znamená, že druhá účelová funkcia tejto optimalizačnej úlohy je nulová v prípadoch, keď výstup filtrovania $\widehat{f_t}$ je afinnou funkciou $\widehat{f_t} = \alpha + \beta t$, kde α a β sú konštanty.

Vychádzajúc z formulácie regularizačnej úlohy (20) dostávame nasledovný tvar pre Hodrick-Prescottov filter:

$$min \|\widehat{f} - Y\|_2^2 + \gamma \|D_{HP}\widehat{f}\|_2^2, \tag{26}$$

kde matica D_{HP} je maticou druhých diferencií v nasledovnej forme:

$$D_{HP} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \end{pmatrix}.$$

Optimalizačná úloha (26) je konvexnou úlohou a podobne ako v (24) sa jej riešenie dá vyjadriť analyticky. Potom dané riešenie vyzerá nasledovne [17]:

$$\widehat{f} = (I + 2\gamma D_{HP}^T D_{HP})^{-1} Y.$$

Vidíme, že výstup Hodrick-Prescottovho filtra je lineárnou funkciou časového radu Y.

Ďalšou užitočnou vlastnosťou tejto metódy je ohraničenosť relatívnej chyby fitovania signálu Y. Relatívna chyba je zhora ohraničená a spĺňa nasledujúcu nerovnosť [17]:

$$\frac{\|Y-\widehat{f}\|_2}{\|Y\|_2} \leq \frac{32\gamma}{1+32\gamma}.$$

Uvedená nerovnosť ukazuje na to, že pri približovaní regularizačného parametra γ k nule konverguje riešenie optimalizačnej úlohy $\widehat{f}$ k pôvodnému zašumenému signálu Y.

Pri zvyšovaní regularizačného parametra γ výstup Hodrick-Prescottovho filtra konverguje k priamke, ktorá najlepšie popisuje dáta Y. To znamená, že pri $\gamma \to \infty$ konverguje výstup filtra k afinnej funkcii:

$$\widehat{f} = \alpha + \beta t,$$

kde konštanty α a β majú nasledovné vyjadrenia [17]:

$$\alpha = \frac{\sum_{t=1}^{N} t^2 \sum_{t=1}^{N} Y_t - \sum_{t=1}^{N} t \sum_{t=1}^{N} tY_t}{N \sum_{t=1}^{N} t^2 - (\sum_{t=1}^{N} t)^2},$$

$$\beta = \frac{N \sum_{t=1}^{N} tY_t - \sum_{t=1}^{N} t \sum_{t=1}^{N} Y_t}{N \sum_{t=1}^{N} t^2 - (\sum_{t=1}^{N} t)^2}.$$

4 Experiment

V predošlých častiach sme videli, že každý filter alebo metóda na odstraňovanie šumu z dát mali svoje výhody a nevýhody v závislosti od typu signálu alebo šumu. Kvalita očistenia signálu je taktiež ovplyvnená voľbou parametra h, čiže rozsahom hodnôt signálu vstupujúcich do filtra v danom časovom bode. Pri regularizačných metódach je výsledok ovplyvnený voľbou parametra γ. Optimálnosť voľby daných parametrov nie je jednoznačne daná pre konkrétne metódy, čiže obvykle hodnotu parametrov vieme len odhadnúť. Napríklad v prípade regularizácie sa parameter γ volí po vykreslení trade-off krivky a vyskúšaním niektorých hodnôt ležiacich v okolí jej zlomu.

4.1 Popis experimentu

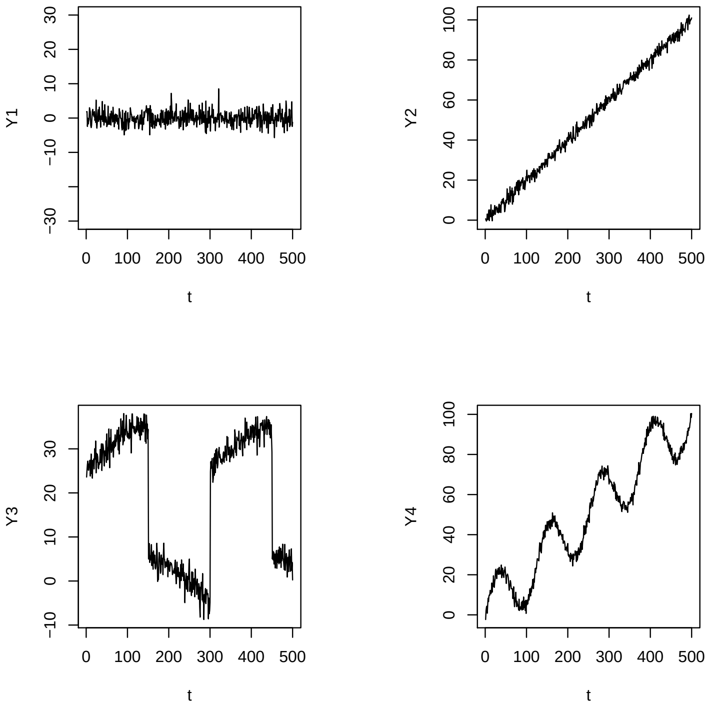
Ideou experimentu bolo porovnať kvalitu očistenia niektorých metód prebraných v tejto práci na generovaných signáloch rôzneho typu v závislosti od množstva impulzného šumu prítomného v dátach. Kvalitu očistenia zašumeného signálu sme merali pomocou priemernej kvadratickej odchýlky (MSE) očisteného signálu $\widehat{f_t}$ od jeho pôvodnej nezašumenej verzie f_t. Taktiež sme chceli experimentálne potvrdiť niektoré teoretické vlastnosti metód a zistiť, či existuje nejaká závislosť medzi množstvom impulzného šumu a optimálnou voľbou parametrov h a γ, pri ktorých sa dosahuje optimálne očistenie v zmysle MSE.

Experiment sme programovali prostredníctvom softvéru *R* a na skúmanie metód regularizácie sme používali modelovací systém *CVX*, ktorý je nadstavbou softvéru *MATLAB*. V Prílohe A je iba časť nami vytvoreného kódu kvôli jeho zdĺhavosti. Daná časť kódu predstavuje experiment s lineárnym sklonom, naprogramovaný pomocou softvéru *R*. Všetky použité filtre boli ručne naprogramované v tvare funkcií.

Skúmali sme štyri typy signálov: konštantný signál, lineárny sklon, krokový signál a rastúcu sínusoidu. Dĺžka experimentálnych signálov je rovná $N = 500$. Dané signály zašumené jednou z realizácii gaussovského šumu $N(0, 4)$ sme uviedli na Obr.39.

V experimente sme používali krokový signál, ktorý bol znázornený na Obr.34. Rastúca sínusoida je súčtom lineárneho sklonu a sínusoidy s predpisom $15sin(x)$, kde $x \in [0, 8\pi]$.

Na daných signáloch sme pozorovali správanie piatich filtrov, ktorými sú: mean filter, mediánový filter, Gaussove kernel filtre (4), (5), a taktiež modifikovaný trimmed mean



Obr. 39: Signály zašumené gaussovským šumom $N(0, 2)$. Konštantný signál (vľavo hore), lineárny sklon (vpravo hore), krokový signál (vľavo dole), rastúca sínusoida (vpravo dole).

filter s parametrom $q = 10$. Do analýzy sme zapojili aj regularizačné metódy, konkrétne metódu kvadratického vyhladenia, rekonštrukciu totálnou variáciou a Hodrick-Prescottov filter.

V každom opakovaní experimentu sa na začiatku generoval zašumený signál, ktorý je súčtom pôvodného signálu bez šumu f_t a realizácií náhodnej premennej $X_t = N(0, 4)$, čiže gaussovského šumu. Ako výsledok dostávame signál tvaru:

$$Y_t = f_t + X_t.$$

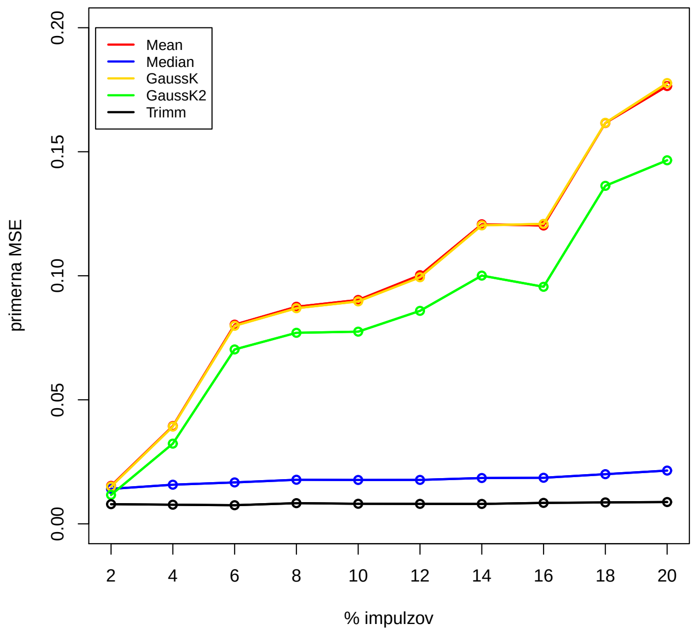
Pre každý takto vygenerovaný signál sme postupne vymieňali 2% jeho hodnôt za impulzný šum, ktorého veľkosť bola taktiež náhodná v intervale od 12 do 18. Impulzný šum sme pridávali desaťkrát, teda na konci sme mali signál obsahujúci 20% impulzov (100 dát) a 80% gaussovského šumu (400 dát). V každej fáze pridania impulzov sme aplikovali na získaný signál filter, pre ktorý sme menili parameter h v intervale od 1 do 250. Prejdením cez všetky hodnoty veľkosti okna sme získali optimálnu mieru vyhladenia daného zašumeného signálu v zmysle MSE a jej zodpovedajúcu optimálnu hodnotu parametra h. V prípade regularizačných metód sme v každej fáze používali 20 rôznych hodnôt parametra γ vygenerovaných pomocou príkazu:

```
gamma = logspace( -2, 2, 20 ).
```

Takto naprogramovaný algoritmus sme aplikovali na štyri typy experimentálnych signálov. Kvôli časovej náročnosti daného experimentu sme spravili 100 opakovaní pre každý typ signálu. Získané výsledky pre všetky typy filtrov a signálov sme potom spriemerovali a dostali sme priemerné hodnoty optimálnej MSE pre rôzne počty impulzného šumu a im zodpovedajúce priemerné optimálne hodnoty parametrov h a γ.

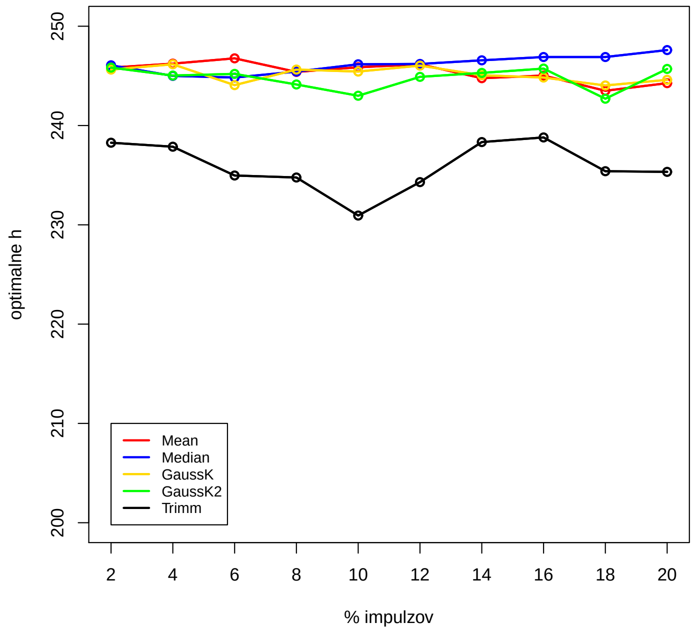
4.2 Výsledky experimentu

Získané priemery sme spracovali do grafov a uviedli v tejto časti. Na Obr.40 sú znázornené priemerné optimálne MSE pre konštantný signál a päť typov filtrov.



Obr. 40: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre konštantný signál a päť skúmaných filtrov.

Červená farba zodpovedá výsledkom pre mean filter (*Mean*), modrá farba – mediánový filter (*Median*), žltá – Gaussov kernel s oknom filtrovania, ktorý bol naformulovaný ako (5) a pre jednoduchosť označený ako *GaussK*, zelená farba zodpovedá výsledkom pre Gaussov kernel (4) (*GaussK2*), čierna – modifikovaný timmed mean filter (*Trimm*). Krúžky na grafoch zodpovedajú hodnotám získaným počas experimentu pre rôzne množstvá impulzného šumu. Podobným spôsobom sme na Obr.41 zobrazili výsledky priemerovania optimálnej hodnoty parametra h.



Obr. 41: Priemerné optimálne hodnoty parametra h v závislosti od počtu impulzov pritomného v dátach pre konštantný signál a päť skúmaných filtrov.

Z Obr.40 je vidieť, že pri zvyšovaní množstva impulzného šumu *Trimm* a *Median* filtre vykazujú najlepšiu kvalitu vyhladenia. *GaussK*, *GaussK2* a *Mean* majú podobný rastúci trend optimálnej MSE pri zvyšovaní počtu impulzov. Dokonca hodnoty optimálnej MSE pre *Mean* a *GaussK* filtre sú skoro identické.

Z Obr.41 je vidieť, že optimálna MSE sa vo väčšine prípadov dosiahla pri maximálnej možnej hodnote parametra $h = 250$. V prvej kapitole sme popísali závislosť výstupu lineárnych filtrov od zvyšovania parametra h a videli sme, že sa približovali k vodorovnej čiare, čiže konštantnému signálu. Iným vysvetlením je aj to, že šum bol generovaný z rozdelenia $N(0, 4)$, čiže čím viac dát bude vstupovať do priemeru, tým sa budeme viac približovať k strednej hodnote daného rozdelenia, ktorá zodpovedá hodnote konštantného signálu v ľubovoľnom čase. Rovnako aj pri mediánovom filtri väčšie okno filtrovania zaručuje to, že výrazne odlišné hodnoty, považované za impulzný šum, budú čo najďalej od mediánu dát, čiže neovplyvnia výstup.

Na Obr.42, Obr.44 a Obr.46 sú znázornené optimálne MSE pre tri ďalšie experimentálne signály. Vieme usúdiť, že vo všetkých prípadoch nezávislé od typu signálu sa lepšie ukázali nelineárne filtre, z ktorých najkvalitnejšie vyhladenie pri rastúcom množstve impulzného šumu zaručuje modifikovaný trimmed mean filter. Rýchlosť nárastu optimálnej MSE pri rastúcom počte impulzov v prípade skúmaných lineárnych filtrov je výrazne vyššia než rýchlosť zodpovedajúca nelineárnym metódam. Dané experimentálne výsledky nie sú v rozpore s teóriou, čiže sa potvrdil ten fakt, že v prípade impulzného šumu sú nelineárne metódy výhodnejšie.

Na Obr.43, Obr.45 a Obr.47 sú zakreslené optimálne hodnoty parametra h, zodpovedajúceho najlepšej kvalite vyhladenia pre tri zvyšné typy signálov. Nezávislé od tvaru zašumeného signálu trend optimálnych hodnôt h pri zvyšujúcom sa množstve impulzov je podobný pre lineárne metódy a taktiež aj pre nelineárne. Pri lineárnych filtroch sa okno filtrovania rozširuje pri zvyšovaní impulzného šumu. Naopak pre nelineárne filtre vidíme konštantný alebo klesajúci trend.

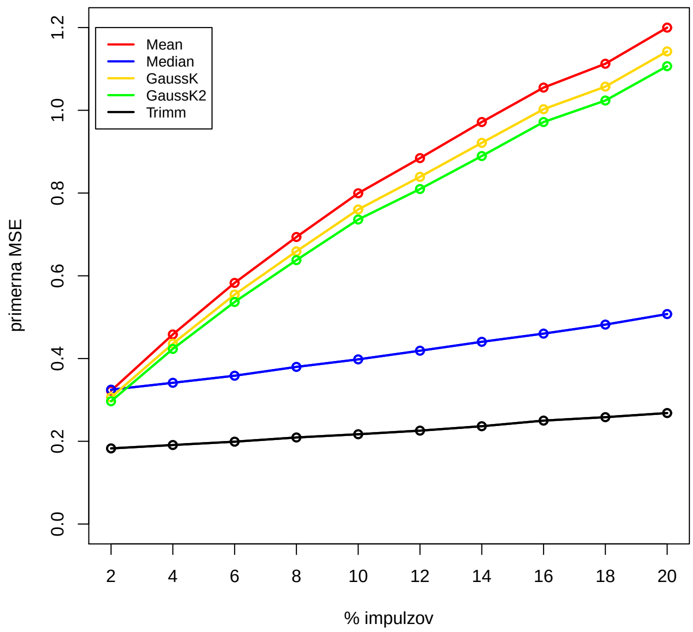
Z Obr.43 a Obr.47 vidíme, že pre lineárny sklon a rastúcu sínusoidu sme dostali podobné výsledky v zmysle trendov jednotlivých kriviek. Napríklad pri zvyšovaní počtu impulzov bol optimálny parameter h pre *Median* a *Trimm* filter zhruba na rovnakej úrovni. Pri lineárnych filtroch pozorujeme rastúci trend s hodnotami vyššími než pri nelineárnych metódach. Pri *GaussK2* filtri nastalo prekročenie optimálnych hodnôt nelineárnych metód pri 10% impulzného šumu pre obidva typy signálov.

Výsledky zodpovedajúce krokovému signálu, ktoré sú zobrazené na Obr.45, poukazujú na klesajúci trend pri nelineárnych metódach. Taktiež vidíme, že lineárne filtre dosahujú najlepšie vyhladenie v zmysle MSE pre malé hodnoty h. V prvej kapitole sme ukazovali, že čím menšie okno filtrovania sa použije pri vyhladení, tým bude výstup filtra viac podobný zašumenému signálu. Daný fakt potvrdzuje to, že rekonštrukcia krokov je problémová pre lineárne metódy, čo vidíme aj z optimálnych hodnôt MSE na Obr.44.

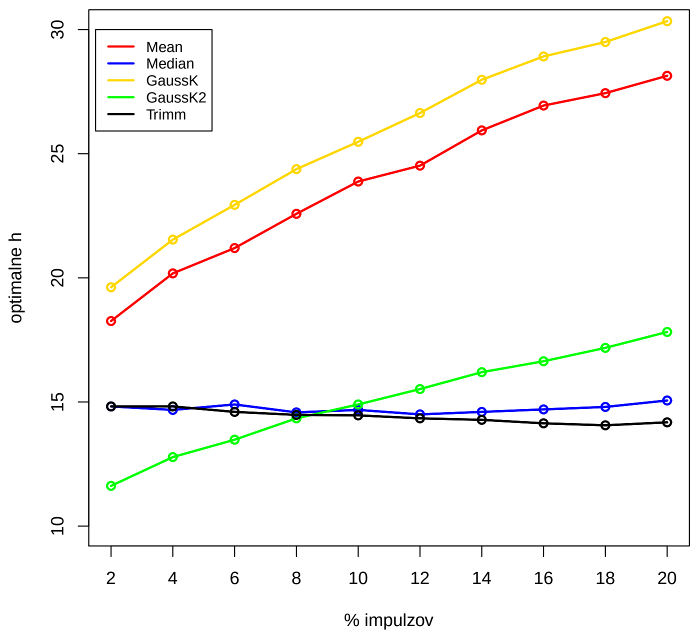
Výsledky experimentu pre regularizačné metódy a štyri typy experimentálnych signálov sme uviedli na ďalších obrázkoch. Priemerné optimálne MSE sú znázornené na Obr.48, Obr.50, Obr.52 a Obr.54. Priemerné optimálne hodnoty regularizačného parametra γ pri ktorých sa dosiahla najlepšia kvalita vyhladenia, sú uvedené na Obr.49, Obr.51, Obr.53 a Obr.55.

Pri všetkých typoch signálov vidíme nárast optimálnej MSE pri zvyšovaní množstva impulzného šumu v dátach. Z Obr.50 a Obr.54, ktoré zodpovedajú lineárnemu sklonu a rastúcej sínusoide vidíme, že najlepšia kvalita vyhladenia pre tieto dva typy signálov sa dosiahla pomocou Hodrick-Prescottovho filtra (*HP*). Najhoršie v tomto prípade dopadla metóda totálnej variácie (*L1*). Opačná situácia nastala pri konštantnom a krokovom signáloch, o čom sa vieme presvedčiť z Obr.48 a Obr.52. V prípade konštantného signálu metóda totálnej variácie a metóda kvadratického vyhladenia (*L2*) majú rovnaké priemerné optimálne hodnoty MSE v intervale od 0.0178 pri 2% impulzného šumu do 0.0929 pri 20%. Regularizáčne metódy sa preukázali najhoršie pri vyhladení krokového signálu. Z Obr.52 vidíme, že najväčšia chybovosť zodpovedá *HP* filtru, konkrétne optimálne hodnoty MSE sa menia od 3.5647 do 8.7072 pri zvyšujúcom sa počte impulzného šumu. Najlepšou je v tomto prípade *L1* metóda s hodnotami optimálnej MSE od 0.6927 do 2.6465.

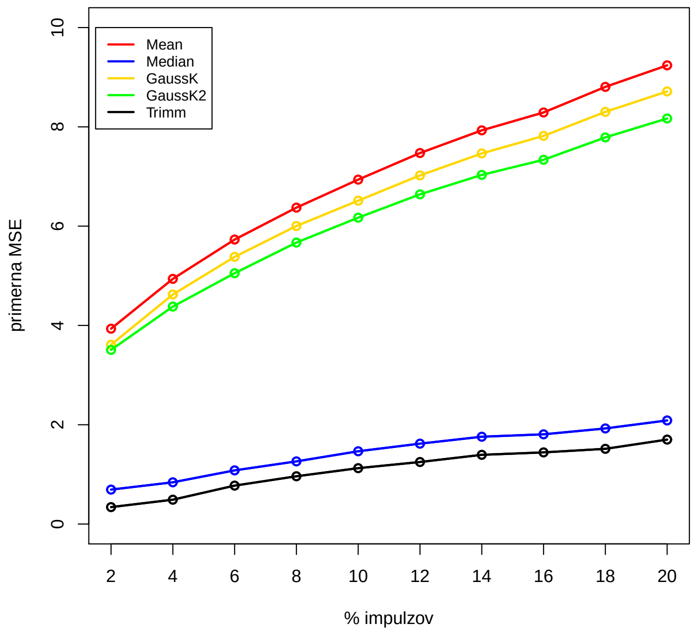
Na všetkých grafoch priemernej optimálnej hodnoty parametra γ vidíme, že najväčšie výsledky zodpovedajú *HP* filtru. Jedine pri krokovom signáli na Obr.53 pozorujeme rastúci trend pri *HP* filtri a po prekročení 8% hranici počtu impulzov, hodnoty γ pre daný filter sú väčšie než hodnoty zodpovedajúce *L2* metóde. Pri spracovaní teórie o danom filtri sme videli, že pri zvyšovaní regularizačného parametra výstup *HP* filtra konverguje k priamke, ktorá najlepšie popisuje dáta. V prípade lineárneho sklonu a konštantného signálu, ktoré majú tvar priamky, *HP* filter dosahuje najlepšie vyhladenie pri hraničnej hodnote $\gamma = 100$.



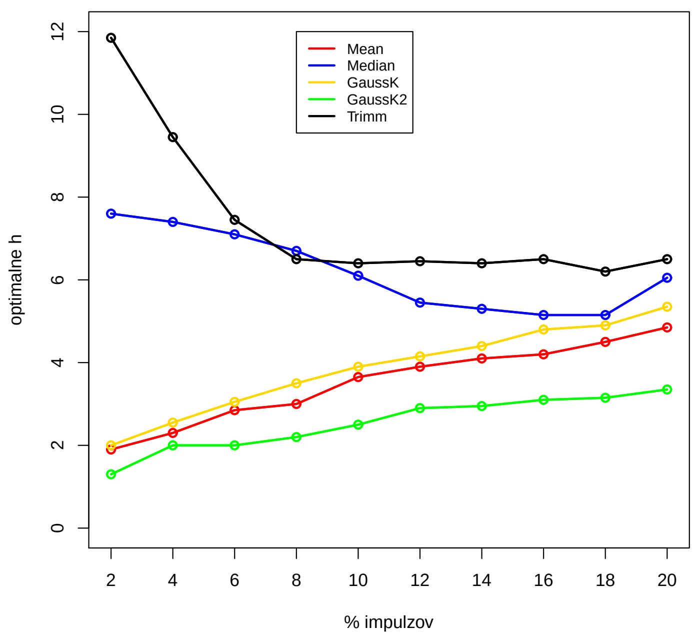
Obr. 42: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre lineárny sklon a päť skúmaných filtrov.



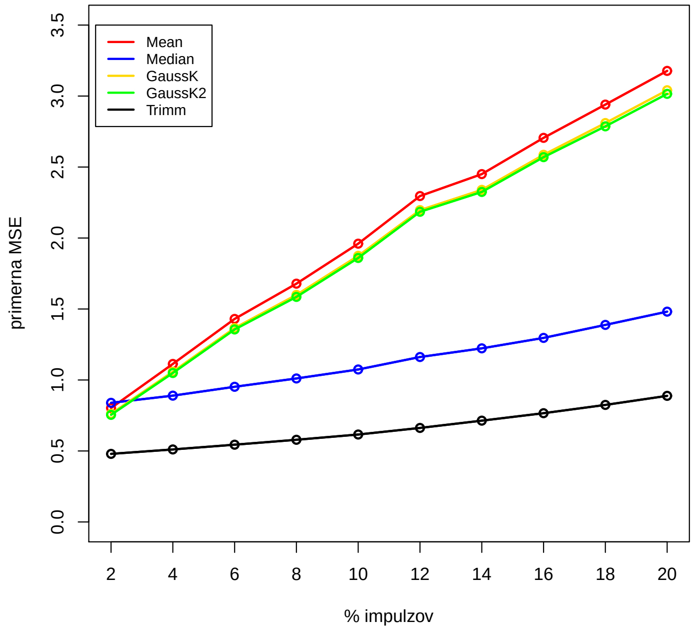
Obr. 43: Priemerné optimálne hodnoty parametra h v závislosti od počtu impulzov pritomného v dátach pre lineárny sklon a päť skúmaných filtrov.



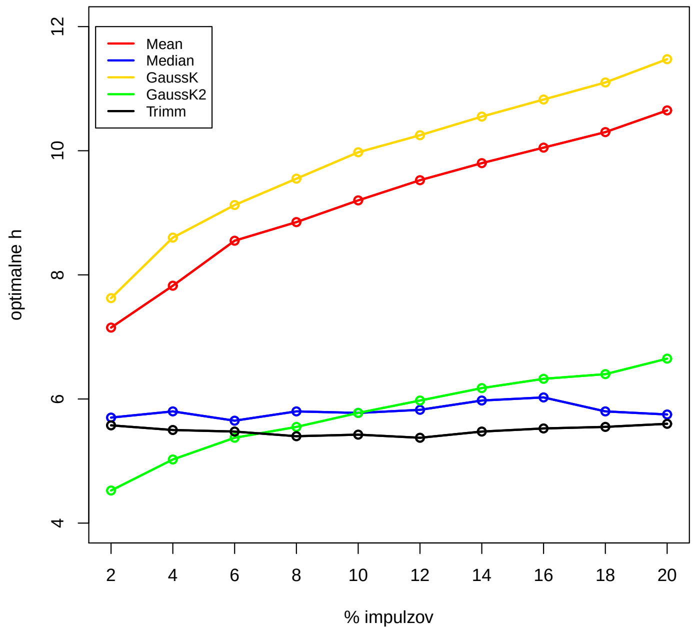
Obr. 44: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre krokový signál a päť skúmaných filtrov.



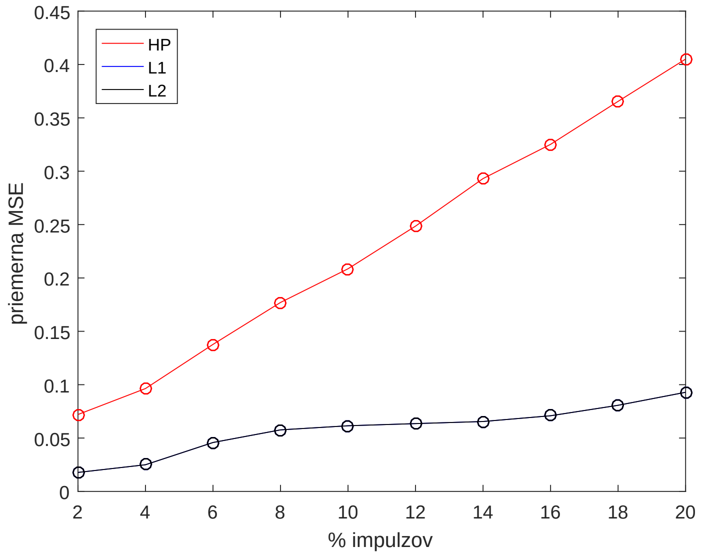
Obr. 45: Priemerné optimálne hodnoty parametra h v závislosti od počtu impulzov pritomného v dátach pre krokový signál a päť skúmaných filtrov.



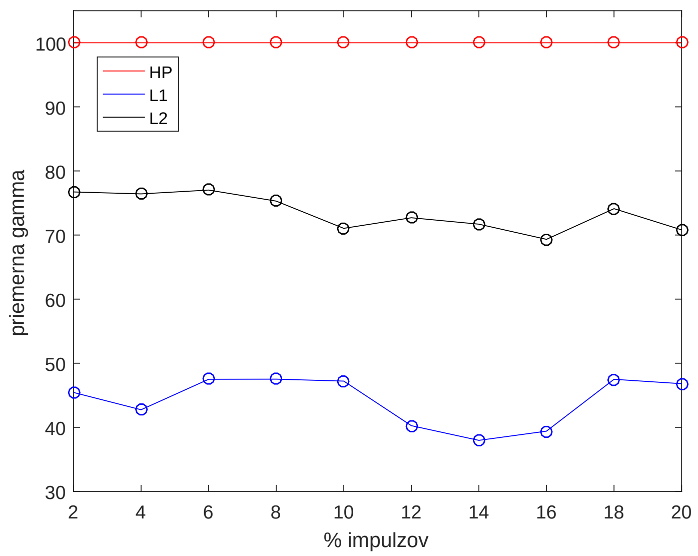
Obr. 46: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre rastúcu sínusoidu a päť skúmaných filtrov.



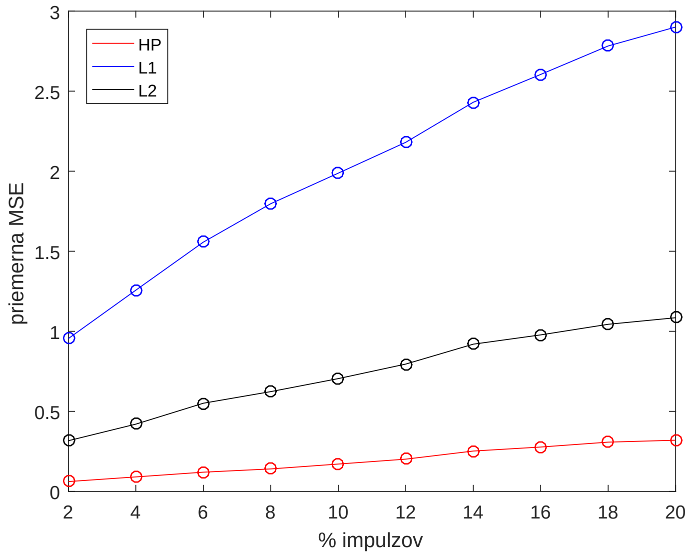
Obr. 47: Priemerné optimálne hodnoty parametra h v závislosti od počtu impulzov pritomného v dátach pre rastúcu sínusoidu a päť skúmaných filtrov.



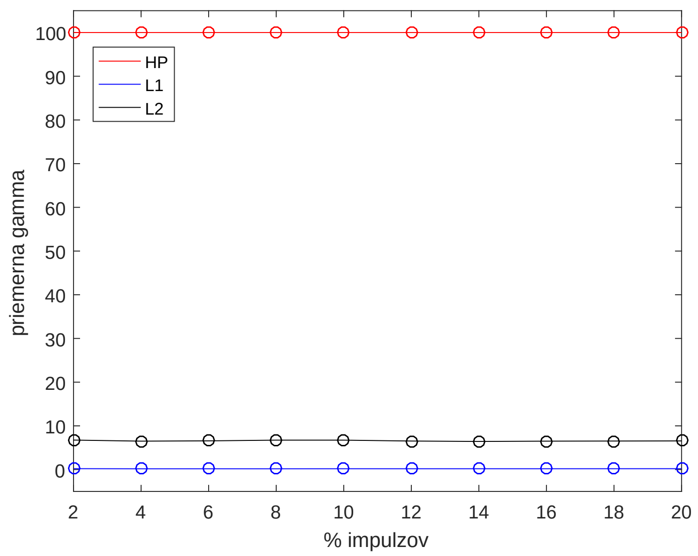
Obr. 48: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre konštantný signál a tri typy regularizáčnych metód.



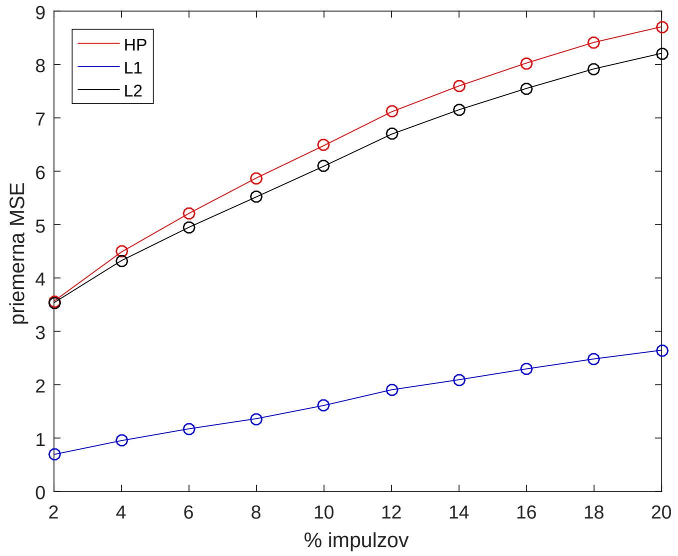
Obr. 49: Priemerné optimálne hodnoty parametra γ v závislosti od počtu impulzov pritomného v dátach pre konštantný signál a tri typy regularizáčnych metód.



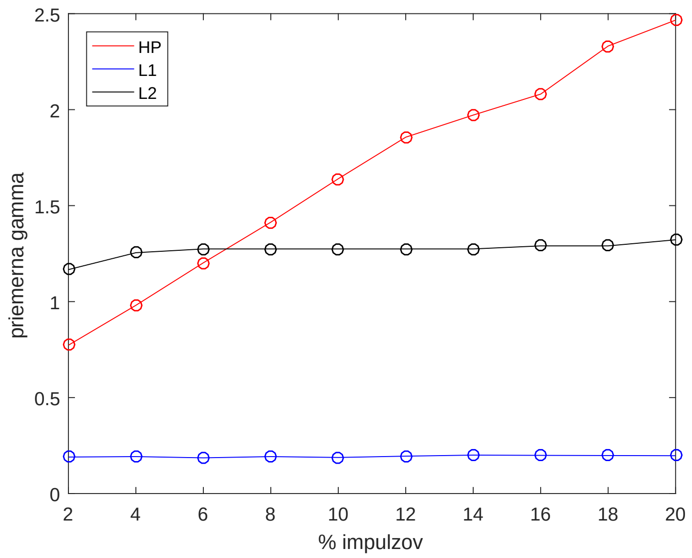
Obr. 50: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre lineárny sklon a tri typy regularizáčnych metód.



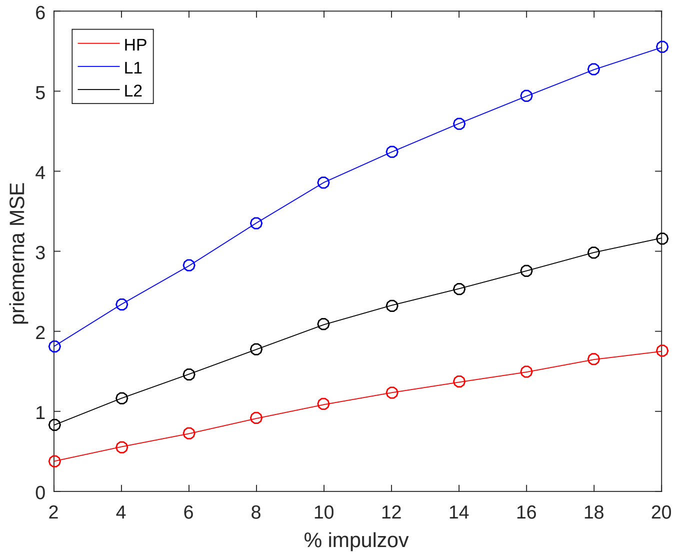
Obr. 51: Priemerné optimálne hodnoty parametra γ v závislosti od počtu impulzov pritomného v dátach pre lineárny sklon a tri typy regularizáčnych metód.



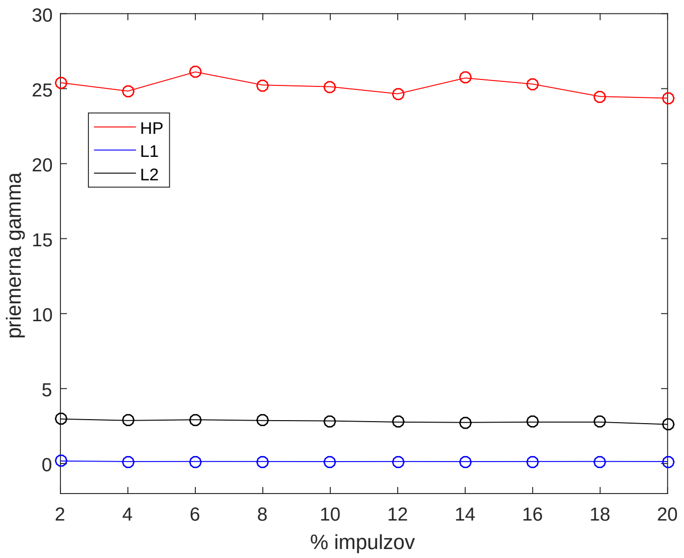
Obr. 52: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre krokový signál a tri typy regularizáčnych metód.



Obr. 53: Priemerné optimálne hodnoty parametra γ v závislosti od počtu impulzov pritomného v dátach pre krokový signál a tri typy regularizáčnych metód.



Obr. 54: Priemerné optimálne MSE v závislosti od počtu impulzov pritomného v dátach pre rastúcu sínusoidu a tri typy regularizáčnych metód.



Obr. 55: Priemerné optimálne hodnoty parametra γ v závislosti od počtu impulzov pritomného v dátach pre rastúcu sínusoidu a tri typy regularizáčnych metód.

Záver

Cieľom diplomovej práce bolo prehľadne spracovať rôzne algoritmy odstraňovania šumu z dát, využívajúc metódy z rôznych oblastí matematiky. Pri spracovaní dát sme použili softvér *R* a *MATLAB*. Tieto systémy umožnili vygenerovať vhodné dáta a porovnať na nich fungovania prebratých metód a algoritmov.

V prvej kapitole sme sa zameriavali na lineárne filtre, ktoré sú neparametrickými metódami odstraňovania šumu z dát. Uviedli sme základnú formuláciu lineárnych filtrov. Následne boli popísané najjednoduchšie, ale často používané filtre: obojstranný a jednostranný moving average algoritmus. Dané algoritmy sme naprogramovali a overili sme ich vlastnosti na dátach pomocou softvéru *R*. Tieto filtre sme taktiež porovnali s komplikovanejšími kernel metódami, ktoré sme taktiež podrobne skúmali a popisovali v danej kapitole. Ukázali sme vplyv parametrov škálovania na výstupy Gaussovho a Epanechnikovho kernelov. Pri spracovaní tejto problematiky sme vychádzali hlavne z [1] a [2]. Na konci kapitoly sme prebrali metódu exponenciálneho vyhladenia, ktorá je vylepšenou verziou jednostranného moving average algoritmu. Pri popise danej metódy sme použili hlavne zdroj [7].

Pri spracovaní druhej kapitoly sme vychádzali hlavne zo zdroja [8]. V tejto časti práce sme sa zaoberali nelineárnymi filtrami, konkrétne sme zaviedli pojem mediánového filtra, trimmed mean filtra, L-filtra, C-filtra a váženého mediánového filtra. Porovnanie lineárnych a nelineárnych metód sme spravili na príklade dvoch základných filtrov, ktorými sú mean a mediánový filter. Vďaka softvéru *R* sme otestovali správanie týchto metód v prípade zašumenia troch ukážkových signálov gaussovským alebo impulzným šumom. Ďalšie metódy prítomné v tejto kapitole sme podrobne popísali, poukázali sme na ich výhody a na spôsoby ich použitia.

V tretej kapitole sme popísali využitie optimalizácie pri probléme odstraňovania šumu z dát. Konkrétne sme sa venovali metóde regularizačnej aproximácie a jej aplikovaniu pri rekonštrukcii signálov. Informácie o tejto problematike sme čerpali hlavne z [13]. Dané metódy sú naformulované v tvare úloh konvexnej optimalizácie, preto sme na ich riešenie používali modelovací systém *CVX*. Na príklade metódy kvadratického vyhladenia a rekonštrukcie totálnou variáciou sme ukázali spôsob filtrovania zašumeného signálu, ktorý je založený na použití trade-off krivky. Skúmali sme závislosť riešenia optimalizačných úloh od hodnoty regularizačného parametra γ. Napokon sme zaviedli formuláciu Hodrick-Prescottovho filtra ako úlohy regularizačnej aproximácie a popísali sme najdôležitejšie vlastnosti danej metódy.

V poslednej časti tejto práce sme sa venovali navrhnutiu a následnému spracovaniu experimentu, v ktorom sme skúmali vplyv zvyšovania množstva impulzného šumu v dátach na optimálnu hodnotu priemernej kvadratickej odchýlky (MSE). Taktiež sme sa pozerali na hodnoty parametrov h a γ, pri ktorých daná optimálna hodnota bola dosiahnutá. Výsledky experimentu sme spracovali do grafov, ktoré sme uviedli v tejto kapitole. Zo všetkých skúmaných filtrov sa najlepšie v zmysle MSE preukázal modifikovaný trimmed mean filter. Taktiež sme usúdili, že pri všetkých skúmaných typoch experimentálnych signálov, nelineárne filtre fungujú lepšie, než lineárne. Tým sme potvrdili teoretické myšlienky o vhodnosti použití nelineárnych filtrov v prípade impulzného šumu. Pri regularizačných metódach sme zistili, že Hodrick-Prescottov filter je lepší na použitie pri signáloch typu lineárneho sklonu alebo rastúcej sínusoidy. V prípade konštantného a krokového signálov sa najlepšie ukázala metóda totálnej variácie. Pre každú použitú metódu sme taktiež odhalili charakter závislosti optimálnej hodnoty parametra h alebo γ od množstva impulzného šumu v experimentálnych signáloch.

Podľa môjho názoru, táto problematika je nielen zaujímavá, ale aj veľmi užitočná. V dnešnej dobe sa častokrát stretávame s obrovským množstvom dát, ktoré majú veľkú frekvenciu, čiže trend je ťažko rozlíšiteľný. Nami prebrané metódy umožňujú nielen odstrániť šum z dát, ale aj zachovať signifikantné výkyvy.

Zoznam použitej literatúry

[1] Sun E.W., Meinl T.: *Methods of Denoising Financial Data*, 2015, dostupné na internete: https://www.researchgate.net/publication/

[2] Yao Q., Fan J.: *Nonlinear Time Series: Nonparametric and Parametric Methods*, Springer-Verlag, New York, 2003

[3] Gasser T., Müller H.G.: *Kernel estimation of regression functions.In Smoothing Techniques for Curve Estimation*, Lecture Notes in Mathematics 757, Springer-Verlag, New York, 1979

[4] Choi E., Hall P., Rousson, V.: *Data sharpening methods for bias reduction in nonparametric regression.*, The Annals of Statistics, 28, 2000

[5] Boissard J.: Applications and Uses of Digital Filters in Finance, Swiss Federal Institute of Technology, Zurich, 2012, dostupné na internete (14.4.2017): http://www.swissquant.com/data/docs/de/1421/Applications-and-Uses-of-Digital-Filters-in-Finance.pdf

[6] Kocsisová K., 2017. Predikcia finančných časových radov pomocou moving average filtrov: diplomová práca. Bratislava : Univerzita Komenského v Bratislave.2017. 62.

[7] NIST/SEMATECH: *e-Handbook of Statistical Methods*, 2003, dostupné na internete: http://www.itl.nist.gov/div898/handbook/

[8] Astola J., Kuosmanen P.: *Fundamentals of Nonlinear Digital Filtering*, CRC Press LLC, New York, 1997

[9] Bednar J.B., Watt T.L.: Alpha-trimmed means and their relationship to median filters. In *IEEE Trans. Acoust., Speech, Signal processing.* 1984, č. 32, s. 145-153.

[10] Lee Y.H., Kassam S.A.: Generalized median filtering and related nonlinear filtering techniques. In *IEEE Trans. Acoust., Speech, Signal processing.* 1985, č. 33, s. 672-683.

[11] Kassam S.A., Peterson S.R., 1987: Nonlinear finite moving window filters for signal restoration. In *Proc. IEEE Pacific RIM Conf. on Comm. Compuer, and Signal Processing.* Victoria, Canada, 1987, s. 17-20.

[12] Brownrigg D.R.K.: The weighted median filter. In *Commun. ACM.* 1984, č. 27, s. 807-818.

[13] Boyd S., Vandenberghe L.: *Convex Optimization*, Cambridge University Press, Cambridge, 2004, dosupné na internete (30.03.2012):
http://www.stanford.edu/boyd/cvxbook/

[14] Kondratyev O., 2016. Približné riešenie systémov lineárnych rovníc: diplomová práca. Bratislava : Univerzita Komenského v Bratislave.2016. 51.

[15] Boyd S., Grant M.: *CVX: Matlab software for disciplined convex programming, version 2.0*, dostupné na internete:
http://cvxr.com/cvx

[16] Boyd S., Grant M.: *Graph implementations for nonsmooth convex programs*, Recent Advances in Learning and Control, Springer, 2008, 95-110, dostupné na internete (30.05.2012):
http://stanford.edu/boyd/graph_dcp.html.

[17] Kim S.J. a kol.: L1 Trend Filtering. In *SIAM Review.* 2009, č. 51, s. 339-360.

Príloha A

Experiment pre lineárny sklon naprogramovaný v softvéri *R*

```
########## kniznice ##########
library(pracma)
########## pocet opakovani v experimente ##########
pocet<-100
########## experiment s linearnym sklonom ##########
error_mean_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
error_GaussK_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
error_med_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
error_tream_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
error_GaussK2_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
h_mean_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
h_GaussK_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
h_med_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
h_tream_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
h_GaussK2_l<-matrix(rep(0, pocet*10), nrow = pocet,  ncol=10)
########## definovanie signalu ##########
t<-seq(1,500,by=1)
f2<-rep(1, 500)
for (i in 1:500){
  f2[i]<-t[i]*0.2
}
for (z in 1:pocet){
X2<-rnorm(500, mean = 0, sd = 2)
Y2<-f2+X2
########## rozsah impulzov ##########
a<-12
b<-20
############## pridanie impulzov ####################
for(i in 1:10) {
```

```
pos<-floor(500*runif(0.02*length(Y2), min = 0, max = 1))
impulz<-(b-a)*runif(0.02*length(Y2), min = 0, max = 1)+a
znak<-runif(0.02*length(Y2), min = 0, max = 1)
  for (n in 1:(0.02*length(Y2))) {
    if (znak[n]<0.5){
       znak[n]<- -1
       }
    else{
       znak[n]<- 1
       }
    }
    for(j in 1:(0.02*length(Y2))){
    Y2[pos[j]]<-f2[pos[j]]+impulz[j]*znak[j]
    }
############ prechod cez vsetky h pre mean ###############
error_mean_l[z,i]<-sum((f2-mean_filter(1, Y2))^2)/length(Y2)
h_mean_l[z,i]<-1
for (j in 1:250) {
if (sum((f2-mean_filter(j, Y2))^2)/length(Y2)<error_mean_l[z,i]){
    error_mean_l[z,i]<-sum((f2-mean_filter(j, Y2))^2)/length(Y2)
    h_mean_l[z,i]<-j
    }
  }
########### prechod cez vsetky h pre GaussK #################
error_GaussK_l[z,i]<-sum((f2-GaussK_filter(1, Y2))^2)/length(Y2)
h_GaussK_l[z,i]<-1
for (j in 1:250) {
if (sum((f2-GaussK_filter(j, Y2))^2)/length(Y2)<error_GaussK_l[z,i]){
    error_GaussK_l[z,i]<-sum((f2-GaussK_filter(j, Y2))^2)/length(Y2)
    h_GaussK_l[z,i]<-j
    }
```

```
  }
########### prechod cez vsetky h pre GaussK2 ################
error_GaussK2_l[z,i]<-sum((f2-GaussK2_filter(1, Y2, 1, t))^2)/length(Y2)
h_GaussK2_l[z,i]<-1
for (j in 1:250) {
if (sum((f2-GaussK2_filter(j, Y2, 1, t))^2)/length(Y2)<error_GaussK2_l[z,i]){
    error_GaussK2_l[z,i]<-sum((f2-GaussK2_filter(j, Y2, 1, t))^2)/length(Y2)
    h_GaussK2_l[z,i]<-j
    }
  }
########### prechod cez vsetky h pre MED ##################
error_med_l[z,i]<-sum((f2-med_filter(1, Y2))^2)/length(Y2)
h_med_l[z,i]<-1
for (j in 1:250) {
if (sum((f2-med_filter(j, Y2))^2)/length(Y2)<error_med_l[z,i]){
    error_med_l[z,i]<-sum((f2-med_filter(j, Y2))^2)/length(Y2)
    h_med_l[z,i]<-j
    }
  }
########### prechod cez vsetky h pre MTM ##################
error_tream_l[z,i]<-sum((f2-MTM_filter(10,1,Y2))^2)/length(Y2)
h_tream_l[z,i]<-1
for (j in 1:250) {
if (sum((f2-MTM_filter(10,j,Y2))^2)/length(Y2)<error_tream_l[z,i]){
    error_tream_l[z,i]<-sum((f2-MTM_filter(10,j,Y2))^2)/length(Y2)
    h_tream_l[z,i]<-j
    }
  }
################################################
}
}
```