Оптимизационни задачи, Цифрови филтри

Практикум по приложение на графични програмни среди

Владимир Димитров

Катедра Силова Електроника
Факултет по Електронна Техника и Технологии

11 октомври 2021 г.

Съдържание

Приближения на данни с помощни криви

1. Приближения на данни с помощни криви
   - Регресия
   - Оптимизация
   - Интерполация на данни

2. Цифрови Филтри
   - Филтри с крайна импулсна характеристика
   - Цифрови филтри с безкрайна импусслна характеристика

Приближения на данни с помощни криви

Регресия

Регресия - отчитане на тенденция

Дефиниция на проблема

- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \ldots (t_n, y_n)$ да се дефинира "най-добрата"аналитична функция, без да е задължително да съвпада с конкретни точки.

Регресия - отчитане на тенденция

Дефиниция на проблема

- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \ldots (t_n, y_n)$ да се дефинира "най-добрата"аналитична функция, без да е задължително да съвпада с конкретни точки.

- Какви аналитични функции има?

Регресия - отчитане на тенденция

Дефиниция на проблема

- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \dots (t_n, y_n)$ да се дефинира "най-добрата"аналитична функция, без да е задължително да съвпада с конкретни точки.

- Какви аналитични функции има?
- По какъв критерии да изберем най-добрата апроксимация?

Аналитични функции

- Линейни по отношение на параметрите

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + ... + a_n x^n$

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + \ldots + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + \ldots + p_n x^n}{q_0 + q_1 x + \ldots + q_n x^n}$

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + \dots + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + \dots + p_n x^n}{q_0 + q_1 x + \dots + q_n x^n}$
  - Сплайн функции - Отсечково полиномна апроксимация. Два съседните полиноми трябва да имат еднакви производни (до някоя степен) в общата им точка.

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + ... + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + \ldots + p_n x^n}{q_0 + q_1 x + \ldots + q_n x^n}$
  - Сплайн функции - Отсечково полиномна апроксимация. Два съседните полиноми трябва да имат еднакви производни (до някоя степен) в общата им точка.
- Нелинейни по отношение на параметрите

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + ... + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + ... + p_n x^n}{q_0 + q_1 x + ... + q_n x^n}$
  - Сплайн функции - Отсечково полиномна апроксимация. Два съседните полиноми трябва да имат еднакви производни (до някоя степен) в общата им точка.
- Нелинейни по отношение на параметрите
  - Степенна $y = ax^b + c$
  - Фуриерова (тригонометрична) $y = a_1 \sin(b_1 x + c_1) + a_2 \sin(b_2 x + c_2) + ... + a_n \sin(b_n x + c_n)$

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + ... + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + \ldots + p_n x^n}{q_0 + q_1 x + \ldots + q_n x^n}$
  - Сплайн функции - Отсечково полиномна апроксимация. Два съседните полиноми трябва да имат еднакви производни (до някоя степен) в общата им точка.
- Нелинейни по отношение на параметрите
  - Степенна $y = ax^b + c$
  - Фуриерова (тригонометрична) $y = a_1 \sin(b_1 x + c_1) + a_2 \sin(b_2 x + c_2) + \ldots + a_n \sin(b_n x + c_n)$
  - Гаусова $y = a_1 e^{\left(-\left(\frac{x - b_1}{c_1}\right)^2\right)}$

Аналитични функции

- Линейни по отношение на параметрите
  - Линейна $y = a_0 + a_1 x$
  - Полином $y = a_0 + a_1 x + ... + a_n x^n$
  - Рационална $y = \frac{p_0 + p_1 x + \ldots + p_n x^n}{q_0 + q_1 x + \ldots + q_n x^n}$
  - Сплайн функции - Отсечково полиномна апроксимация. Два съседните полиноми трябва да имат еднакви производни (до някоя степен) в общата им точка.
- Нелинейни по отношение на параметрите
  - Степенна $y = ax^b + c$
  - Фуриерова (тригонометрична) $y = a_1 \sin(b_1 x + c_1) + a_2 \sin(b_2 x + c_2) + ... + a_n \sin(b_n x + c_n)$
  - Гаусова $y = a_1 e^{\left(-\left(\frac{x - b_1}{c_1}\right)^2\right)}$
  - Още много други

Минимизиране на грешката

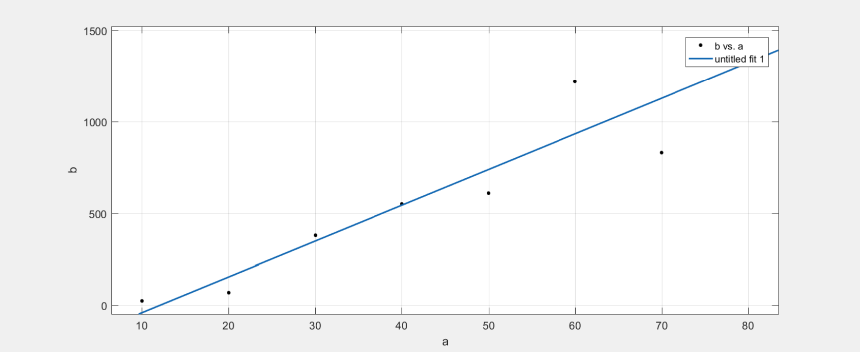
Дефиниция на проблема

Имаме n измервания, който са представени като (x_1, y_1),(x_2, y_2)...(x_n, y_n). Да се апроксимират със кривата $y = a_0 + a_1 x + e$, като се намерят коефициентите a_1, a_2, които минимизират грешката e.

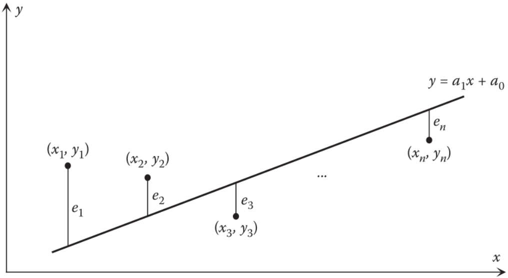
Минимизиране на грешката

Дефиниция на проблема

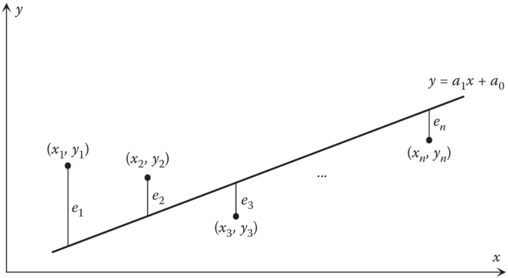
Имаме n измервания, който са представени като $(x_1, y_1),(x_2, y_2)\ldots(x_n, y_n)$. Да се апроксимират със кривата $y = a_0 + a_1x + e$, като се намерят коефициентите a_1, a_2, които минимизират грешката e.



Критерии за най-добри коефициенти?



- Критерий 1 - Минимизиране на сумата за всички стойности $\sum_i^n e_i = \sum_i^n (y_i - a_o - a_1 x_i)$.



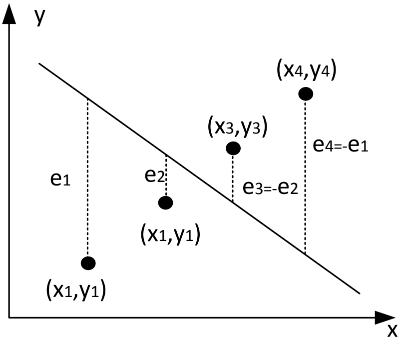
- Критерий 1 - Минимизиране на сумата за всички стойности $\sum_i^n e_i = \sum_i^n (y_i - a_o - a_1 x_i)$.

Критерий за най-добри коефициенти?

- Възможно ли е грешката да е 0 а кривата да не отговаря на тенденцията при данните?

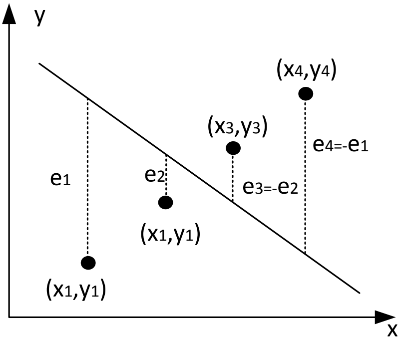
Критерий за най-добри коефициенти?

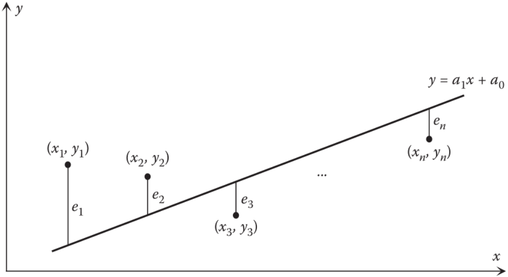
- Възможно ли е грешката да е 0 а кривата да не отговаря на тенденцията при данните?
- Извод: Критерий 1 е неподходящ.



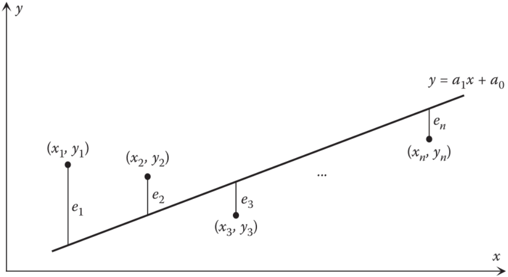
Критерий за най-добри коефициенти?

- Възможно ли е грешката да е 0 а кривата да не отговаря на тенденцията при данните?
- Извод: Критерий 1 е неподходящ.
- Необходима е функция, която да е стриктно положителна при отчитането на грешката (0 единствено, когато грешката е 0) - норма



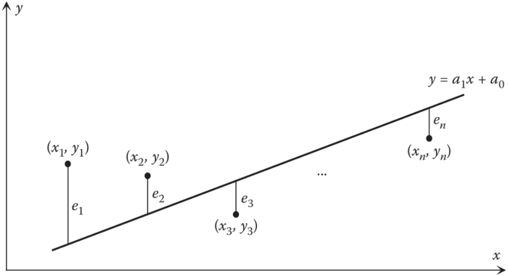


- Критерий 2 - Минимизиране на сумата по модул за всички стойности $\sum_i^n e_i = \sum_i^n (|y_i - a_o - a_1 x_i|)$ - Норма ℓ_1



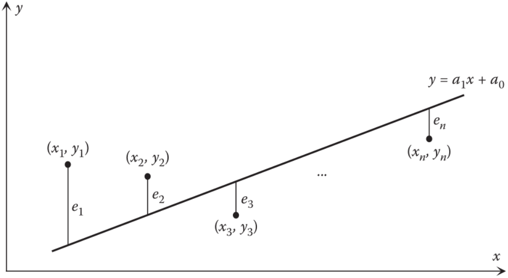
- Критерий 2 - Минимизиране на сумата по модул за всички стойности $\sum_i^n e_i = \sum_i^n (|y_i - a_o - a_1 x_i|)$ - Норма ℓ_1
- Основен проблем: може да не даде уникално решение на проблема - дайте пример

Критерии за най-добри коефициенти? - Норма ℓ_2



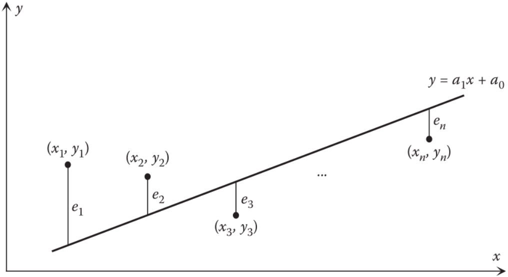
- Критерий 3 - Минимизиране на сумата от квадратите $SSE = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1x_i)^2$ - Норма ℓ_2

Критерии за най-добри коефициенти? - Норма ℓ_2



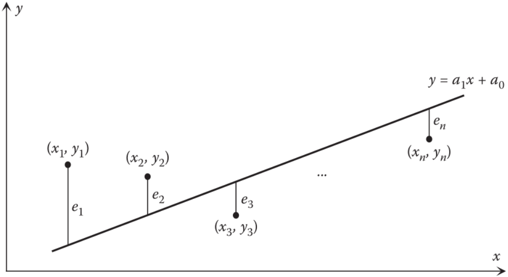
- Критерий 3 - Минимизиране на сумата от квадратите $SSE = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$ - Норма ℓ_2
- Най- често използван, но не е устойчив на грешки при измерване ("outliners") - дайте пример

Използвани норми



- Критерий 2 - Минимизиране на сумата по модул за всички стойности $\sum_i^n e_i = \sum_i^n (|y_i - a_o - a_1 x_i|)$ - Норма ℓ_1
- Критерий 3 - Минимизиране на сумата от квадратите $SSE = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$ - Норма ℓ_2

Използвани норми



- Критерий 2 - Минимизиране на сумата по модул за всички стойности $\sum_i^n e_i = \sum_i^n (|y_i - a_o - a_1 x_i|)$ - Норма ℓ_1
- Критерий 3 - Минимизиране на сумата от квадратите $SSE = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$ - Норма ℓ_2
- Критерий 4 - Минимизиране на максималната грешка за всяка стойност $\max \{|y_i - a_o - a_1 x_i|\}$ - Норма ℓ_∞

Сравнение между Критерии 2 и 3

- Основно се използват 2 и 3, като всяка от тях има предимства за определена ситуация.

Сравнение между Критерии 2 и 3

- Основно се използват 2 и 3, като всяка от тях има предимства за определена ситуация.
- Норма ℓ_2 $SSE = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$ - най- често използван при апроксимация, по-бързо изчисление и винаги уникално решение
- Норма ℓ_1 $\sum_i^n e_i = \sum_i^n (y_i - a_o - a_1 x_i)$ - предимства при ясно изразени грешки при измерването , но може да няма уникално решение

Линейна регресия -Как да намерим коефициентите?

Дефиниция на проблема

- Да се намерят коефициентите a_0 и a_1 на полинома $y = a_0 + a_1 x$, минимизиращи квадратната норма за n броя точки (x_1, y_1), $(x_2, y_2) \dots (x_n, y_n)$

- Квадратната норма има вида $S_r = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$

Линейна регресия -Как да намерим коефициентите?

Дефиниция на проблема

- Да се намерят коефициентите a_0 и a_1 на полинома $y = a_0 + a_1 x$, минимизиращи квадратната норма за n броя точки (x_1, y_1), $(x_2, y_2) \ldots (x_n, y_n)$

- Квадратната норма има вида $S_r = \sum_i^n e_i^2 = \sum_i^n (y_i - a_o - a_1 x_i)^2$
- Приравняваме производните на израза $\frac{dS_r}{da_0}, \frac{dS_r}{da_1}$ на израза по отношение на всеки от параметрите на 0

$$\frac{dS_r}{da_0} = -2 \sum (y_i - a_0 - a_1 x_i)$$

$$\frac{dS_r}{da_1} = -2 \sum ((y_i - a_0 - a_1 x_i) x_i)$$

Линейна регресия - Намиране на коефициентите

- Имайки предвид, че $(\sum a_0 = na_0)$

Линейна регресия - Намиране на коефициентите

- Имайки предвид, че $(\sum a_0 = na_0)$

$$0 = \sum y_i - na_0 - \sum a_1 x_i$$

$$0 = \sum x_i y_i - \sum a_0 x_i - \sum a_1 x_i^2$$

- Имайки предвид, че $(\sum a_0 = na_0)$

$$0 = \sum y_i - na_0 - \sum a_1 x_i$$

$$0 = \sum x_i y_i - \sum a_0 x_i - \sum a_1 x_i^2$$

- От двете уравнения се изразяват двата коефициента

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$a_0 = \frac{\sum y_i}{n} - a_1 \frac{\sum x_i}{n}$$

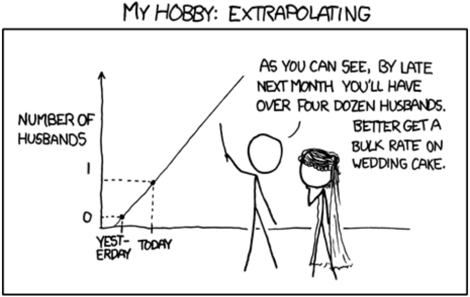
- $p = polyfit(x, y, n)$ - апроксимира (x,y) данните с полином от n-ти ред и връща коефициентите p на полинома

Функции регресионен анализ I - преди Matlab 2013

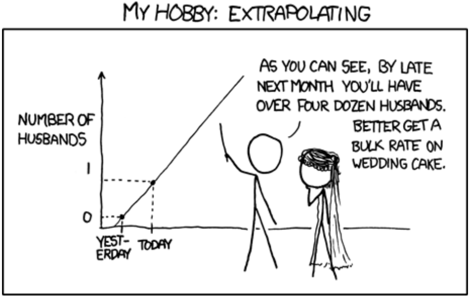
- $p = polyfit(x, y, n)$ - апроксимира (x,y) данните с полином от n-ти ред и връща коефициентите p на полинома
  - Ако имате коефициентите и ви интересува друга стойност $y = polyval(p, x)$ - изчислява стойността y на полином, с коефициенти p в точка x

Функции регресионен анализ I - преди Matlab 2013

- $p = polyfit(x, y, n)$ - апроксимира (x,y) данните с полином от n-ти ред и връща коефициентите p на полинома
  - Ако имате коефициентите и ви интересува друга стойност $y = polyval(p, x)$ - изчислява стойността y на полином, с коефициенти p в точка x



- $p = polyfit(x, y, n)$ - апроксимира (x,y) данните с полином от n-ти ред и връща коефициентите p на полинома
  - Ако имате коефициентите и ви интересува друга стойност $y = polyval(p, x)$ - изчислява стойността y на полином, с коефициенти p в точка x



- $fitobject = fit(x, y, fitType)$ - fitType определя апроксимиращата функция - полиномна, степенна, сплайн, Фурие или друга апроксимация. Използва се от curve fitting tool

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
  - Ползва се когато знаем кривата, трябват ни само коефициентите.

Statistics and Machine Learning Toolbox

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
  - Ползва се когато знаем кривата, трябват ни само коефициентите.
  - Опция Robust - ако има стойности с голяма разлика "Outliners"

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
    - Ползва се когато знаем кривата, трябват ни само коефициентите.
    - Опция Robust - ако има стойности с голяма разлика "Outliners"
    - R^2 - коефициент на определеност, колко добре модела описва съществуващите данни

Statistics and Machine Learning Toolbox

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
  - Ползва се когато знаем кривата, трябват ни само коефициентите.
  - Опция Robust - ако има стойности с голяма разлика "Outliners"
  - R^2 - коефициент на определеност, колко добре модела описва съществуващите данни
  - $p - value$ - колко съответния параметър е от значение при определяне на границата

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
  - Ползва се когато знаем кривата, трябват ни само коефициентите.
  - Опция Robust - ако има стойности с голяма разлика "Outliners"
  - R^2 - коефициент на определеност, колко добре модела описва съществуващите данни
  - $p - value$ - колко съответния параметър е от значение при определяне на границата
- Предсказване от съществуващ модел за други стойности $[ypred, yci] = predict(mdl, Xnew)$. Xnew трябва да е нов ред на таблицата

Statistics and Machine Learning Toolbox

- Линейна регресия $mdl = fitlm(tbl), modelspec$, където tbl е таблица с данните, а modelspec е уравнението на кривата
    - Ползва се когато знаем кривата, трябват ни само коефициентите.
    - Опция Robust - ако има стойности с голяма разлика "Outliners"
    - R^2 - коефициент на определеност, колко добре модела описва съществуващите данни
    - $p - value$ - колко съответния параметър е от значение при определяне на границата
- Предсказване от съществуващ модел за други стойности $[ypred, yci] = predict(mdl, Xnew)$. Xnew трябва да е нов ред на таблицата
- Аналогично действие има и feval(mdl,column1,column2), но приема директно вектор колона

- Линейна регресия $mdl = stepwiselm(X, y)$

- Линейна регресия $mdl = stepwiselm(X, y)$
  - Ползва се когато искаме да пробваме различни видове функции за апроксимация

Statistics and Machine Learning Toolbox

- Линейна регресия $mdl = stepwiselm(X, y)$
  - Ползва се когато искаме да пробваме различни видове функции за апроксимация
  - Няма опция Robust

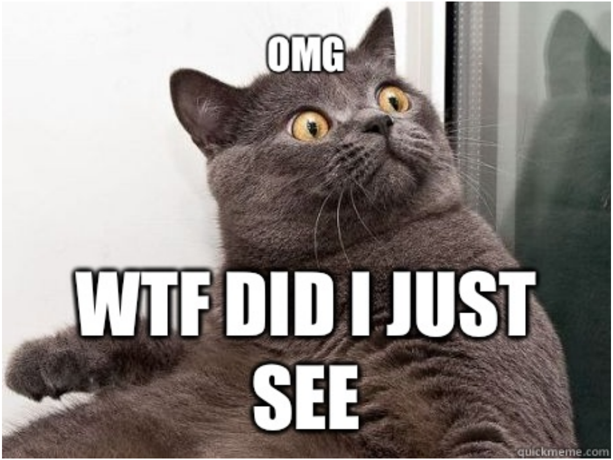
Statistics and Machine Learning Toolbox

- Линейна регресия $mdl = stepwiselm(X, y)$
  - Ползва се когато искаме да пробваме различни видове функции за апроксимация
  - Няма опция Robust
- Нелинейна регресия $mdl = fitnlm(X, y)$

Matlab Демонстрация

Да се демонстрират функциите fitlm и stepwiselm с готови данни от Matlab за линейна регресия.

Въпроси?



Приближения на данни с помощни криви

Оптимизация

Оптимизационна перспектива

- Регресионния проблем е да се намерят коефициентите $a_1, a_2 \ldots$, който да минимизират израза $y \approx Ax$.

Оптимизационна перспектива

- Регресионния проблем е да се намерят коефициентите $a_1, a_2 \dots,$ който да минимизират израза $y \approx Ax$.
- Иначе казано да се намери минималната стойност на нормата

$$\underset{x}{\text{minimize}} \quad ||y - Ax||$$

Оптимизационна перспектива

- Регресионния проблем е да се намерят коефициентите $a_1, a_2 \ldots$, който да минимизират израза $y \approx Ax$.
- Иначе казано да се намери минималната стойност на нормата

$$\underset{x}{\text{minimize}} \quad ||y - Ax||$$

- Това е линеен оптимизационен проблем без условия. В общия случай, ако се добавят условия той има вида

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \mathrm{fx} \\ \text{subject to} \quad & Ax \leq b \\ & A_{eq}x = b_{eq} \\ & lb \leq x \leq ub \end{aligned}$$

Проблем

Да се намери оптималното съотношение на производство на павета от фабрика като е известно:

Използван ресурс	Продукт		Ограничения
	Паве 1	Паве 2	
Брой Обществени поръчки [брой]	7	11	77
Производствено време [ч/тон]	10	8	80
Съхранение[тона]	6	3	
Печалба [мил. лева]	150	175	

Проблем

Да се намери оптималното съотношение на производство на павета от фабрика като е известно:

	Продукт		
Използван ресурс	Паве 1	Паве 2	Ограничения
Брой Обществени поръчки [брой]	7	11	77
Производствено време [ч/тон]	10	8	80
Съхранение[тона]	6	3	
Печалба [мил. лева]	150	175	

- В Matlab $x = \text{linprog}(f,A,b,Aeq,beq,lb,ub)$

Оптимизация на паветата

Проблем

Да се намери оптималното съотношение на производство на павета от фабрика като е известно:

Използван ресурс	Продукт		Ограничения
	Паве 1	Паве 2	
Брой Обществени поръчки [брой]	7	11	77
Производствено време [ч/тон]	10	8	80
Съхранение[тона]	6	3	
Печалба [мил. лева]	150	175	

- В Matlab x = linprog(f,A,b,Aeq,beq,lb,ub)
- Повече в упражнение 3 и литературата към него

Приближения на данни с помощни криви

Интерполация на данни

Интерполация - апроксимиране за междинни точни данни

Дефиниция на проблема

- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \ldots (t_n, y_n)$ да се дефинира аналитична функция, съвпадаща със съществуващите точки, която да даде "най-доброто"предположение за междинни или бъдещи стойности.

Интерполация - апроксимиране за междинни точни данни

Дефиниция на проблема

- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \ldots (t_n, y_n)$ да се дефинира аналитична функция, съвпадаща със съществуващите точки, която да даде "най-доброто"предположение за междинни или бъдещи стойности.

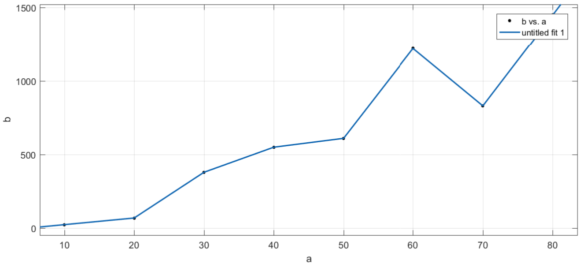
- Какви аналитични функции има?

Интерполация - апроксимиране за междинни точни данни

Дефиниция на проблема

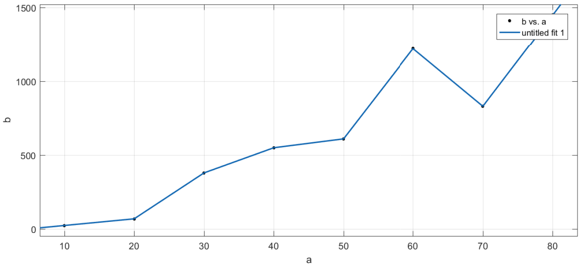
- На база съществуващи измервания (t_1, y_1), $(t_2, y_2) \ldots (t_n, y_n)$ да се дефинира аналитична функция, съвпадаща със съществуващите точки, която да даде "най-доброто"предположение за междинни или бъдещи стойности.

- Какви аналитични функции има?
- По какъв критерии да изберем най-добрата апроксимация?



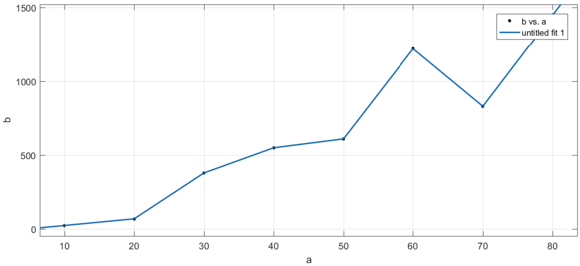
- Функция в Matlab vq1 = interp1(x,y,xq,method).

Интерполация на данни в Matlab



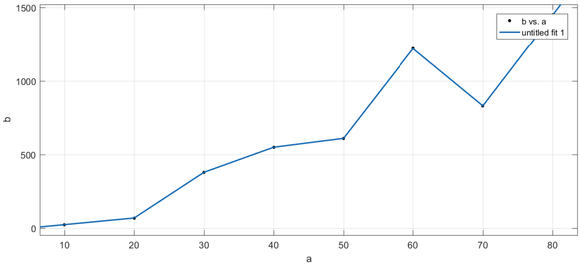
- Функция в Matlab vq1 = interp1(x,y,xq,method).
- Линейна 'linear'

Интерполация на данни в Matlab



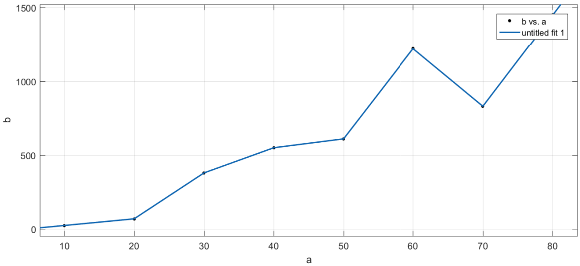
- Функция в Matlab vq1 = interp1(x,y,xq,method).
- Линейна 'linear'
- Най-близката съществуваща 'nearest', 'next', 'previous' - не генерира нови данни. Работи аналогично на ЦАП.

Интерполация на данни в Matlab



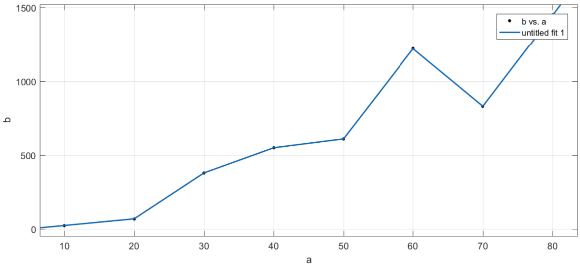
- Функция в Matlab vq1 = interp1(x,y,xq,method).
- Линейна 'linear'
- Най-близката съществуваща 'nearest', 'next', 'previous' - не генерира нови данни. Работи аналогично на ЦАП.
- Кубична сплайн функция 'cubic' - по-добре отразява осцилации

Интерполация на данни в Matlab



- Функция в Matlab vq1 = interp1(x,y,xq,method).
- Линейна 'linear'
- Най-близката съществуваща 'nearest', 'next', 'previous' - не генерира нови данни. Работи аналогично на ЦАП.
- Кубична сплайн функция 'cubic' - по-добре отразява осцилации
- Хермитиев полином 'makima' - по добра при монотонни функции (без осцилации)

Интерполация на данни в Matlab



- Функция в Matlab vq1 = interp1(x,y,xq,method).
- Линейна 'linear'
- Най-близката съществуваща 'nearest', 'next', 'previous' - не генерира нови данни. Работи аналогично на ЦАП.
- Кубична сплайн функция 'cubic' - по-добре отразява осцилации
- Хермитиев полином 'makima' - по добра при монотонни функции (без осцилации)
- за две дименсии Vq = interp2(X,Y,V,Xq,Yq)

Демонстрация с Matlab

Да се покажат различните видове интерполации върху общи данни

Цифрови Филтри

Видове

- С крайна импулсна характеристика - необходимо е да се помнят предишни стойност само на входа

Видове

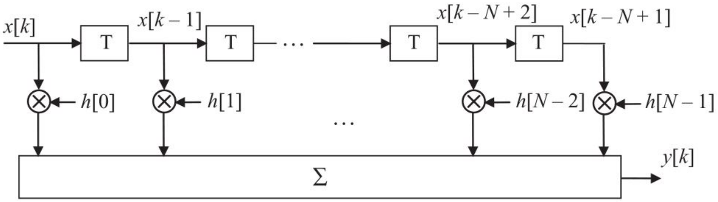
- С крайна импулсна характеристика - необходимо е да се помнят предишни стойност само на входа
- С безкрайна импулсна характеристика - необходимо е да се помнят предишни стойност и на входа и на изхода

Цифрови Филтри

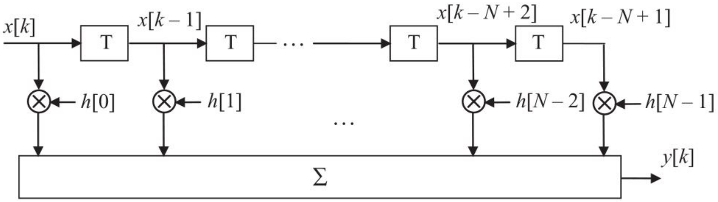
Филтри с крайна импулсна характеристика

Крайна импулсна характеристика (FIR)

- Използват се само входния сигнал в сегашния и предишни моменти.



- Използват се само входния сигнал в сегашния и предишни моменти.
- Необходими логически елементи за реализация: преместващ регистър, умножител и акумулатор



Пример за FIR филтър - коефициенти?

- Стандартен проблем: Да се усреднят измерените стойности от АЦП-то на микроконтролера

```
unsigned int ADCAVG(int16_t *adcBuffer)
{
unsigned int result1 = *adcBuffer;
unsigned int result2 = *(adcBuffer+2);
unsigned int result3 = 0;
result1+=*(adcBuffer+1);
result1=result1>>1;
result2+=*(adcBuffer+3);
result2=result2>>1;
result3= (result1+result2)>>1;
return result3;
}
```

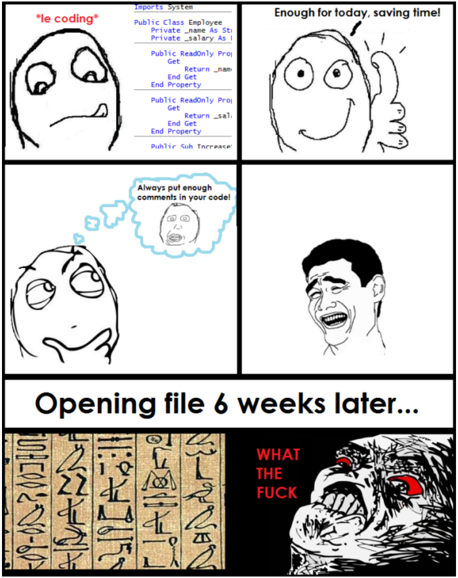
Пример за FIR филтър - коефициенти?

- Стандартен проблем: Да се усреднят измерените стойности от АЦП-то на микроконтролера

```
unsigned int ADCAVG(int16_t *adcBuffer)
{
unsigned int result1 = *adcBuffer;
unsigned int result2 = *(adcBuffer+2);
unsigned int result3 = 0;
result1+=*(adcBuffer+1);
result1=result1>>1;
result2+=*(adcBuffer+3);
result2=result2>>1;
result3= (result1+result2)>>1;
return result3;
}
```

- Какви са коефициентите на FIR филтъра?

И какво разбрахме от кода...



- h = fir1(n,Wn,ftype), връща коефициентите h на FIR филтър ftype от n-ти ред, със вектор на честотни ограничения Wn (срязова честота на НЧ и ВЧ)

Matlab

- h = fir1(n,Wn,ftype), връща коефициентите h на FIR филтър ftype от n-ти ред, със вектор на честотни ограничения Wn (срязова честота на НЧ и ВЧ)
- d = designfilt(resp,Name,Value)

Matlab

- h = fir1(n,Wn,ftype), връща коефициентите h на FIR филтър ftype от n-ти ред, със вектор на честотни ограничения Wn (срязова честота на НЧ и ВЧ)
- d = designfilt(resp,Name,Value)
  - Визуализация със fvtool(d)

- h = fir1(n,Wn,ftype), връща коефициентите h на FIR филтър ftype от n-ти ред, със вектор на честотни ограничения Wn (срязова честота на НЧ и ВЧ)
- d = designfilt(resp,Name,Value)
  - Визуализация със fvtool(d)
- Чрез вграденото приложение в Matlab filter designer tool

Диференчно уравнение, предавателна функция

Изходният сигнал за филтър с импулсна характеристика $h[k]$ за произволна входна поредица $x[k]$ е

$$y[k] = \sum_{m=0}^{N-1} h[m]\, x[k-m]$$

Диференчно уравнение, предавателна функция

Изходният сигнал за филтър с импулсна характеристика $h[k]$ за произволна входна поредица $x[k]$ е

$$y[k] = \sum_{m=0}^{N-1} h[m]\, x[k-m]$$

Z-преобразуван израз

$$(z) = \sum_{k=0}^{N-1} h_k z^{-k}$$

Честотната характеристика на филтърът се получава при $z = e^{j\omega+2\pi k}$, където $\omega \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$

Предимства и недостатъци

- Винаги са устойчиви

Предимства и недостатъци

- Винаги са устойчиви
- Не се променя фазата на сигнала

Предимства и недостатъци

- Винаги са устойчиви
- Не се променя фазата на сигнала
- Слабо чувствителни към точността на имплементация на коефициентите

Предимства и недостатъци

- Винаги са устойчиви
- Не се променя фазата на сигнала
- Слабо чувствителни към точността на имплементация на коефициентите
- По-голям брой коефициенти за дадено задание - повече памет и умножения

Цифрови Филтри

Цифрови филтри с безкрайна импулсна характеристика

Идея и видове

Да се дискретизира предавателната функция на проектиран аналогов филтър. Видове аналогови филтри, които се апроксимират:

- Butterworth - без осцилации в лентата на пропускане или филтриране, но малка стръмност

Идея и видове

Да се дискретизира предавателната функция на проектиран аналогов филтър. Видове аналогови филтри, които се апроксимират:

- Butterworth - без осцилации в лентата на пропускане или филтриране, но малка стръмност
- Bessel - Най-малка стръмност , но най-линейно ФЧХ.

Идея и видове

Да се дискретизира предавателната функция на проектиран аналогов филтър. Видове аналогови филтри, които се апроксимират:

- Butterworth - без осцилации в лентата на пропускане или филтриране, но малка стръмност
- Bessel - Най-малка стръмност , но най-линейно ФЧХ.
- Chebyshev - по-стръмено затихване за същия ред от Butterworth и Bessel, но с осцилации в лентата на пропускане(тип 1) или филтриране (тип 2). Не добро ФЧХ на сигнала

Идея и видове

Да се дискретизира предавателната функция на проектиран аналогов филтър. Видове аналогови филтри, които се апроксимират:

- Butterworth - без осцилации в лентата на пропускане или филтриране, но малка стръмност
- Bessel - Най-малка стръмност , но най-линейно ФЧХ.
- Chebyshev - по-стръмено затихване за същия ред от Butterworth и Bessel, но с осцилации в лентата на пропускане(тип 1) или филтриране (тип 2). Не добро ФЧХ на сигнала
- Elliptical - Най-стръмно затихване от всички, но с осцилации в лентата на пропускане и филтриране и нелинейно ФЧХ.

Диференчно уравнение, предавателна функция

Два метода на дискретизация:

- Чрез z-преобразуване на предавателната функция.

Диференчно уравнение, предавателна функция

Два метода на дискретизация:

- Чрез z-преобразуване на предавателната функция.
- Билинейна трансформация

Z-преобразуване - [bz,az] = impinvar(b,a,fs,tol) в Matlab

При z-преобразуването се запазва импулсната характеристика на филтъра, при условие, че честотата на дискретизация е безкрайност.

Z-преобразуване - [bz,az] = impinvar(b,a,fs,tol) в Matlab

При z-преобразуването се запазва импулсната характеристика на филтъра, при условие, че честотата на дискретизация е безкрайност.
Аналогова предавателна функция:

$$H_a(p) = \sum_{m=1}^{N} \frac{A_m}{p - a_m} \xrightarrow{L^{-1}} h_a(t) = \sum_{m=1}^{N} A_m e^{a_m t} u(t)$$

При z-преобразуването се запазва импулсната характеристика на филтъра, при условие, че честотата на дискретизация е безкрайност. Аналогова предавателна функция:

$$H_a(p) = \sum_{m=1}^{N} \frac{A_m}{p - a_m} \xrightarrow{L^{-1}} h_a(t) = \sum_{m=1}^{N} A_m e^{a_m t} u(t)$$

Дискретизирайки импулсната характеристика

$$h_d[kT_s] = \sum_{m=1}^{N} A_m e^{a_m k T_s} u(t)$$

При z-преобразуването се запазва импулсната характеристика на филтъра, при условие, че честотата на дискретизация е безкрайност. Аналогова предавателна функция:

$$H_a(p) = \sum_{m=1}^{N} \frac{A_m}{p - a_m} \xrightarrow{L^{-1}} h_a(t) = \sum_{m=1}^{N} A_m e^{a_m t} u(t)$$

Дискретизирайки импулсната характеристика

$$h_d[kT_s] = \sum_{m=1}^{N} A_m e^{a_m k T_s} u(t)$$

Чрез z-преобразуване се получава предавателната характеристика $z = e^{sT_s}$.

$$H_d(z) = \sum_{m=1}^{N} \frac{T_s A_m}{(1 - e^{s_m T_s} z^{-1})}$$

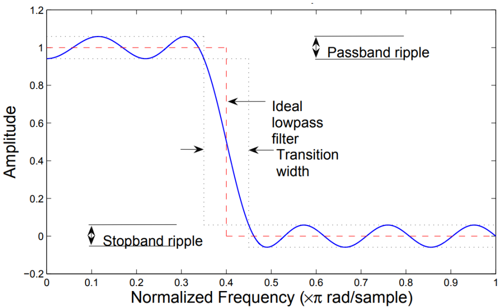
Билинейна трансформация - bilinear в Matlab

При билинейната трансформация се замества от аналоговата предавателна функкция

$$p = \frac{2}{T_s}\frac{z-1}{z+1}$$

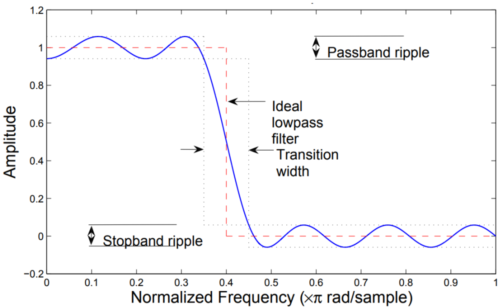
Функция в Matlab [Ad,Bd,Cd,Dd] = bilinear(A,B,C,D,fs,fp)

Проектиране на IIR филтри в Matlab



- Filter Designer приложението

Проектиране на IIR филтри в Matlab



- Filter Designer приложението
- fdesign.lowpass('Fp,Fst,Ap,Ast',Fp,Fst,0.05,0.05,'linear');

Предимства и недостатъци

- По-бързи (по-малко RAM, по-малко умножения)

Предимства и недостатъци

- По-бързи (по-малко RAM, по-малко умножения)
- По-ефективни за реализация на дадена АЧХ

Предимства и недостатъци

- По-бързи (по-малко RAM, по-малко умножения)
- По-ефективни за реализация на дадена АЧХ
- Чувствителни към точността на коефициентите - дори и неустойчиви

Предимства и недостатъци

- По-бързи (по-малко RAM, по-малко умножения)
- По-ефективни за реализация на дадена АЧХ
- Чувствителни към точността на коефициентите - дори и неустойчиви
- Нелилейна ФЧХ при определени условия

На когото му се чете вкъщи ...

- Stephen Boyd and Lieven Vandenberghe.
  Convex Optimization.
  Cambridge University Press, New York, NY, USA, 2004.
- R.P. Canale and D. Steven C. Chapra.
  Numerical Methods for Engineers.
  McGraw-Hill Education, 2014.
- A.V. Oppenheim and R.W. Schafer.
  Discrete-Time Signal Processing: Pearson New International Edition.
  Pearson Education Limited, 2013.