Uz 1.daļu Uz 3.daļu

Matemātika datoriķiem

Skaitliskās metodes (2.daļa)

Kārlis Podnieks, LU profesors

This work is licensed under a **Creative Commons License** and is copyrighted
© 2019-2023 by me, Karlis Podnieks. **17.10.23**

Literatūra

Harijs Kalis. Skaitliskās metodes ar datorprogrammu Maple, Mathematica lietošanu: mācību līdzeklis, Rīga: Latvijas Universitāte, 2001. Pieejama LUB 35 eksemplāros.

Funkciju interpolācija

Interpolācija, ekstrapolācija, aproksimācija, prognozēšana

Google attēli: extrapolation, what is the difference

Pats vispārīgākais uzdevums ir t.s. **regresija**, jeb funkcijas vērtību **prognozēšana.**

Vispārīgajā gadījumā runa ir par viena vai vairāku argumentu funkcijām $f(z_1,\dots,z_p)$ jeb $f(\boldsymbol{z})$, kuru argumenti var būt kā skaitliski, tā nomināli (*true/false* utml.)

Ir dotas funkcijas *f* vērtības $y_1,..,y_n$ datu punktos $\boldsymbol{x_1},\dots,\boldsymbol{x_n}$ (funkcijas argumentus te apvienojam vektoros),

$$f(\boldsymbol{x_1})=y_1,$$
$$\dots,$$
$$f(\boldsymbol{x_n})=y_n$$

(t.s. treniņa dati, $n\times(p+1)$ tabula) un citi datu punkti $\boldsymbol{x_{n+1}},\dots,\boldsymbol{x_{n+m}}$ (t.s. testa dati), kuros funkcija vērtības nav dotas (ir paslēptas no algoritma vai nav zināmas vispār).

Uzdevums ir piedāvāt funkcijas vērtību labas prognozes testa punktos, t.i.

$$f(\boldsymbol{x}_{n+1}) = ?,$$
$$\dots,$$
$$f(\boldsymbol{x}_{n+m}) = ?$$

Interpolācijas gadījumā testa punkti atrodas treniņa kopas apgabala iekšienē, **ekstrapolācijas** gadījumā tie ir ārpus šī apgabala.

Google attēli: *extrapolation, what is the difference*

Saistīts uzdevums

Google attēli: function approximation

Funkciju **aproksimācija** nozīmē vienas funkcijas aizstāšanu ar citu funkciju no kādas "regulāras" kopas (piemēram, aizstāšana ar polinomu). Parasti aizstāšana ir vajadzīga kāda segmenta $[a, b]$ *vai citāda galīga apgabala* ietvaros.

Sk. arī https://en.wikipedia.org/wiki/Interpolation;

https://en.wikipedia.org/wiki/Multivariate_interpolation;

https://en.wikipedia.org/wiki/Approximation_theory.

k-tuvāko kaimiņu metode

(k-nearest neighbour, kNN)

Ja treniņa kopā ir **ļoti daudz punktu**, tad var izmantot triviālu ideju:

Google attēli: *k-nearest neighbour regression*

Izvēlamies nelielu skaitli k ($k=1, 2, 3, \dots$), un prognozējam, ka $f(\boldsymbol{x}) \approx \frac{1}{k} \sum_i f(\boldsymbol{x}_i)$, kur summēšana notiek pa treniņa kopas tiem k punktiem $\boldsymbol{x}_i$, kas ir **vistuvākie** punktam $\boldsymbol{x}$.

Sk. bildi https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

Viena argumenta funkcijām šo metodi praktiski nelieto (lineārā interpolācija te ir tikpat vienkārša, bet daudz labāka).

Vairāku argumentu gadījumā kNN triviālo ideju realizēt nav tik vienkārši: vai nu katram testa punktam ir jāskata cauri visa treniņa kopa, vai arī jāizdomā kāda efektīvāka metode.

Tuvāko kaimiņu metode ir viena no t.s. **neparametriskās modelēšanas metodēm**. To galvenais trūkums: pat ja mums izdodas

iemācīties labi prognozēt iepriekš nezināmas vērtības, kas rodas kādā procesā, mēs *neiegūstam nekādu informāciju par šī procesa mehānismu (**funkcijas formulu**)*, kas nosaka šo vērtību rašanos.

Un, ja treniņa kopā ir maz punktu, tad šī metode neder, ir jāizmanto citas metodes.

Interpolācija ar polinomu: kļūdu novērtēšana

Ar funkciju interpolāciju mēs nodarbosimies šādā nostādnē: mēģināsim funkciju $f(x)$ aizstāt ar k-ās pakāpes polinomu:

$$P(x)=a_k x^k+a_{k-1}x^{k-1}+\ldots+a_1 x+a_0 ,$$

panākot, ka abu vērtības sakrīt dotajos k+1 kontrolpunktos $x_1<x_2<\ldots<x_{k+1}$:

$$P(x_i)=f(x_i) \text{ visiem } i: 1\le i\le k+1 .$$

Google attēli: *quadratic interpolation*

Interpolācijas/aproksimācijas kļūda ir starpība starp funkcijas *f* "īsto" vērtību $f(x)$ (ja tāda ir pieejama) un prognozēto vērtību $\hat{f}(x)$. Ja funkcijai eksistē pietiekami augstas kārtas atvasinājumi $f^{(m)}$, tad kļūdu novērtēšanai var izmantot Rolla teorēmu.

Rolla teorēma. Ja funkcija f(x) ir nepārtraukta segmentā [a, b], un diferencējama intervālā (a, b), un tai šajā intervālā ir vismaz **k+1** sakne, tad atvasinājumam f'(x) tajā ir vismaz **k** saknes.

Google atteli: *Rolle's theorem*

Sekas. Ja funkcija f(x) ir nepārtraukta segmentā *[a, b]*, un *k* reizes diferencējama intervālā (a, b), un tai šajā intervālā ir vismaz **k+1** sakne, tad atvasinājumam $f^{(k)}(x)$ te ir vismaz viena sakne.

No tā seko Teilora teorēmas analogs:

Teorēma. Ja funkcija f(x) un k-tās pakāpes polinoms P(x) sakrīt k+1 punktos $x_1<x_2<\ldots<x_{k+1}$, tad katram *x*, $x_1\le x\le x_{k+1}$ eksistē kāds skaitlis ζ , $x_1\le\zeta\le x_{k+1}$ tāds, ka

$$f(x)=P(x)+\frac{f^{(k+1)}(\zeta)}{(k+1)!}(x-x_1)(x-x_2)\dots(x-x_{k+1}) \ .$$

Redzam, ka šajā izteiksmē ieiet punkta *x* attālumi līdz visiem kontrolpunktiem.

Fīlinga līmenī: tā nav nejaušība, ka atlikuma loceklī ieiet funkcijas (k+1)-ais atvasinājums. Jo: **k-tās pakāpes polinoma (k+1)-ais atvasinājums ir nulle**, tāpēc "visas nepatikšanas" nāk no pašas funkcijas (k+1)-ā atvasinājuma lieluma. Piemēram,

$$(x^2)'=2x\,;(x^2)''=2\,;(x^2)'''=0 \ .$$

Un tāpēc $(f(x)-P(x))^{(k+1)}=f^{(k+1)}(x)$.

Piemērs: kvadrātiskās interpolācijas (parabolas) **kļūda**

Šajā gadījumā mums ir dotas funkcijas *f* vērtības trijos punktos $x_1<x_2<x_3$ un P(x) ir kvadrātisks polinoms (parabola). Kļūdas novērtēšanai rakstām:

$$f(x)=P(x)+\frac{f'''(\zeta)}{3!}(x-x_1)(x-x_2)(x-x_3) \ ,$$

kur ζ ir kāds skaitlis, $x_1\le\zeta\le x_3$.

Pierādījums. Izveidosim asprātīgu izteiksmi F(t) ar argumentu t, kam ir k+2 saknes $x, x_0, x_1, \dots, x_k$:

$$F(t)=f(t)-P(t)-\frac{f(x)-P(x)}{(x-x_0)(x-x_1)\dots(x-x_k)}(t-x_0)(t-x_1)\dots(t-x_k) \ .$$

Tiešām, arī *x* ir F sakne: $F(x)=0$. No iepriekšējās teorēmas seko, ka kādam ζ , $F^{(k+1)}(\zeta)=0$. Tā kā k-tās pakāpes polinomam $P^{(k+1)}(\zeta)=0$, bet labajā pusē esošā reizinājuma (k+1)-ais atvasinājums ir $(k+1)!$, tad

$$0=f^{(k+1)}(\zeta)-\frac{f(x)-P(x)}{(x-x_0)(x-x_1)\dots(x-x_k)}(k+1)! \ .$$ Q.E.D.

Sk. arī https://en.wikipedia.org/wiki/Polynomial_interpolation

Ja doto *x* vērtību atstarpes ir vienādas:

$x_0\,; x_1=x_0+h\,; x_2=x_0+2h\,; \dots\,; x_k=x_0+kh\,; h>0$, tad apzīmējot

$z=\frac{x-x_0}{h}$ jeb $x=x_0+zh$:

$f(x)=P(x)+\frac{f^{(k+1)}(\zeta)}{(k+1)!}zh(z-1)h\ldots(z-k)h$, un, pārveidojot, ir iegūta

Teorēma. Ja funkcija f(x) un k-tās pakāpes polinoms P(x) sakrīt *k+1* vienāda atstatuma punktos

$$x_0; x_1=x_0+h; x_2=x_0+2h; \ldots; x_k=x_0+kh; h>0 \ ,$$

tad katram *x*, $x_0\le x\le x_0+kh$ eksistē kāds skaitlis ζ , $x_0\le\zeta\le x_0+kh$ tāds, ka

$$f(x)=P(x)+f^{(k+1)}(\zeta)\frac{z(z-1)\ldots(z-k)}{(k+1)!}h^{k+1} \ ,$$

kur $z=\frac{x-x_0}{h}$ (ievērosim, ka *z* te nav vesels skaitlis).

Cits pieraksts: $f(x)=P(x)+f^{(k+1)}(\zeta)\boldsymbol{C_z^{k+1}}h^{k+1}$ vai

$f(x)=P(x)+f^{(k+1)}(\zeta)\binom{\boldsymbol{z}}{\boldsymbol{k+1}}h^{k+1}$.

Lineārā interpolācija

Šo vienkāršo metodi izmanto **viena argumenta funkciju** f(x) interpolācijai un ekstrapolācijai.

Google attēli: *linear interpolation*

Prognozējot f(x), atrodam treniņa kopā divus *x*-am tuvākos punktus x_1, x_2 un plaknē caur punktiem $(x_1, f(x_1)); (x_2, f(x_2))$ velkam taisni $y=a+bx$, t.i. atrodam *a*, *b* no vienādojumu sistēmas:

$$\begin{matrix} a+bx_1=f(x_1); \\ a+bx_2=f(x_2) \end{matrix} \text{ jeb}$$

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix}=\begin{pmatrix} f(x_1) \\ f(x_2) \end{pmatrix} \text{ , tātad:}$$

$$f(x)\approx a+bx=f(x_1)+(x-x_1)\frac{f(x_2)-f(x_1)}{x_2-x_1} \ .$$

Faktiski šādā veidā funkcija *f* tiek **aproksimēta** ar gabaliem-lineāru funkciju (*piecewise linear*), jeb t.s. **lineāro splainu** (par splainiem sk. tālāk).

Google attēli: linear spline

Lineārās interpolācijas kļūdas novērtēšana

Liekas skaidrs, ka lineārās interpolācijas prognozes būs jo precīzākas, jo mazāk izlocīts ("kurvīgs") būs funkcijas f grafiks. Grafika liekumu nosaka pieskares, t.i. pirmā atvasinājuma f'(x) maiņas ātrums, t.i. funkcijas otrais atvasinājums f''(x). Jo lielāka |f''(x)| vērtība, jo "līkāks" ir f(x) grafiks. Un ko sliktāka sanāk lineārā interpolācija. **Fīlings!**

Google attēli: *linear interpolation error*

Iepriekš minētās vispārīgās teorēmas speciāl-gadījums:

Teorēma. Ja funkcijai f segmentā $[x_1, x_2]$ eksistē **otrais** atvasinājums, tad katram x eksistē skaitlis ζ , $x_1 \le \zeta \le x_2$ tāds, ka:

$$f(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) + \frac{f''(\zeta)}{2!}(x - x_1)(x - x_2) \quad .$$

Tests. Pārliecinieties, ka lineārās interpolācijas kļūda nebūs lielāka par

$$\frac{1}{8}(x_2 - x_1)^2 \max_{x_1 \le \zeta \le x_2} |f''(\zeta)| \quad .$$

Norāde: atrodiet izteiksmes $(x - x_1)(x - x_2)$ maksimumu.

Sk. arī https://en.wikipedia.org/wiki/Linear_interpolation.

Aproksimācija ar polinomu

Sk. arī https://en.wikipedia.org/wiki/Stone%E2%80%93Weierstrass_theorem

Sk. Google attēli: *polynomial approximation*

Bildēs redzam, ka polinomi principā ir pietiekami "lokani", lai ar patvaļīgu iepriekš dotu precizitāti aproksimētu jebkuru nepārtrauktu funkciju. Precīzāk:

Veierštrāsa aproksimācijas teorēma. Ja funkcija *f* ir nepārtraukta segmentā *[a, b]*, tad jebkuram skaitlim $\epsilon > 0$ var uzbūvēt tādu polinomu *P*, ka visiem *x* šajā segmentā: $|f(x) - P(x)| \le \epsilon$.

Piemērs:

Google attēli: *sine polynomial approximation* (sin(x) aproks. ar 7.pak.polinomu)

Šo teorēmu (tai eksistē konstruktīvs pierādījums) var izmantot gadījumos, kad ir jānoprogrammē kādas analītiski sarežģītās funkcijas vērtību rēķināšana. Tā vietā varam rēķināt laba aproksimējošā polinoma vērtības, izmantojot reizināšanas un

saskaitīšanas operācijas, un neko vairāk. Kādreiz tas bija svarīgi...

Interpolācija ar polinomu

Sk. arī https://en.wikipedia.org/wiki/Polynomial_interpolation.

Interpolācija nav aproksimācija: te mums ir dotas nezināmas funkcijas vērtības tikai atsevišķos punktos. Tāpēc būvēsim polinomus, kas šajos punktos ar dotajām funkcijas vērtībām precīzi sakrīt.

Svarīgākais fenomens:

Teorēma. Eksistē viens un tikai viens polinoms *P(x)* , kura pakāpe nav lielāka par *k*, un kurš dotajām x vērtībām $x_1, \dots, x_{k+1}$ (to skaits ir *k+1*) pieņem dotās vērtības $y_1, \dots, y_{k+1}$.

[Kāpēc "nav lielāka" nevis "vienāda" ar *k*?]

Piemēri:

$k+1=2\,;\, y=ax+b$ – caur diviem punktiem var novilkt tieši vienu taisni;

$k+1=3\,;\, y=ax^2+bx+c$ – caur trim punktiem (var novilkt tieši vienu taisni vai vienu parabolu.

Savienojot šo teorēmu ar (*k+1*)-tuvāko kaimiņu metodi, mēs iegūstam **polinomiālo interpolāciju** viena argumenta funkcijām: lai prognozētu f(x), sameklējam treniņa kopā k+1 tuvākos punktus $x_1, \dots, x_{k+1}$ un atbilstošās *f* vērtības $y_1, \dots, y_{k+1}$, un uzbūvējam vienīgo *0*-tās līdz *k*-tās pakāpes polinomu

$$P(x)=a_k x^k+a_{k-1}x^{k-1}+\dots+a_1 x+a_0 \ ,$$

kurš pieņem šīs vērtības, un tad prognozējam $\hat{f}(x)\approx P(x)$.

Piemēri: Google attēli: *quadratic interpolation*

Brīdinājums: Vēlāk redzēsim, ka augstāku (par 3) pakāpju polinomi interpolācijas uzdevumam nereti nav piemēroti (sk. tālāk par t.s. Runges fenomenu). Vēl mazāk piemēroti tie ir *ekstrapolācijai*, jo ārpus uzdoto vērtību apgabala, polinomi strauji tiecas uz bezgalību! Piemēru sk. kursā "Datizraces algoritmi", sk. sadaļu par pārpielāgotību (*overfitting*).

Tātad **polinoms te eksistē tikai viens**, bet ir piedāvātas vairākas elegantas metodes tā konstruēšanai, piemēram: *Vandermonda*

matrica, Lagranža interpolācijas polinoms, Ņūtona interpolācijas polinoms.

Ir arī citas – vēl sarežģītākas metodes, katrai ir savs pamatojums, kad to lietot: *Gauss interpolation formula, Stirling interpolation formula, Bessel interpolation formula* (meklējiet ar Google).

Pirmā metode: Vandermonda matrica

No prasītajām *k+1* vienādībām $P(x_i)=y_i$ (i=1..k+1), jeb:

$$P(x_i)=a_k x_i^k+a_{k-1}x_i^{k-1}+...+a_1 x_i+a_0=y_i$$

iegūstam (k+1)x(k+1) lineāru vienādojumu sistēmu, kuras atrisināšana dos polinoma koeficientu a_i vērtības:

$$\begin{pmatrix} x_1^k & x_1^{k-1} & ... & x_1 & 1 \\ x_2^k & x_2^{k-1} & ... & x_2 & 1 \\ ... & ... & ... & ... & ... \\ x_k^k & x_k^{k-1} & ... & x_k & 1 \\ x_{k+1}^k & x_{k+1}^{k-1} & ... & x_{k+1} & 1 \end{pmatrix} \begin{pmatrix} a_k \\ a_{k-1} \\ ... \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_k \\ y_{k+1} \end{pmatrix} .$$

Šis sistēmas matricu sauc par Vandermonda matricu.

Testa uzdevums. Parādiet, ka šīs matricas determinants ir $\prod_{1\le i<j\le k}(x_i-x_j)$.

Tā kā visi $x_1,...,x_{k+1}$ ir dažādi, tad sistēmas determinants nav 0, un tai eksistē tieši viens atrisinājums (tas būtu arī viens no iepriekšējās teorēmas pierādījumiem).

Ar šo metodi iegūtā polinoma izteiksme

$$P(x)=a_k x^k+a_{k-1}x^{k-1}+...+a_1 x+a_0$$

polinoma vērtību aprēķinam izmanto k saskaitīšanas operācijas un $2k-1$ reizināšanas. Bet, pārveidojot šo izteiksmi par

$$P(x)=a_0+x(a_1+x(a_2+x(...+x\,a_k)))\ ,$$

polinoma vērtību aprēķinam pietiek ar k saskaitīšanām un k reizināšanām.

Testa uzdevums. Izmantojot Vandermonda matricu, uzbūvējiet polinomu P, kas pieņem šādas trīs vērtības: *P(-1)=-2; P(1)=2; P(2)=-1.*

Lagranža interpolācijas polinoms

Sk. arī https://en.wikipedia.org/wiki/Lagrange_polynomial

Lagranža ieteiktā metode kā būvēt (tā paša vienīgā) polinoma izteiksmi izskatās ļoti eleganti [saprast formulas ideju]:

$$P(x)=\sum_{i=1}^{k+1} y_i \prod_{j\neq i} \frac{x-x_j}{x_i-x_j} \quad .$$

Lineārā izteiksme 2 vērtību gadījumam:

$$P(x)=y_1\frac{x-x_2}{x_1-x_2}+y_2\frac{x-x_1}{x_2-x_1} \quad .$$

Kvadrātiskais polinoms 3 vērtību gadījumam:

$$P(x)=y_1\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}+y_2\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}+y_3\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} \quad .$$

Testa uzdevums. Izmantojot Lagranža interpolāciju, uzbūvējiet polinomu P, kas pieņem šādas trīs vērtības: *P(-1)=-2; P(1)=2; P(2)=-1.*

Dalītās diferences

Elegants izgudrojums, ko izmanto Ņūtona interpolācija polinomu būvēšanai.

Sk. arī https://en.wikipedia.org/wiki/Divided_differences.

Pieņemsim, ka ir dots *k+1* datu punkts: $(x_0,y_0);\ldots,(x_k,y_k)$.

Sekojošie (kaut arī vispār-pieņemtie) apzīmējumi ir nedaudz maldinoši, jo tajos figurē tikai y-i, bet x-i paliek “fonā”.

Null-tās kārtas dalītās diferences:

$$[y_i]=y_i \quad .$$

Pirmās kārtas dalītās diferences (it kā tuvināts pirmais atvasinājums – y-a pieaugums dalīts ar x pieaugumu):

$$[y_0,y_1]=\frac{[y_1]-[y_0]}{x_1-x_0}=\frac{y_1-y_0}{x_1-x_0} \quad ;$$

$$[y_i,y_{i+1}]=\frac{[y_{i+1}]-[y_i]}{x_{i+1}-x_i}=\frac{y_{i+1}-y_i}{x_{i+1}-x_i} \quad .$$

Otrās kārtas dalītās diferences (fantāzija – it kā tuvināts otrais atvasinājums):

$$[y_0, y_1, y_2] = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0} = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} \; ;$$

$$[y_i, y_{i+1}, y_{i+2}] = \frac{[y_{i+1}, y_{i+2}] - [y_i, y_{i+1}]}{x_{i+2} - x_i} \; .$$

Trešās kārtas:

$$[y_0, y_1, y_2, y_3] = \frac{[y_1, y_2, y_3] - [y_0, y_1, y_2]}{x_3 - x_0} \; .$$

Utt., vispārīgā rekurentā formula **j-tās kārtas diferencei** (skaitītājā ir divu secīgu (j–1)-ās kārtas diferenču starpība):

$$[y_i, y_{i+1}, \dots, y_{i+j}] = \frac{[y_{i+1}, \dots, y_{i+j}] - [y_i, \dots, y_{i+j-1}]}{x_{i+j} - x_i} \; .$$

Tālāk maldināšana beidzas:

Ja y-i ir funkcijas f vērtības: $(x_0, f(x_0)); \dots, (x_k, f(x_k))$, tad ir pieņemts $[y_i, y_{i+1}, \dots, y_{i+j}]$ rakstīt kā $f[x_i, x_{i+1}, \dots, x_{i+j}]$, piemēram,

$$f[x_0] = f(x_0);$$

$$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}; \; .$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0};$$

$$f[x_i, x_{i+1}, \dots, x_{i+j}] = \frac{f[x_{i+1}, \dots, x_{i+j}] - f[x_i, \dots, x_{i+j-1}]}{x_{i+j} - x_i} \; .$$

Konkrētus dalīto diferenču aprēķinu piemērus sk. Google attēli: *divided differences* (apskatīsim aprēķinu kubiskajai funkcijai 2.2.2).

Testa uzdevums. Doti 4 datu punkti: (1, 1); (2, 1.41); (3, 1.73); (4, 2). Runa ir par funkciju $f(x) = \sqrt{x}$. Aprēķiniet visas dalītās diferences.

Ņūtona interpolācijas polinoms

Sk. arī https://en.wikipedia.org/wiki/Newton_polynomial.

Vienīgo k-tās pakāpes polinomu P(x), kura grafiks iet caur *k+1* datu punktiem $(x_0, y_0); \ldots, (x_k, y_k)$, var uzbūvēt vēl vienā veidā, kura ideja nāk no Ņūtona.

Ņūtons gribēja iegūt P(x) šāda veida formulu ar skaitliskiem koeficientiem:

$$P(x) = Q_0 + Q_1(x - x_0) + Q_2(x - x_0)(x - x_1) + \ldots + Q_k(x - x_0)\ldots(x - x_{k-1}) \quad .$$

Izrādās, ka šajā formā koeficienti Q_i ir aprēķināmi kā *dalītās diferences*!

$P(x_0) = Q_0 = y_0$, tātad $Q_0 = y_0$.

$P(x_1) = y_0 + Q_1(x_1 - x_0) = y_1$, tātad $Q_1 = \frac{y_1 - y_0}{x_1 - x_0} = [y_0, y_1]$.

$P(x_2) = y_0 + [y_0, y_1](x_2 - x_0) + Q_2(x_2 - x_0)(x_2 - x_1) = y_2$,

tātad $Q_2 = \frac{y_2 - y_0 - [y_0, y_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$;

bet tālāk jārīkojas asprātīgi:

$$Q_2 = \frac{y_2 - \mathbf{y_1} + \mathbf{y_1} - y_0 - [y_0, y_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \quad ;$$

$$Q_2 = \frac{y_2 - y_1 + [y_0, y_1](x_1 - x_0) - [y_0, y_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \quad ;$$

$$Q_2 = \frac{[y_1, y_2] - [y_0, y_1]}{x_2 - x_0} = [y_0, y_1, y_2] \quad , \text{utt.}$$

Tātad k-tās pakāpes polinomu, kura grafiks iet caur *k+1* datu punktiem $(x_0, y_0); \ldots, (x_k, y_k)$ ar dalītajām diferencēm var uzrakstīt šādi:

$$P(x) = [y_0] + [y_0, y_1](x - x_0) + \ldots + [y_0, \ldots, y_k](x - x_0)\ldots(x - x_{k-1}) \quad .$$

Un ja interpolējam funkciju f ar *k*-tās pakāpes polinomu, tad:

$$f(x) \approx f[x_0] + f[x_0, x_1](x - x_0) + \ldots + f[x_0, \ldots, x_k](x - x_0)\ldots(x - x_{k-1})$$

To sauc par **Ņūtona interpolācijas polinomu.**

Ņūtona metode ir **inkrementāla**: ja datu punktiem $(x_0, y_0), \ldots, (x_k, y_k)$

pievieno klāt vēl vienu punktu (x_{k+1}, y_{k+1}) , tad citas polinomu formula ir jāpārrēķina no sākuma, bet Ņūtona formulai tikai jāpievieno jauns saskaitāmais:

$$f(x) \approx f[x_0] + \ldots + f[x_0, \ldots, x_{k+1}](x - x_0)\ldots(x - x_k) \ .$$

Divu punktu gadījumā: $f(x) \approx [y_0] + f[x_0, x_1](x - x_0)$, t.i. sanāk jau zināmā lineārās interpolācijas formula:

$$f(x) \approx f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \ .$$

Trīs punktu gadījumā (kvadrātiskā interpolācija):

$$f(x) \approx f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_{1,} x_2](x - x_0)(x - x_1) \ , \text{ jeb}$$

$$f(x) \approx f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + f[x_0, x_{1,} x_2](x - x_0)(x - x_1) \ ,$$

kur $f[x_0, x_1, x_2] = \dfrac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$.

Testa uzdevums. Izmantojot Ņūtona metodi, uzbūvējiet polinomu P, kas pieņem šādas trīs vērtības: *P(-1)=-2; P(1)=2; P(2)=-1*.

Ja punktu x_j atstarpes ir vienādas:

$$x_0; x_1 = x_0 + h; x_2 = x_0 + 2h; \ldots; x_k = x_0 + kh; h > 0 \ ,$$

tad, apzīmējot $z = \dfrac{x - x_0}{h}$, iegūstam:

$$P(x) = [y_0] + [y_0, y_1]zh + \ldots + [y_0, \ldots, y_k]zh(z-1)h\ldots(z-k+1)h \ .$$

$$z(z-1)\ldots(z-j+1) = \frac{z(z-1)\ldots(z-j+1)}{j!}j! = C_z^j j! \ , \text{ kur}$$

$$\frac{z(z-1)\ldots(z-j+1)}{j!} = C_z^j = \binom{z}{j}$$ (te z nav domāts kā vesels skaitlis).

Esam ieguvuši *Ņūtona dalīto diferenču formulu:*

$$P(x)=\sum_{j=0}^{k}[y_0,\ldots,y_j]\binom{z}{j}h^j\,j!$$

...