Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Začneme něčím, co už dobře známe...

Matematická formulace problému

Remark 1

Buď $f : \mathbb{R} \to \mathbb{R}$ funkce jejíž hodnoty známe ve vzájemně různých bodech $x_0 < x_1 < \ldots < x_n$. Hledáme polynom $L_n(x)$ co nejnižšího stupně tak, aby platilo

$$L_n(x_i) = f(x_i) = y_i \quad \text{pro} \quad i = 0, \ldots, n.$$

Za **vhodných podmínek** *by mohlo platit, že $L_n(x)$ bude blízko $f(x)$ i v ostatních bodech. Odhadujeme-li hodnotu f mezi body $x_0, \ldots, x_n$, jde o* **interpolaci** *jinak jde o* **extrapolaci**.

Je-li

$$L_n(\vec{\theta}, x) = \sum_{i=0}^{n} \theta_i x^i,$$

pak lze podmínku $L_n(\vec{\theta}, x_i) = f(x_i) = y_i$ pro $i = 0, \ldots, n$ přepsat jako

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

resp.

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n} \vec{\theta} = \vec{y}.$$

Koeficienty $\theta_0, \ldots, \theta_n$ lze tedy získat vyřešením soustavy lineárních rovnic

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n}\vec{\theta} = \vec{y}.$$

Definition 2

Nechť $x_0, \ldots, x_n \in \mathbb{R}$. Matice $\mathbb{V}^{(m)}_{x_0,\ldots,x_n} \in \mathbb{R}^{n+1,m+1}$ definovaná jako

$$\mathbb{V}^{(m)}_{x_0,\ldots,x_n} = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^m \\ 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{pmatrix}$$

se nazývá Vandermondova matice.

Theorem 3

Nechť $x_0, \ldots, x_n \in \mathbb{R}$ jsou navzájem různé. Pak příslušná Vandermondova matice $\mathbb{V}^{(n)}_{x_0,\ldots,x_n} \in \mathbb{R}^{n+1,n+1}$

$$\mathbb{V}^{(n)}_{x_0,\ldots,x_n} = \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix}$$

je regulární.

Proof.

Video na Youtube

Theorem 4

Buď $f : \mathbb{R} \to \mathbb{R}$ a body $x_0, \ldots, x_n \in D_f$. Pak existuje právě jeden polynom P stupně n splňující

$$P(x_i) = f(x_i) \quad \text{pro} \quad \forall i = 0, \ldots, n.$$

Proof.

Důkaz plyne z regularity matice $\mathbb{V}^{(n)}_{x_0,\ldots,x_n}$.

Alternativní důkaz: Video na Youtube

Lagrangeův polynom

Co kdybychom zkoušeli interpolaci **polynomem nižšího stupně**?

- jednodušší polynom by byl praktičtější
- museli bychom řešit soustavu rovnic tvaru

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^m \\ 1 & x_1 & \dots & x_1^m \\ 1 & x_2 & \dots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^m \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

  pro $n > m$, která obecně nemá řešení.
- tj. polynom nižšího stupně obecně nemůže nabývat všech předepsaných hodnot

Co kdybychom zkoušeli interpolaci **polynomem vyššího stupně**?

- polynom vyššího stupně by mohl možná dát lepší interpolaci
- museli bychom řešit soustavu rovnic tvaru

$$\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^m \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{pmatrix} = \begin{pmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{pmatrix}$$

pro $n < m$, která má nekonečně mnoho řešení

- tj. polynom vyššího stupně není určen jednoznačně

Lagrangeův polynom

- v předchozím jsme tedy měli daný model = polynom stupně n
- víme, že i správná volba n není úplně jednoduchá
  - závisí na diferencovatelnosti funkce f
  - omezuje ji také Rungův jev
- v reálných úlohách jsou navíc data $y_1, \ldots, y_n$ zatížena šumem
- místo interpolace se budeme zabývat **regresí**

Polynomiální regrese

Trochu změníme značení:

- $0, \ldots, n \to 1, \ldots, N$
- $\vec{\theta} \to \vec{w} = (w_0, \ldots, w_d)$
- $L_n(\vec{\theta}, x) \to P(\vec{w}, x)$
- máme body $x_1, \ldots, x_N$ a k nim přiřazené hodnoty $y_1, \ldots, y_N$
- hledáme polynom

$$P(\vec{w}, x) = w_d x^d + \ldots w_1 x + w_0,$$

takový, že

$$P(\vec{w}, x_i) = y_i,$$

pro $i = 1, \ldots, N$.

Polynomiální regrese

- předpokládejme, že data y_i jsou poškozena Gaussovským šumem, tj.

$$y_i = f(x_i) + e_i \text{ pro } e_i \sim N(0, \sigma)$$

- tj.

$$e_i = y_i - f(x_i) \sim N(0, \sigma) \Rightarrow y_i \sim N(f(x_i), \sigma)$$

- potom je pravděpodobnostní rozdělení pro y_i dáno jako

$$p(y_i \mid x_i, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{f(x_i)-y_i}{\sigma}\right)^2}$$

- pravděpodobnostní rozdělení pro všechny body $y_1, \ldots, y_N$ pak je

$$p(\vec{y} \mid \vec{x}, \sigma) = \prod_{i=1}^{N} N(f(x_i), \sigma)$$

- odhad funkce $f(x)$ pomocí polynomu $P(\vec{w}, x)$ lze nyní získat pomocí *maximálně věrohodného odhadu* parametrů $\vec{w}$, tj.

$$\vec{w}^* = \arg\max_{\vec{w}} p(\vec{y} \mid \vec{x}, \vec{w}, \sigma) = \arg\max_{\vec{w}} \prod_{i=1}^{n} N(P(\vec{w}, x_i), \sigma) = \arg\max_{\vec{w}} \prod_{i=1}^{n} \mathcal{N}(x_i, \sigma, \vec{w}),$$

kde

$$\mathcal{N}(x_i, \sigma, \vec{w}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{P(\vec{w},x)-y_i}{\sigma}\right)^2}$$

- logaritmováním dostaneme

$$\vec{w}^* = \arg\max_{\vec{\theta}} \sum_{i=1}^{n} \ln \mathcal{N}(x_i, \sigma, \vec{w})$$

- a následně pak

$$
\begin{aligned}
\vec{w}^* &= \arg\max_{\vec{w}} \sum_{i=1}^{n} \ln \mathcal{N}(x_i, \sigma, \vec{w}) \\
&= \arg\max_{\vec{w}} \sum_{i=1}^{n} \ln \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{P(\vec{w},x_i)-y_i}{\sigma}\right)^2} \right) \\
&= \arg\max_{\vec{w}} \left(-n \ln(\sigma\sqrt{2\pi}) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(P(\vec{w}, x_i) - y_i\right)^2 \right) \\
&= \arg\min_{\vec{w}} \sum_{i=1}^{n} \left(P(\vec{w}, x_i) - y_i\right)^2 \\
&= \arg\min_{\vec{w}} \left\| P(\vec{w}, x_i) - y_i \right\|_2^2
\end{aligned}
$$

Polynomiální regrese - ztrátová funkce

- došli jsme tak k minimalizaci L_2-normy, tj. ztrátová funkce L (*loss function*) je dána jako

$$L(\vec{w}, \sigma, \vec{x}, \vec{y}) = \left\| \vec{P}(\vec{w}, \vec{x}) - \vec{y} \right\|_2^2,$$

kde jsme použili značení

$$\vec{P}(\vec{w}, \vec{x}) = \begin{pmatrix} P(\vec{w}, x_i) \\ \vdots \\ P(\vec{w}, x_n) \end{pmatrix}$$

Polynomiální regrese - ztrátová funkce

- podobně bychom s pomocí Laplaceova rozdělení

$$f(x \mid \mu, b) = \frac{1}{2b} e^{\left(-\frac{|x-\mu|}{b}\right)}$$

došli k minimalizaci L_1-normy

Polynomiální regrese - metoda největšího spádu

- napočítáme gradient $\nabla_{\vec{w}}$
- máme

$$P(\vec{w}, x) = w_n x^n + \ldots + w_1 x + w_0$$

- dále

$$\begin{aligned} \frac{\partial}{\partial w_j} \left\| P(\vec{w}, \vec{x}) - \vec{y} \right\|_2^2 &= \sum_{i=1}^{N} 2 \left(P(\vec{w}, x_i) - y_i \right) \frac{\partial}{\partial w_j} P(\vec{w}, x_i) \\ &= 2 \sum_{i=1}^{N} \left(P(\vec{w}, x_i) - y_i \right) x_i^j \end{aligned}$$

- všimneme si, že díky výrazu x_i^j mohou mít vyšší složky gradientu tendenci růst do velkých hodnot v absolutní hodnotě

Polynomiální regrese - metoda největšího spádu

- k řešení můžeme použít metodu největšího spádu - *gradient descent*
- tato metoda je dána předpisem

$$\vec{w}^{(k+1)} = \vec{w}^{(k)} - \gamma \nabla_{\vec{w}} L(\vec{w}, \sigma, \vec{x}, \vec{y})$$

- parametr γ je relaxační parametr (*learning rate*)
  - menší hodnoty vedou k pomalejší ale stabilnější konvergenci
  - větší hodnoty metodu urychlují, ale mohou vést k divergenci metody
- předpis iterací tedy vypadá takto

$$\vec{w}_j^{(k+1)} = \vec{w}_j^{(k)} - 2\gamma \left(\vec{P}(\vec{w}, \vec{x}) - \vec{y} \right) \cdot \vec{x}^j,$$

kde $\vec{x}^j$ znamená mocnění vektoru po složkách

Polynomiální regrese

- místo přímé metody pro konstrukci Lagrangeova polynomu nyní máme iterační metodu
- výhodou ale je, že stupeň polynomu P nyní můžeme volit libovolně nezávisle na počtu bodů (x_i, y_i)
- díky šumu teď obzvlášť neplatí, že polynom vyššího stupně dává lepší výsledek
- lepší fitování dat nemusí lépe aproximovat původní křivku
- to se pozná tak, že provedeme test na datech, která jsme nepoužili při optimalizaci

Polynomiální regrese

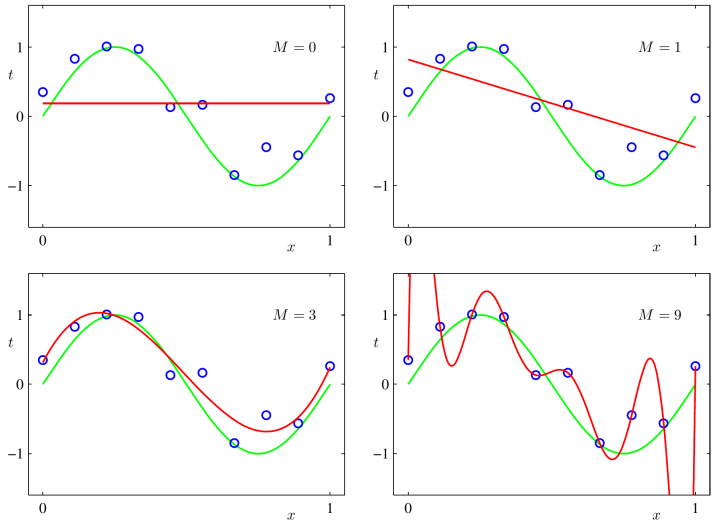
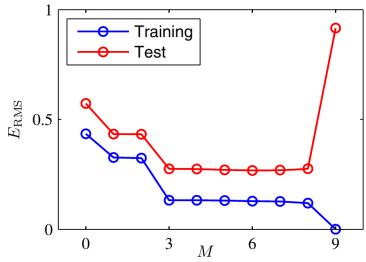


Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

Polynomiální regrese

Figure 1.5 Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of M.



Polynomiální regrese

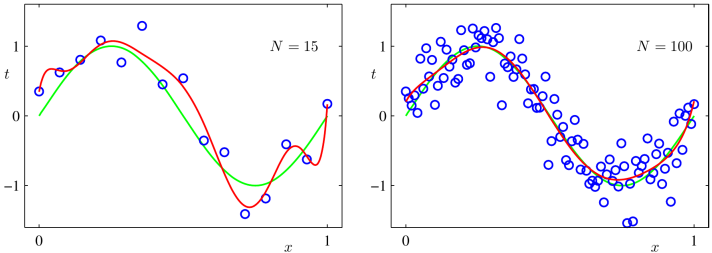


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

Polynomiální regrese

- ve strojovém učení se procesu optimalizace říká trénování nebo učení
- situaci, kdy klesá chyba při trénování ale roste chyba při testování se říká *přetrénování, overfitting*
- odpovídá to situaci, kdy třeba člověk spíše memoruje poznatky, než aby jim rozuměl
- ukazuje se, že této situaci se dá zabránit, pokud model udržíme dostatečně jednoduchý
- rádi bychom toho ale dosáhli automaticky
- n zvolíme dostatečně velké, ale budeme se snažit, aby se některé parametry vynulovaly nebo aspoň byly hodně malé

Polynomiální regrese - regularizace

- použijeme tzv. *regularizaci*
- ztrátovou funkci

$$L(\vec{w}, \sigma, \vec{x}, \vec{y}) = \left\| \vec{P}(\vec{w}, \vec{x}) - \vec{y} \right\|_2^2$$

- doplníme o regularizační člen

$$L(\vec{w}, \sigma, \lambda_1, \vec{x}, \vec{y}) = \left\| \vec{P}(\vec{w}, \vec{x}) - \vec{y} \right\|_2^2 + \frac{\lambda_1}{2} \left\| \vec{w} \right\|_2^2$$

- ten bude penalizovat velké hodnoty parametrů $\vec{w}$
- minimalizace metodou nejvyššího spádu pak bude vypadat takto

$$\vec{w}_j^{(k+1)} = \vec{w}_j^{(k)} - 2\gamma \left(\vec{P}(\vec{w}, \vec{x}) - \vec{y} \right) \cdot x^j - \lambda_1 w_j$$

Polynomiální regrese

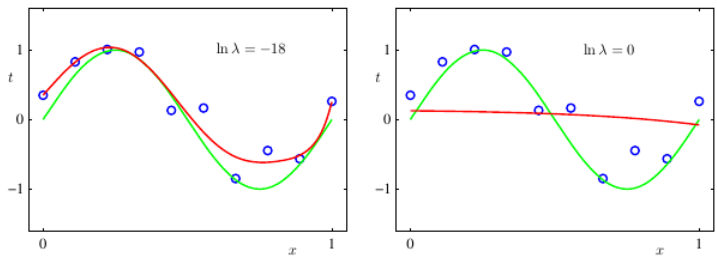


Figure 1.7 Plots of $M = 9$ polynomials fitted to the data set shown in Figure 1.2 using the regularized error function (1.4) for two values of the regularization parameter λ corresponding to $\ln \lambda = -18$ and $\ln \lambda = 0$. The case of no regularizer, i.e., $\lambda = 0$, corresponding to $\ln \lambda = -\infty$, is shown at the bottom right of Figure 1.4.

Example 5

- stáhněte si příklady z repozitáře

```
git clone git@gitlab.com:oberhuber.tomas/optimizations.git
```

- vygenerujte data pro polynomiální regresi pomocí skriptu

```
polynomial-regresion-data.py
```

- proveďte výpočet polynomiální regrese s pomocí skriptu

```
polynomial-regresion.py
```

Uděláme krok zpět od polynomů k lineární regresi ...

Budeme se zabývat lineární úlohou ale s vektory:

- mějme data $(\vec{x}_i, y_i)_{i=1}^N$, kde $\vec{x}_i \in \mathbb{R}^n$ a $y_i \in \mathbb{R}$ a $N > n$
- předpokládáme následující model

$$y_i = \vec{x}_i^T \vec{w} + e_i \text{ pro } i = 1, \ldots, N,$$

kde

$$\vec{w} = (w_1, \ldots, w_n),$$

a $\vec{e}$ je šum nebo chyba měření s normálním rozdělením.

- vektorově lze zapsat jako

$$\vec{y} = \mathbb{X}\vec{w} + \vec{e},$$

kde

$$\mathbb{X} = \begin{pmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_N^T \end{pmatrix} \in \mathbb{R}^{N,n}$$

Lineární regrese - metoda nejmenších čtverců

- již víme, že parametry $\vec{w}$ lze hledat pomocí minimalizace ztrátové funkce

$$L(\vec{w}, \mathbb{X}, \vec{y}) = \frac{1}{2} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2,$$

tj.

$$\vec{w}^* = \arg\min_{\vec{w}} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2$$

- jde o metoda nejmenších čtverců
- před samotným odvozením

$$\nabla_{\vec{w}} L(\vec{w}, \mathbb{X}, \vec{y}) = \nabla_{\vec{w}} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2$$

provedeme pomocné výpočty

Lineární regrese - metoda nejmenších čtverců

Remark 6

Buď $L = L(\vec{x}) : \mathbb{R}^n \to \mathbb{R}^m$. *Pak*

$$\nabla_{\vec{x}} L = \frac{\partial L}{\partial \vec{x}} = \begin{pmatrix} \frac{\partial L_1}{\partial x_1} & \frac{\partial L_1}{\partial x_2} & \cdots & \frac{\partial L_1}{\partial x_n} \\ \frac{\partial L_2}{\partial x_1} & \frac{\partial L_2}{\partial x_2} & \cdots & \frac{\partial L_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L_m}{\partial x_1} & \frac{\partial L_m}{\partial x_2} & \cdots & \frac{\partial L_m}{\partial x_n} \end{pmatrix}$$

Lineární regrese - metoda nejmenších čtverců

Remark 7

Pro $\mathbb{A} \in \mathbb{R}^{m,n}$ *a* $\vec{w} \in \mathbb{R}^n$ *platí*

$$\frac{\partial \mathbb{A}\vec{w}}{\partial \vec{w}} = \begin{pmatrix} \frac{\partial}{\partial \vec{w}} \left(\sum_{j=1}^{n} a_{1j} w_j \right) \\ \vdots \\ \frac{\partial}{\partial \vec{w}} \left(\sum_{j=1}^{n} a_{mj} w_j \right) \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial \vec{w}} \left(a_{11} w_1 + a_{12} w_2 + \ldots a_{1n} w_n \right) \\ \vdots \\ \frac{\partial}{\partial \vec{w}} \left(a_{m1} w_1 + a_{n2} w_2 + \ldots a_{mn} w_n \right) \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{pmatrix} = \mathbb{A}$$

Remark 8

Pro $\mathbb{A} \in \mathbb{R}^{n,n}$ *a* $\vec{w} \in \mathbb{R}^n$ *platí*

$$\begin{aligned}
\frac{\partial \vec{w}^T \mathbb{A} \vec{w}}{\partial \vec{w}} &= \frac{\partial}{\partial \vec{w}} \left(\sum_{i,j=1}^{n} a_{ij} w_i w_j \right) \\
&= \left(\frac{\partial \sum_{i,j=1}^{n} a_{ij} w_i w_j}{\partial w_1}, \ldots, \frac{\partial \sum_{i,j=1}^{n} a_{ij} w_i w_j}{\partial w_n} \right) \\
&= \left(\sum_{j=1}^{n} a_{1j} w_j + \sum_{i=1}^{n} a_{i1} w_i, \ldots, \sum_{j=1}^{n} a_{nj} w_j + \sum_{i=1}^{n} a_{in} w_i \right) \\
&= \vec{w}^T \left(\mathbb{A} + \mathbb{A}^T \right)
\end{aligned}$$

Remark 9

Pro $\mathbb{A} \in \mathbb{R}^{m,n}$ *a* $\vec{w} \in \mathbb{R}^n, \vec{v} \in \mathbb{R}^m$ *platí*

$$\begin{aligned}
\frac{\partial \vec{v}^T \mathbb{A} \vec{w}}{\partial \vec{w}} &= \frac{\partial}{\partial \vec{w}} \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} v_i w_j \right) \\
&= \left(\frac{\partial \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} v_i w_j}{\partial w_1}, \ldots, \frac{\partial \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} v_i w_j}{\partial w_n} \right) \\
&= \left(\sum_{i=1}^{m} a_{i1} v_i, \ldots, \sum_{i=1}^{m} a_{in} v_i \right) \\
&= \left(\mathbb{A}^T \vec{v} \right)^T = \vec{v}^T \mathbb{A}
\end{aligned}$$

Remark 10

Pro $\mathbb{A} \in \mathbb{R}^{n,m}$ *a* $\vec{w} \in \mathbb{R}^n, \vec{v} \in \mathbb{R}^m$ *platí*

$$\begin{aligned}
\frac{\partial \vec{w}^T \mathbb{A} \vec{v}}{\partial \vec{w}} &= \frac{\partial}{\partial \vec{w}} \left(\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} w_i v_j \right) \\
&= \left(\frac{\partial \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} w_i v_j}{\partial w_1}, \ldots, \frac{\partial \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} w_i v_j}{\partial w_n} \right) \\
&= \left(\sum_{j=1}^{m} a_{1j} v_j, \ldots, \sum_{j=1}^{m} a_{nj} v_j \right) \\
&= \left(\mathbb{A} \vec{v} \right)^T
\end{aligned}$$

Lineární regrese - metoda nejmenších čtverců

Celkem jsme tedy ukázali:
Pro $\mathbb{A} \in \mathbb{R}^{m,n}$ a $\vec{w} \in \mathbb{R}^n$, $\vec{v} \in \mathbb{R}^m$ platí

$$\begin{aligned}
\frac{\partial \mathbb{A}\vec{w}}{\partial \vec{w}} &= \mathbb{A}, \\
\frac{\partial \vec{w}^T \mathbb{A}\vec{w}}{\partial \vec{w}} &= \vec{w}^T \left(\mathbb{A} + \mathbb{A}^T\right), \\
\frac{\partial \vec{v}^T \mathbb{A}\vec{w}}{\partial \vec{w}} &= \vec{v}^T \mathbb{A}, \\
\frac{\partial \vec{w}^T \mathbb{A}\vec{v}}{\partial \vec{w}} &= \left(\mathbb{A}\vec{v}\right)^T.
\end{aligned}$$

Lineární regrese - metoda nejmenších čtverců

- nyní zpět k metodě nejmenších čtverců
- ztrátovou funkci $L(\vec{w}, \mathbb{X}, \vec{y}) = \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2$ upravíme na tvar

$$\begin{aligned} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2 &= \left(\mathbb{X}\vec{w} - \vec{y} \right)^T \left(\mathbb{X}\vec{w} - \vec{y} \right) \\ &= \left(\mathbb{X}\vec{w} \right)^T \left(\mathbb{X}\vec{w} \right) - \vec{y}^T \mathbb{X}\vec{w} - \left(\mathbb{X}\vec{w} \right)^T \vec{y} + \vec{y}^T \vec{y} \\ &= \vec{w}^T \mathbb{X}^T \mathbb{X}\vec{w} - \vec{y}^T \mathbb{X}\vec{w} - \vec{w}^T \mathbb{X}^T \vec{y} + \vec{y}^T \vec{y} \end{aligned}$$

- dále použijeme vztah

$$\vec{w}^T \mathbb{X}^T \vec{y} = \vec{y}^T \mathbb{X}\vec{w}$$

neboť jde o skalární výrazy

Lineární regrese - metoda nejmenších čtverců

- dostáváme tedy

$$\left\|\mathbb{X}\vec{w} - \vec{y}\right\|_2^2 = \vec{w}^T\mathbb{X}^T\mathbb{X}\vec{w} - 2\vec{y}^T\mathbb{X}\vec{w} + \vec{y}^T\vec{y}$$

- a následně

$$\nabla_{\vec{w}}\left\|\mathbb{X}\vec{w} - \vec{y}\right\|_2^2 = 2\vec{w}^T\mathbb{X}^T\mathbb{X} - 2\vec{y}^T\mathbb{X}$$

kde jsme využili dříve odovozené vztahy a fakt, že

$$\left(\mathbb{X}^T\mathbb{X}\right)^T = \mathbb{X}^T\mathbb{X}$$

Lineární regrese - metoda nejmenších čtverců

- jelikož vztah pro gradient je lineární, můžeme řešit přímo

$$0 = \nabla_{\vec{w}} L = 2\mathbb{X}^T \mathbb{X}\vec{w} - 2\mathbb{X}^T \vec{y}$$

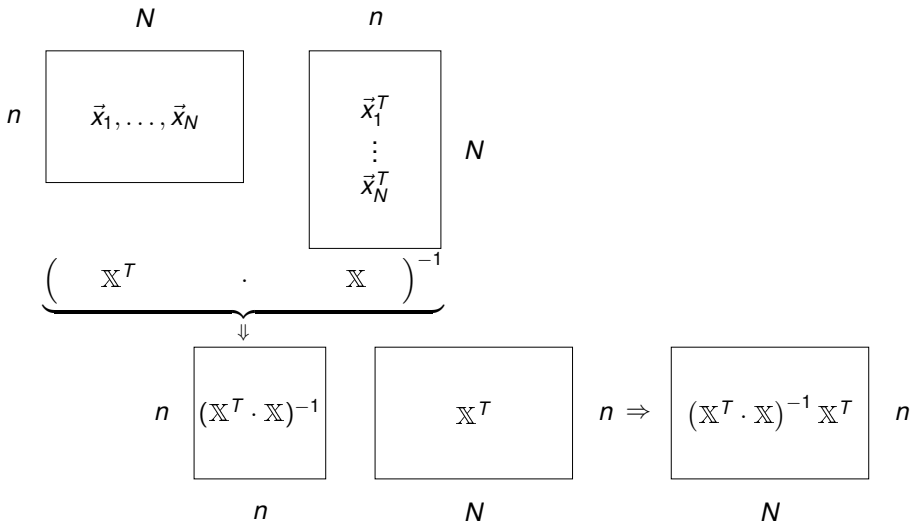
- a tedy

$$\mathbb{X}^T \mathbb{X}\vec{w} = \mathbb{X}^T \vec{y}$$

tj.

$$\vec{w} = \underbrace{\left(\mathbb{X}^T \mathbb{X}\right)^{-1}}_{\in \mathbb{R}^{n,n}} \underbrace{\mathbb{X}^T \vec{y}}_{\in \mathbb{R}^n}$$

Lineární regrese - metoda nejmenších čtverců



Definition 11
Pokud existuje matice $\left(\mathbb{X}^T \cdot \mathbb{X}\right)^{-1}$, pak matice

$$\mathbb{X}^+ = \left(\mathbb{X}^T \cdot \mathbb{X}\right)^{-1} \mathbb{X}^T$$

se nazývá Moorova-Penrosova inverze nebo také pseudoinverze.

Lineární regrese - metoda nejmenších čtverců

- je-li $n > N$, matice $\mathbb{X}^T\mathbb{X}$ může být špatně podmíněná nebo singulární
- řešením je opět snaha omezit počet optimalizovaných parametrů $\vec{w}$ přidáním Tichonovovy regularizace

$$L(\vec{w}, \mathbb{X}, \vec{y}) = \frac{1}{2} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2 + \frac{\lambda_1}{2} \left\| \vec{w} \right\|_2^2,$$

- gradient má potom tvar

$$\nabla_{\vec{w}} L = \mathbb{X}^T\mathbb{X}\vec{w} + \lambda_1 \vec{w} - \mathbb{X}\vec{y}$$

- a řešením je

$$\vec{w} = \left(\mathbb{X}^T\mathbb{X} + \lambda_1 \mathbb{I} \right)^{-1} \mathbb{X}\vec{y}$$

Lineární regrese - bias

- model lze vylepšit přidáním tzv. *biasu*

$$y_i = x_i^T \vec{w} + b + e_i,$$

kde neznámé parametry jsou nyní $(b, \vec{w})$

- ztrátové funkce pak má tvar

$$L(\vec{w}, b, \mathbb{X}, \vec{y}) = \frac{1}{2} \left\| \mathbb{X}\vec{w} + \vec{b} - \vec{y} \right\|_2^2, \text{ kde } \vec{b} = (b, \ldots, b)$$

- a tedy gradient je

$$\begin{aligned} \nabla_{\vec{w}} L &= \mathbb{X}^T \left(\mathbb{X}\vec{w} + \vec{b} - \vec{y} \right), \\ \nabla_b L &= \sum_{i=1}^{N} \left(\vec{x}_i^T \vec{w} + b - y_i \right) \end{aligned}$$

Lineární regrese - bias

- celkový předpis pro GD lze přepsat takto

$$\begin{aligned} \vec{w} &= \vec{w} - \gamma_1 \sum_{i=1}^{N} \left(\vec{x}_i^T \vec{w} + b - y_i \right) \vec{x}_i - \lambda_1 \vec{w}, \\ b &= b - \gamma_2 \sum_{i=1}^{N} \left(\vec{x}_i^T \vec{w} + b - y_i \right) - \lambda_2 b \end{aligned}$$

kde γ_1, γ_2 jsou relaxační parametry a λ_1, λ_2 jsou parametry z Tichonovovy regularizace

Lineární regrese - bias

Pro zjednodušení se často používá následující zápis:

$$\vec{y} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \dots & x_{Nn} \end{pmatrix} \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_n \end{pmatrix} + \vec{e} = \overline{\mathbb{X}}\vec{w} + \vec{e},$$

kde

$$\overline{\mathbb{X}} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & \dots & x_{Nn} \end{pmatrix} \text{ a } \vec{\overline{w}} = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_n \end{pmatrix}.$$

- díky tomu pak není nutné ošetřovat bias zvlášť
- v následujícím nebudeme rozlišovat $\mathbb{X}, \vec{w}$ a $\overline{\mathbb{X}}, \vec{\overline{w}}$ a implicitně budeme předpokládat, že model je možné uvažovat s biasem, pokud to dává smysl

- Místo L_2 normy se také často používá tzv. **mean squared error = MSE**:

$$L(\vec{w}, \mathbb{X}, \vec{y}) = \frac{1}{N} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2 .$$

- Výhoda MSE je nezávilost na počtu dat a lepší numerická stabilita.
- Jako **ridge regression** se označuje regrese s L_2 Tichonovou regularizací, tj. např.

$$L(\vec{w}, \mathbb{X}, \vec{y}) = \frac{1}{N} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2 + \lambda_1 \left\| \vec{w} \right\|_2^2 .$$

- Jako **LASSO = Least Absolute Shrinkage and Selection Operator** se označuje regrese s L_1 Tichonovou regularizací, tj. např.

$$L(\vec{w}, \mathbb{X}, \vec{y}) = \frac{1}{N} \left\| \mathbb{X}\vec{w} - \vec{y} \right\|_2^2 + \lambda_1 \left\| \vec{w} \right\|_1 .$$

Lineární regrese

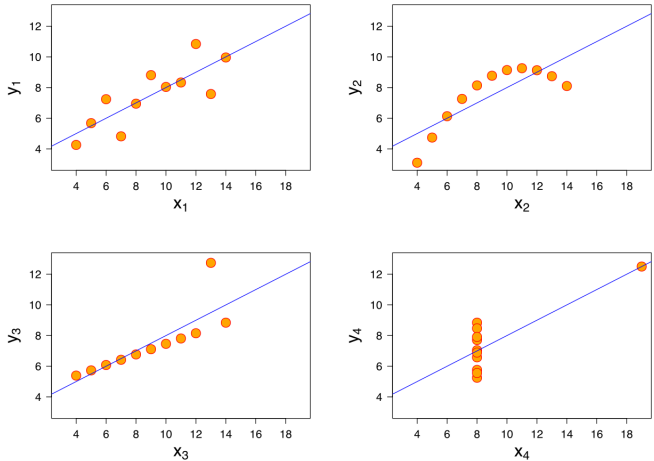


Figure: Anscombe quartet - ukazuje omezení lineární regrese

Pomocí lineární regrese lze řešit i úlohu polynomiální regrese:

- jelikož platí, že

$$\vec{P}(\vec{w}, \vec{x}) = \begin{pmatrix} P(\vec{w}, x_i) \\ \vdots \\ P(\vec{w}, x_n) \end{pmatrix} = \mathbb{V}^{(d)}_{x_0,\ldots,x_N} \vec{w},$$

- pak

$$\left\| \vec{P}(\vec{w}, \vec{x}) - \vec{y} \right\|_2^2 = \left\| \mathbb{V}^{(d)}_{x_0,\ldots,x_N} \vec{w} - \vec{y} \right\|_2^2,$$

- a lze psát

$$\vec{w} = \left(\left(\mathbb{V}^{(d)}_{x_0,\ldots,x_N} \right)^T \mathbb{V}^{(d)}_{x_0,\ldots,x_N} \right)^{-1} \left(\mathbb{V}^{(d)}_{x_0,\ldots,x_N} \right)^T \vec{y}$$

Example 12

- vygenerujte data pro lineární regresi pomocí skriptu

```
linear-regresion-data.py
```

- proveďte výpočet lineární regrese s pomocí skriptu

```
linear-regresion.py
```

Stochastic gradient descent - SGD

- N (=vstupní data) může být hodně velké
- pak by každá iterace byla velmi náročná na výpočet
- často se proto používá *stochastic gradient descent* - SGD
- do každé iterace k si vybere jen malá podmnožina indexů $S_k \subset \{1, \ldots, N\}$
- této podmnožině se říká batch nebo mini-batch

Stochastic gradient descent - SGD

- předpis pak má tvar

$$\vec{w}^{(k+1)} = \vec{w}^{(k)} - \gamma \sum_{i \in S_k} \left(\vec{x}_i^T \vec{w}^{(k)} - y_i \right) \vec{x}_i - \lambda \vec{w}^{(k)},$$

Odkazy:
Důkaz konvergence
H. Robbins, S. Monro, *A stochastic approximation method*, The annals of mathematical statistics, 1951.
J. Kiefer, J. Wolfowitz *Stochastic estimation of the maximum of a regression function*, The Annals of Mathematical Statistics, 1952.