

Lineární modely pro regresi a klasifikaci. Učení.

Tomáš Svoboda and Petr Pošík

thanks to Matěj Hoffmann, Daniel Novák, Filip Železný, Ondřej Drbohlav

Vision for Robots and Autonomous Systems, Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague

22. května 2025

Učení s učitelem (supervised learning)

Máme k dispozici trénovací multi-množinu příkladů. Správné „odpovědi“ (skryté stavy, třídy, hodnoty, které chceme odhadovat) jsou *známé* pro všechny trénovací příklady.

Klasifikace :

- ▶ Závislá proměnná je nominální
- ▶ Příklady: predikce spamu/hamu na základě obsahu emailu, predikce 0/1/.../9 na základě obrazu čísla, atd.

Regrese :

- ▶ Závislá proměnná je kvantitativní/spojitá
- ▶ Příklady: predikce teploty v Praze na základě času a data, predikce výšky osoby na základě hmotnosti a pohlaví, atd.

2 / 45

Notes

Supervised learning Existují i další typy strojového učení:

- Self-supervised
- Unsupervised (bez učitele)
- Weakly supervised
- ...

ale tato přednáška pokrývá jen učení s učitelem.

Učení s učitelem (supervised learning)

Máme k dispozici trénovací multi-množinu příkladů. Správné „odpovědi“ (skryté stavy, třídy, hodnoty, které chceme odhadovat) jsou *známé* pro všechny trénovací příklady.

Klasifikace :

- ▶ Závislá proměnná je nominální
- ▶ Příklady: predikce spamu/hamu na základě obsahu emailu, predikce 0/1/.../9 na základě obrazu čísla, atd.

Regrese :

- ▶ Závislá proměnná je kvantitativní/spojitá
- ▶ Příklady: predikce teploty v Praze na základě času a data, predikce výšky osoby na základě hmotnosti a pohlaví, atd.

2 / 45

Notes

Supervised learning Existují i další typy strojového učení:

- Self-supervised
- Unsupervised (bez učitele)
- Weakly supervised
- ...

ale tato přednáška pokrývá jen učení s učitelem.

Učení s učitelem (supervised learning)

Máme k dispozici trénovací multi-množinu příkladů. Správné „odpovědi“ (skryté stavy, třídy, hodnoty, které chceme odhadovat) jsou *známé* pro všechny trénovací příklady.

Klasifikace :

- ▶ Závislá proměnná je nominální
- ▶ Příklady: predikce spamu/hamu na základě obsahu emailu, predikce 0/1/.../9 na základě obrazu čísla, atd.

Regrese :

- ▶ Závislá proměnná je kvantitativní/spojitá
- ▶ Příklady: predikce teploty v Praze na základě času a data, predikce výšky osoby na základě hmotnosti a pohlaví, atd.

2 / 45

Notes

Supervised learning Existují i další typy strojového učení:

- Self-supervised
- Unsupervised (bez učitele)
- Weakly supervised
- ...

ale tato přednáška pokrývá jen učení s učitelem.

Učení: minimalizace empirického rizika

- Mějme množinu parametrizovaných strategií $\delta: \mathcal{X} \rightarrow \mathcal{D}$ a ztrátovou funkci $\ell: \mathcal{S} \times \mathcal{D} \rightarrow \mathbb{R}$. Kvalitu každé strategie δ popisuje riziko

$$R(\delta) = \sum_{s \in \mathcal{S}} \sum_{x \in \mathcal{X}} P(x, s) \ell(s, \delta(x)),$$

ale P neznáme.

- Používáme proto tzv. empirické riziko, tj. průměrnou ztrátu na trénovací (multi)množině $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^N$, $x \in \mathcal{X}$, $s \in \mathcal{S}$:

$$R_{\text{emp}}(\delta) = \frac{1}{N} \sum_{(x^{(i)}, s^{(i)}) \in \mathcal{T}} \ell(s^{(i)}, \delta(x^{(i)})).$$

- Optimální strategie $\delta^* = \operatorname{argmin}_{\delta} R_{\text{emp}}(\delta)$.
- Předpokládáme, že data $\mathcal{T}$ pocházejí z rozdělení $P(x, s)$.

3 / 45

Notes

Příklady metod: Perceptron, neuronové množiny, klasifikační stromy, ...

V zásadě jde o statistiku, data neodpovídající předpokládanému rozdělení jsou vždy problém. Můžeme tomu trochu pomoci, když metody uděláme robustnější, aby lépe generalizovaly. Pamatujete se na regularizační trik z minulého týdne (Laplaceovo vyhlazování)?

Učení: minimalizace empirického rizika

- Mějme množinu parametrizovaných strategií $\delta: \mathcal{X} \rightarrow \mathcal{D}$ a ztrátovou funkci $\ell: \mathcal{S} \times \mathcal{D} \rightarrow \mathbb{R}$. Kvalitu každé strategie δ popisuje riziko

$$R(\delta) = \sum_{s \in \mathcal{S}} \sum_{x \in \mathcal{X}} P(x, s) \ell(s, \delta(x)),$$

ale P neznáme.

- Používáme proto tzv. **empirické riziko**, tj. průměrnou ztrátu na trénovací (multi)množině $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^N$, $x \in \mathcal{X}$, $s \in \mathcal{S}$:

$$R_{\text{emp}}(\delta) = \frac{1}{N} \sum_{(x^{(i)}, s^{(i)}) \in \mathcal{T}} \ell(s^{(i)}, \delta(x^{(i)})).$$

- Optimální strategie $\delta^* = \operatorname{argmin}_{\delta} R_{\text{emp}}(\delta)$.
- Předpokládáme, že data $\mathcal{T}$ pocházejí z rozdělení $P(x, s)$.

3 / 45

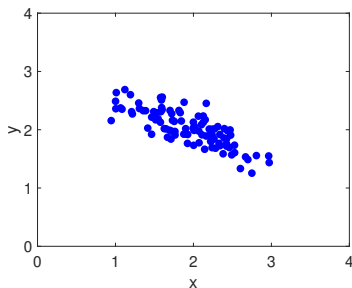
Notes

Příklady metod: Perceptron, neuronové množiny, klasifikační stromy, ...

V zásadě jde o statistiku, data neodpovídající předpokládanému rozdělení jsou vždy problém. Můžeme tomu trochu pomoci, když metody uděláme robustnější, aby lépe generalizovaly. Pamatujete se na regularizační trik z minulého týdne (Laplaceovo vyhlazování)?

Kvíz: Proložení bodů přímkou

Chceme přímkou ve tvaru $\hat{y} = w_0 + w_1x$ proložit následující data:



Nejlépe bude datům odpovídat přímkou s parametry

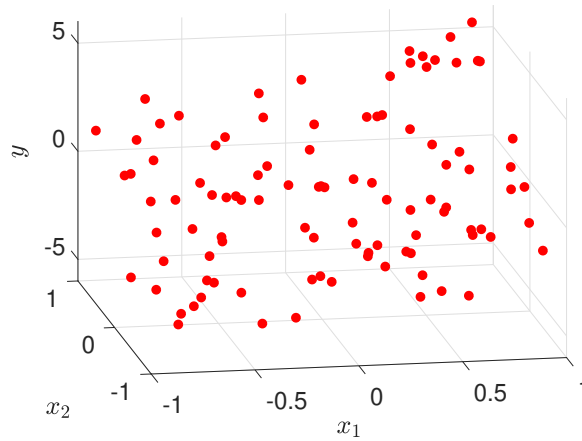
A $w_0 = -1, w_1 = -2$

B $w_0 = -\frac{1}{2}, w_1 = 1$

C $w_0 = 3, w_1 = -\frac{1}{2}$

D $w_0 = 2, w_1 = \frac{1}{3}$

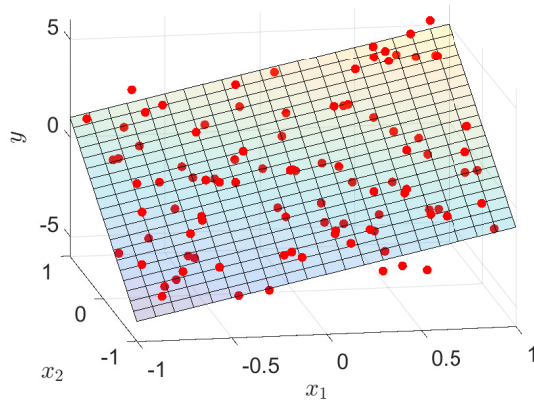
Lineární regrese: ilustrace



Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné hodnoty výstupní proměnné $y^{(i)}$.

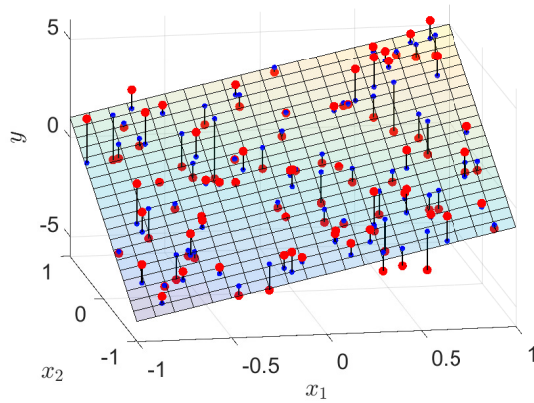
Např. prokládání roviny daty z lidarů.

Lineární regrese: ilustrace



Pro tuto datovou sadu bychom chtěli nalézt lineární model, ...

Lineární regrese: ilustrace



... který minimalizuje odchylky **odhadů** od **správných hodnot**.

Např. prokládání roviny daty z lidarů.

Lineární algebra přeformulovaná do jazyka strojového učení.

Regresní úloha je úloha učení s učitelem, tj.

- ▶ máme trénovací (multi-)množinu $\mathcal{T} = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(N)}, y^{(N)})\}$, kde
- ▶ hodnoty $y^{(i)}$ jsou *kvantitativní*, často *spojité* (narozdíl od klasifikačních úloh, kde jsou $y^{(i)}$ nominální).
- ▶ Cílem je vytvořit model vztahu mezi nezávislými proměnnými (vstupy) $\vec{x} = (x_1, \dots, x_D)$ a závislou proměnnou (výstupem) y .

Lineární regrese

Lineární regrese používá regresní model, který předpokládá (a učí se) lineární závislosti mezi vstupy a výstupem:

$$\hat{y} = \delta(\vec{x}) = w_0 + w_1 x_1 + \dots + w_D x_D = w_0 + \langle \vec{w}, \vec{x} \rangle = w_0 + \vec{w}^\top \vec{x},$$

kde

- ▶ $\hat{y}$ je *predikce* modelu (*odhad* skutečné hodnoty y),
- ▶ $\delta(\vec{x})$ je rozhodovací strategie (v tomto případě lineární model),
- ▶ $w_0, \dots, w_D$ jsou koeficienty lineární funkce (váhy), w_0 je *absolutní člen* (*bias*),
- ▶ $\langle \vec{w}, \vec{x} \rangle$ je *skalární součin* vektorů $\vec{w}$ a $\vec{x}$ (dot product),
- ▶ který lze také spočítat jako maticový součin $\vec{w}^\top \vec{x}$, pokud jsou $\vec{w}$ a $\vec{x}$ *sloupcové vektory*, tj. matice velikosti $[D \times 1]$.

Poznámky k notaci

Homogenní souřadnice :

- Pokud přidáme “1” jako první prvek do všech vektorů $\vec{x}$ tak, že $\vec{x} = (1, x_1, \dots, x_D)$, a
- pokud zahrneme absolutní člen w_0 do vektoru $\vec{w}$ tak, že $\vec{w} = (w_0, w_1, \dots, w_D)$, pak

$$\hat{y} = \delta(\vec{x}) = w_0 \cdot 1 + w_1 x_1 + \dots + w_D x_D = \langle \vec{w}, \vec{x} \rangle = \vec{w}^\top \vec{x}.$$

Maticová notace: Pokud uspořádáme data $\mathcal{T}$ do matic X a Y tak, že

$$X = \begin{pmatrix} 1 & \dots & 1 \\ \vec{x}^{(1)} & \dots & \vec{x}^{(N)} \end{pmatrix} \quad \text{a} \quad Y = (y^{(1)}, \dots, y^{(N)}),$$

pak můžeme zapsat dávkový výpočet predikcí pro všechna pozorování v X jako

$$\hat{Y} = (\delta(\vec{x}^{(1)}), \dots, \delta(\vec{x}^{(N)})) = (\vec{w}^\top \vec{x}^{(1)}, \dots, \vec{w}^\top \vec{x}^{(N)}) = \vec{w}^\top X.$$

Notes

Jaké jsou rozměry $\hat{y}$, $\vec{w}$, X ?

Poznámky k notaci

Homogenní souřadnice :

- Pokud přidáme “1” jako první prvek do všech vektorů $\vec{x}$ tak, že $\vec{x} = (1, x_1, \dots, x_D)$, a
- pokud zahrneme absolutní člen w_0 do vektoru $\vec{w}$ tak, že $\vec{w} = (w_0, w_1, \dots, w_D)$, pak

$$\hat{y} = \delta(\vec{x}) = w_0 \cdot 1 + w_1 x_1 + \dots + w_D x_D = \langle \vec{w}, \vec{x} \rangle = \vec{w}^\top \vec{x}.$$

Maticová notace Pokud uspořádáme data $\mathcal{T}$ do matic X a Y tak, že

$$X = \begin{pmatrix} 1 & \dots & 1 \\ \vec{x}^{(1)} & \dots & \vec{x}^{(N)} \end{pmatrix} \quad \text{a} \quad Y = \begin{pmatrix} y^{(1)} & \dots & y^{(N)} \end{pmatrix},$$

pak můžeme zapsat dávkový výpočet predikcí pro všechna pozorování v X jako

$$\hat{Y} = \left(\delta(\vec{x}^{(1)}), \dots, \delta(\vec{x}^{(N)}) \right) = \left(\vec{w}^\top \vec{x}^{(1)}, \dots, \vec{w}^\top \vec{x}^{(N)} \right) = \vec{w}^\top X.$$

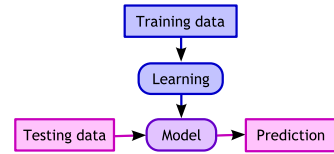
Notes

Jaké jsou rozměry $\hat{y}$, $\vec{w}$, X ?

Dvě pracovní fáze ML modelů

Každý ML model má dvě pracovní fáze:

1. učení (trénování) strategie δ a
2. použití strategie δ (testování, tvorba predikcí).



Na strategii δ lze nahlížet jako na funkci 2 proměnných: $\delta(\vec{x}, \vec{w})$.

Aplikace modelů (Inference): Známe-li $\vec{w}$, můžeme měnit $\vec{x}$, a zjišťovat tak odhady:

$$\hat{y} = \delta(\vec{x}, \vec{w}) = \delta_{\vec{w}}(\vec{x}).$$

Učení modelu: Známe-li $\mathcal{T}$, můžeme ladit parametry $\vec{w}$ tak, aby model odpovídal datům:

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} R_{\text{emp}}(\delta_{\vec{w}}) = \underset{\vec{w}}{\operatorname{argmin}} J(\vec{w}, \mathcal{T}),$$

kde obvykle $J(\vec{w}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} \ell(y, \delta(\vec{x}, \vec{w}))$. Jak model naučit?

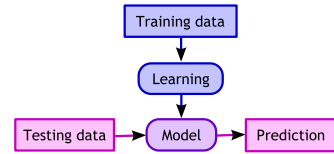
Notes

- $\delta(\vec{x}, \vec{w})$ represents a whole family of strategies if $\vec{w}$ is not fixed.
- By fixing $\vec{w}$ we chose a particular strategy from this family.
- Empirical risk evaluates prediction error on all data points.

Dvě pracovní fáze ML modelů

Každý ML model má dvě pracovní fáze:

1. učení (trénování) strategie δ a
2. použití strategie δ (testování, tvorba predikcí).



Na strategii δ lze nahlížet jako na funkci 2 proměnných: $\delta(\vec{x}, \vec{w})$.

Aplikace modelu (Inference): Známe-li $\vec{w}$, můžeme měnit $\vec{x}$, a zjišťovat tak odhady:

$$\hat{y} = \delta(\vec{x}, \vec{w}) = \delta_{\vec{w}}(\vec{x}).$$

Učení modelu: Známe-li $\mathcal{T}$, můžeme ladit parametry $\vec{w}$ tak, aby model odpovídal datům:

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} R_{\text{emp}}(\delta_{\vec{w}}) = \underset{\vec{w}}{\operatorname{argmin}} J(\vec{w}, \mathcal{T}),$$

kde obvykle $J(\vec{w}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} \ell(y, \delta(\vec{x}, \vec{w}))$. Jak model naučit?

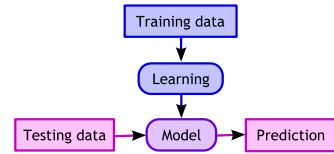
Notes

- $\delta(\vec{x}, \vec{w})$ represents a whole family of strategies if $\vec{w}$ is not fixed.
- By fixing $\vec{w}$ we chose a particular strategy from this family.
- Empirical risk evaluates prediction error on all data points.

Dvě pracovní fáze ML modelů

Každý ML model má dvě pracovní fáze:

1. učení (trénování) strategie δ a
2. použití strategie δ (testování, tvorba predikcí).



Na strategii δ lze nahlížet jako na funkci 2 proměnných: $\delta(\vec{x}, \vec{w})$.

Aplikace modelu (Inference): Známe-li $\vec{w}$, můžeme měnit $\vec{x}$, a zjišťovat tak odhady:

$$\hat{y} = \delta(\vec{x}, \vec{w}) = \delta_{\vec{w}}(\vec{x}).$$

Učení modelu: Známe-li $\mathcal{T}$, můžeme ladit parametry $\vec{w}$ tak, aby model odpovídal datům:

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} R_{\text{emp}}(\delta_{\vec{w}}) = \underset{\vec{w}}{\operatorname{argmin}} J(\vec{w}, \mathcal{T}),$$

kde obvykle $J(\vec{w}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} \ell(y, \delta(\vec{x}, \vec{w}))$. Jak model naučit?

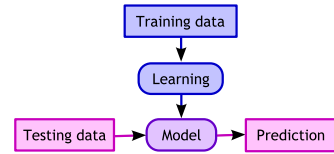
Notes

- $\delta(\vec{x}, \vec{w})$ represents a whole family of strategies if $\vec{w}$ is not fixed.
- By fixing $\vec{w}$ we chose a particular strategy from this family.
- Empirical risk evaluates prediction error on all data points.

Dvě pracovní fáze ML modelů

Každý ML model má dvě pracovní fáze:

1. učení (trénování) strategie δ a
2. použití strategie δ (testování, tvorba predikcí).



Na strategii δ lze nahlížet jako na funkci 2 proměnných: $\delta(\vec{x}, \vec{w})$.

Aplikace modelu (Inference): Známe-li $\vec{w}$, můžeme měnit $\vec{x}$, a zjišťovat tak odhady:

$$\hat{y} = \delta(\vec{x}, \vec{w}) = \delta_{\vec{w}}(\vec{x}).$$

Učení modelu: Známe-li $\mathcal{T}$, můžeme ladit parametry $\vec{w}$ tak, aby model odpovídal datům:

$$\vec{w}^* = \operatorname{argmin}_{\vec{w}} R_{\text{emp}}(\delta_{\vec{w}}) = \operatorname{argmin}_{\vec{w}} J(\vec{w}, \mathcal{T}),$$

kde obvykle $J(\vec{w}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} \ell(y, \delta(\vec{x}, \vec{w}))$. Jak model naučit?

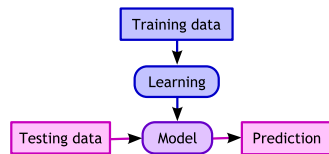
Notes

- $\delta(\vec{x}, \vec{w})$ represents a whole family of strategies if $\vec{w}$ is not fixed.
- By fixing $\vec{w}$ we chose a particular strategy from this family.
- Empirical risk evaluates prediction error on all data points.

Dvě pracovní fáze ML modelů

Každý ML model má dvě pracovní fáze:

1. učení (trénování) strategie δ a
2. použití strategie δ (testování, tvorba predikcí).



Na strategii δ lze nahlížet jako na funkci 2 proměnných: $\delta(\vec{x}, \vec{w})$.

Aplikace modelu (Inference): Známe-li $\vec{w}$, můžeme měnit $\vec{x}$, a zjišťovat tak odhady:

$$\hat{y} = \delta(\vec{x}, \vec{w}) = \delta_{\vec{w}}(\vec{x}).$$

Učení modelu: Známe-li $\mathcal{T}$, můžeme ladit parametry $\vec{w}$ tak, aby model odpovídal datům:

$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}} R_{\text{emp}}(\delta_{\vec{w}}) = \underset{\vec{w}}{\operatorname{argmin}} J(\vec{w}, \mathcal{T}),$$

kde obvykle $J(\vec{w}, \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(\vec{x}, y) \in \mathcal{T}} \ell(y, \delta(\vec{x}, \vec{w}))$. Jak model naučit?

9 / 45

Notes

- $\delta(\vec{x}, \vec{w})$ represents a whole family of strategies if $\vec{w}$ is not fixed.
- By fixing $\vec{w}$ we chose a particular strategy from this family.
- Empirical risk evaluates prediction error on all data points.

Příklad: Jednoduchá (jednorozměrná) lineární regrese

Jednoduchá regrese

- ▶ $\vec{x}^{(i)} = x^{(i)}$, tj. příklady jsou popsány jedinou vstupní proměnnou (jsou 1-rozměrné).
- ▶ Najděte parametry w_0, w_1 lineárního modelu $\hat{y} = w_0 + w_1 x$, máte-li trénovací (multi-)množinu $\mathcal{T} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$.

Kolik přímek lze proložit N lineárně nezávislými trénovacími příklady?

- ▶ $N = 1$ (1 rovnice, 2 parametry) $\Rightarrow \infty$ lineárních funkcí s nulovou chybou
- ▶ $N = 2$ (2 rovnice, 2 parametry) $\Rightarrow 1$ lineární funkce s nulovou chybou
- ▶ $N \geq 3$ (> 2 rovnice, parametry) $\Rightarrow$ žádná lineární funkce s nulovou chybou
 $\Rightarrow$ ale lze proložit přímkou, která minimizuje velikost chyb $y - \hat{y}$:

$$\vec{w}^* = (w_0^*, w_1^*) = \underset{w_0, w_1}{\operatorname{argmin}} R_{\text{emp}}(w_0, w_1) = \underset{w_0, w_1}{\operatorname{argmin}} J(w_0, w_1, \mathcal{T}).$$

Příklad: Jednoduchá (jednorozměrná) lineární regrese

Jednoduchá regrese

- ▶ $\vec{x}^{(i)} = x^{(i)}$, tj. příklady jsou popsány jedinou vstupní proměnnou (jsou 1-rozměrné).
- ▶ Najděte parametry w_0, w_1 lineárního modelu $\hat{y} = w_0 + w_1 x$, máte-li trénovací (multi-)množinu $\mathcal{T} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$.

Kolik přímek lze proložit N lineárně nezávislými trénovacími příklady?

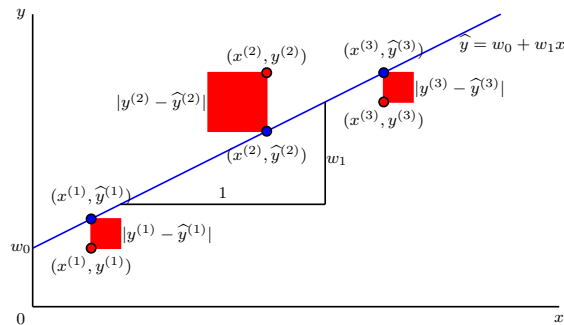
- ▶ $N = 1$ (1 rovnice, 2 parametry) $\Rightarrow \infty$ lineárních funkcí s nulovou chybou
- ▶ $N = 2$ (2 rovnice, 2 parametry) $\Rightarrow 1$ lineární funkce s nulovou chybou
- ▶ $N \geq 3$ (> 2 rovnice, parametry) $\Rightarrow$ žádná lineární funkce s nulovou chybou
 $\Rightarrow$ ale lze proložit přímkou, která minimalizuje velikost chyb $y - \hat{y}$:

$$\vec{w}^* = (w_0^*, w_1^*) = \underset{w_0, w_1}{\operatorname{argmin}} R_{\text{emp}}(w_0, w_1) = \underset{w_0, w_1}{\operatorname{argmin}} J(w_0, w_1, \mathcal{T}).$$

Metoda nejmenších čtverců

Zvolte takové parametry $\vec{w}$, které minimalizují *střední kvadratickou chybu* (mean squared error, MSE)

$$\begin{aligned} J_{MSE}(\vec{w}) &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \hat{y}^{(i)} \right)^2 \\ &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \delta_{\vec{w}}(\vec{x}^{(i)}) \right)^2. \end{aligned}$$



Existuje analytické řešení? Explicitní řešení:

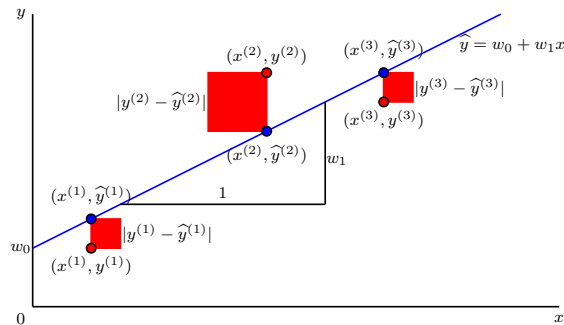
$$w_1 = \frac{\sum_{i=1}^N (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^N (x^{(i)} - \bar{x})^2} = \frac{s_{xy}}{s_x^2} = \frac{\text{kovariance } X \text{ a } Y}{\text{rozptyl } X}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

Metoda nejmenších čtverců

Zvolte takové parametry $\vec{w}$, které minimalizují *střední kvadratickou chybu* (mean squared error, MSE)

$$\begin{aligned} J_{MSE}(\vec{w}) &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \hat{y}^{(i)} \right)^2 \\ &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \delta_{\vec{w}}(\vec{x}^{(i)}) \right)^2. \end{aligned}$$

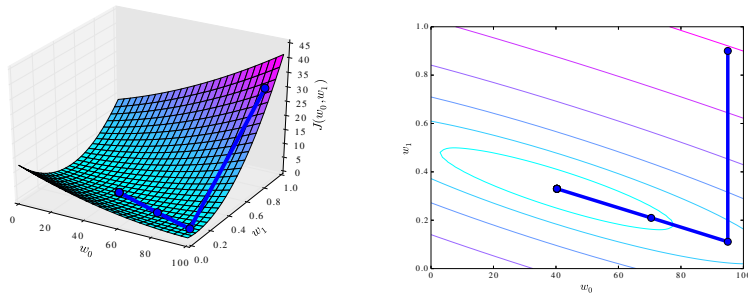


Existuje analytické řešení? **Explicitní řešení:**

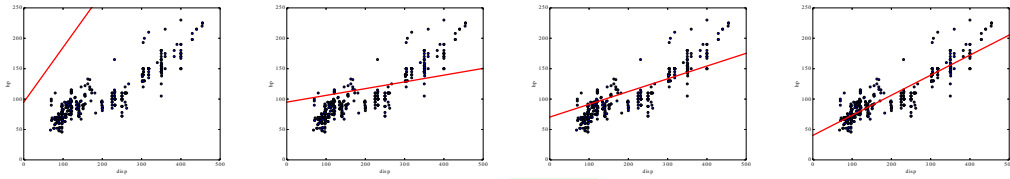
$$w_1 = \frac{\sum_{i=1}^N (x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum_{i=1}^N (x^{(i)} - \bar{x})^2} = \frac{s_{xy}}{s_x^2} = \frac{\text{kovariance } X \text{ a } Y}{\text{rozptyl } X} \quad w_0 = \bar{y} - w_1 \bar{x}$$

Univerzální metoda: minimalizace kriteriální funkce J

„Povrch“ funkce J v prostoru parametrů w_0 and w_1 (pro data níže):



Postupně se zlepšující lineární modely nalezené iterační opt. metodou (BFGS):



Notes

Obrázky dole zleva doprava odpovídají bodům na lomené křivce výše.

Algoritmus nejstrmějšího sestupu (gradient descent)

Pro danou funkci $J(w_0, w_1)$, kterou chceme minimalizovat,

- ▶ začni s libovolnými hodnotami w_0 a w_1 a
- ▶ modifikuj je tak, aby se $J(w_0, w_1)$ zmenšilo, tj.
- ▶ aktualizuj hodnoty w_0 a w_1 posunem v opačném směru, než ukazuje gradient:

$$\vec{w} \leftarrow \vec{w} - \alpha \nabla J(w_0, w_1), \text{ tj.}$$

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} J(w_0, w_1),$$

kde se všechny w_i aktualizují současně a α je tzv. **rychlost učení** (learning rate, step size).

Notes

Gradient ukazuje ve směru nejstrmějšího růstu funkční hodnoty a my chceme minimalizovat, proto opačný směr.

Gradientní sestup pro minimalizaci MSE

Pro ztrátovou funkci

$$J(w_0, w_1) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \delta_{\vec{w}}(x^{(i)}) \right)^2 = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - (w_0 + w_1 x^{(i)}) \right)^2,$$

lze gradient spočítat jako

$$\begin{aligned} \frac{\partial}{\partial w_0} J(w_0, w_1) &= -\frac{2}{N} \sum_{i=1}^N \left(y^{(i)} - \delta_{\vec{w}}(x^{(i)}) \right) \\ \frac{\partial}{\partial w_1} J(w_0, w_1) &= -\frac{2}{N} \sum_{i=1}^N \left(y^{(i)} - \delta_{\vec{w}}(x^{(i)}) \right) x^{(i)} \end{aligned}$$

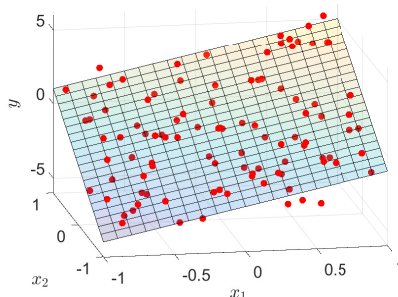
Mnoharozměrná lineární regrese

- ▶ $\vec{x}^{(i)} = (x_1^{(i)}, \dots, x_D^{(i)})^\top$, tj. příklady jsou popsány více než jednou vstupní proměnnou (jsou D -rozměrné).
- ▶ Najděte parametry $\vec{w} = (w_0, \dots, w_D)^\top$ lineárního modelu $\hat{y} = \vec{w}^\top \vec{x}$ pro danou trénovací (multi-)množinu $\mathcal{T} = \{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^N$.

Trénování: chtěli bychom,
aby pro každé (i) : $y^{(i)} = \vec{w}^\top \vec{x}^{(i)}$.

Nebo maticově: $Y = \vec{w}^\top X$

Modelem je *nadrovina* (hyperplane)
v $(D + 1)$ -rozměrném prostoru.



Jaké jsou rozměry X ?

A $(D + 1) \times (D + 1)$

B $(D + 1) \times N$

C $N \times (D + 1)$

D $N \times N$

Mnoharozměrná lineární regrese

- ▶ $\vec{x}^{(i)} = (x_1^{(i)}, \dots, x_D^{(i)})^\top$, tj. příklady jsou popsány více než jednou vstupní proměnnou (jsou D -rozměrné).
- ▶ Najděte parametry $\vec{w} = (w_0, \dots, w_D)^\top$ lineárního modelu $\hat{y} = \vec{w}^\top \vec{x}$ pro danou trénovací (multi-)množinu $\mathcal{T} = \{(\vec{x}^{(i)}, y^{(i)})\}_{i=1}^N$.

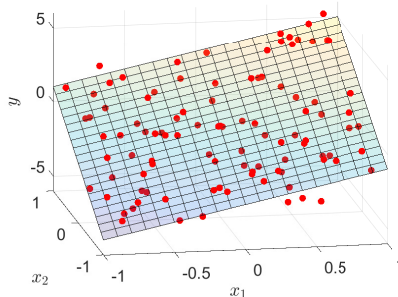
Trénování: chtěli bychom,
aby pro každé (i) : $y^{(i)} = \vec{w}^\top \vec{x}^{(i)}$.

Nebo maticově: $Y = \vec{w}^\top X$

Jaké jsou rozměry X ?

- A** $(D+1) \times (D+1)$
- B** $(D+1) \times N$
- C** $N \times (D+1)$
- D** $N \times N$

Modelem je *nadrovina* (hyperplane)
v $(D+1)$ -rozměrném prostoru.



Mnoharozměrná lineární regrese: učení

1. Numerická optimalizace kritéria $J(\vec{w}, T)$:

- ▶ Funguje stejně jako u jednoduché regrese, jen prohledává prostor s více dimenzemi.
- ▶ Někdy je třeba pro správnou funkci optimalizačního algoritmu dobře naladit jeho parametry (rychlost učení gradientního sestupu, apod.).
- ▶ Může trvat dlouho (hodně iterací), ale je použitelná i pro velká D .

2. Explicitní řešení (Normální rovnice) $Y = \vec{w}^T X$:

$$\vec{w}^* = (XX^T)^{-1}XY^T$$

- ▶ Metoda pro nalezení optimálních $\vec{w}^*$ analyticky!
- ▶ Není třeba volit parametry optimalizačního algoritmu. Žádné iterace.
- ▶ Je třeba spočítat $(XX^T)^{-1}$; časová složitost $O((D+1)^3)$. Pro velké D nepraktické.

Notes

D může být hodně velké! Viz. množství pixelů v obrázcích! My, lidé, jsme zvyklí na nízké dimenze - svět je 3D. Stroje nakládají s případy $D \leq 3$ a $D > 3$ v principu stejně.

Mnoharozměrná lineární regrese: učení

1. Numerická optimalizace kritéria $J(\vec{w}, T)$:

- ▶ Funguje stejně jako u jednoduché regrese, jen prohledává prostor s více dimenzemi.
- ▶ Někdy je třeba pro správnou funkci optimalizačního algoritmu dobře naladit jeho parametry (rychlost učení gradientního sestupu, apod.).
- ▶ Může trvat dlouho (hodně iterací), ale je použitelná i pro velká D .

2. Explicitní řešení (Normální rovnice) $Y = \vec{w}^\top X$:

$$\vec{w}^* = (XX^\top)^{-1}XY^\top$$

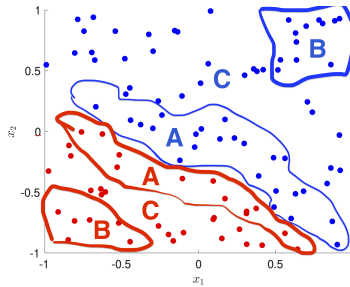
- ▶ Metoda pro nalezení optimálních $\vec{w}^*$ analyticky!
- ▶ Není třeba volit parametry optimalizačního algoritmu. Žádné iterace.
- ▶ Je třeba spočítat $(XX^\top)^{-1}$; časová složitost $O((D+1)^3)$. Pro velké D nepraktické.

Notes

D může být hodně velké! Viz. množství pixelů v obrázcích! My, lidé, jsme zvyklí na nízké dimenze - svět je 3D. Stroje nakládají s případy $D \leq 3$ a $D > 3$ v principu stejně.

- ▶ Binární klasifikace
- ▶ Diskriminační funkce
- ▶ Klasifikace jako regresní problém (lineární a logistická regrese)
- ▶ Jaká ztrátová funkce je vhodná?
- ▶ Etalonový klasifikátor (nejbližší soused a lineární klasifikátor)
- ▶ Pravdivost a preciznost (Accuracy vs precision)

Kvíz: Důležitost trénovacích příkladů



Intuitivně, které z trénovacích příkladů by měly mít největší vliv na rozhodnutí, zda nový, dosud neznámý příklad má být **červený** nebo **modrý**?

- A** Ty, které jsou nejbliž příkladům opačné barvy.
- B** Ty, které jsou nejdál od příkladů opačné barvy.
- C** Ty, které jsou uprostřed příkladů stejné barvy.
- D** Žádné. Všechny příklady jsou stejně důležité.

Úloha binární klasifikace

Mějme trénovací datovou množinu $\mathcal{T} = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(N)}, y^{(N)})\}$:

- ▶ každý příklad popsán vektorem $\vec{x} = (x_1, \dots, x_D)$,
- ▶ označen skutečnou třídou $y \in \{+1, -1\}$.

Cíl:

- ▶ Najděte klasifikátor (rozhodovací strategii/pravidlo) δ , který minimalizuje empirické riziko $R_{\text{emp}}(\delta)$.

Diskriminační funkce

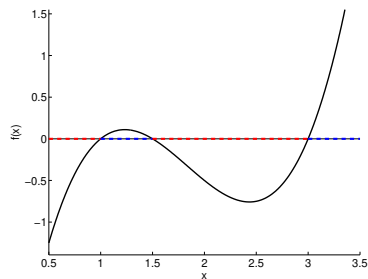
Diskriminační funkce $f(\vec{x})$:

- Každému pozorování $\vec{x}$ přiřazuje reálné číslo. Může být lineární, ale nemusí.
- Pro 2 třídy stačí jediná diskriminační funkce.
- Používá se k vytvoření rozhodovací strategie (která přiřazuje třídu každému pozorování):

$$\hat{y} = \delta(\vec{x}) = \begin{cases} +1 & \text{iff } f(\vec{x}) > 0, \text{ a} \\ -1 & \text{iff } f(\vec{x}) < 0, \end{cases}$$

tj. $\hat{y} = \delta(\vec{x}) = \text{sign}(f(\vec{x}))$.

- Rozhodovací hranice: $\{\vec{x} | f(\vec{x}) = 0\}$
- Lineární klasifikace: rozhodovací hranice musí být lineární.
- Učení pak odpovídá hledání vhodné funkce f (nebo jejích parametrů).



Notes

U lineárního klasifikátoru vyžadujeme lineární rozhodovací hranice, nikoli jeho diskriminační funkci!

Diskriminační funkce

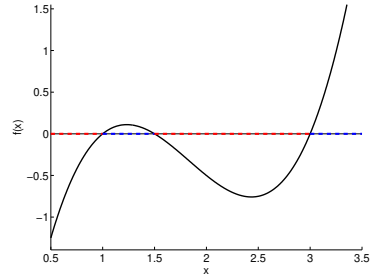
Diskriminační funkce $f(\vec{x})$:

- Každému pozorování $\vec{x}$ přiřazuje reálné číslo. Může být lineární, ale nemusí.
- Pro 2 třídy stačí jediná diskriminační funkce.
- Používá se k vytvoření rozhodovací strategie (která přiřazuje třídu každému pozorování):

$$\hat{y} = \delta(\vec{x}) = \begin{cases} +1 & \text{iff } f(\vec{x}) > 0, \text{ a} \\ -1 & \text{iff } f(\vec{x}) < 0, \end{cases}$$

tj. $\hat{y} = \delta(\vec{x}) = \text{sign}(f(\vec{x}))$.

- **Rozhodovací hranice:** $\{\vec{x} | f(\vec{x}) = 0\}$
- **Lineární klasifikace:** rozhodovací hranice musí být lineární.
- *Učení* pak odpovídá hledání vhodné funkce f (nebo jejích parametrů).



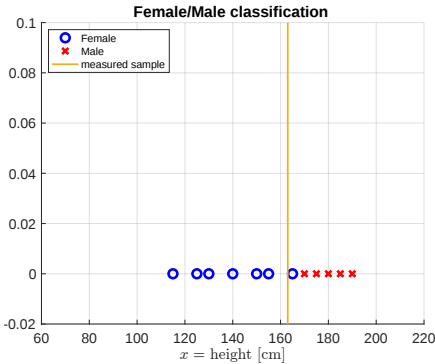
Notes

U lineárního klasifikátoru vyžadujeme lineární rozhodovací hranice, nikoli jeho diskriminační funkci!

Příklad: Klasifikace Muž/Žena na základě výšky

Trénovací datová sada $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^N, x^{(i)} \in \mathcal{X}, s^{(i)} \in \mathcal{S} = \{F, M\}$

i	1	2	3	4	5	6	7	8	9	10	11	12
Výška $x^{(i)}$	115	125	130	140	150	155	165	170	175	180	185	190
Pohlaví $s^{(i)}$	F	F	F	F	F	F	F	M	M	M	M	M
Pohlaví $y^{(i)}$ (+1/-1)	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1



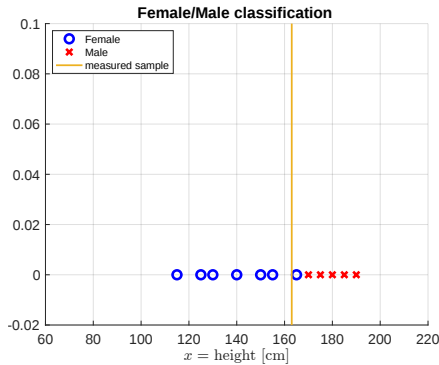
Notes

Run onedim_linclass_learning

Příklad: Klasifikace Muž/Žena na základě výšky

Trénovací datová sada $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^N, x^{(i)} \in \mathcal{X}, s^{(i)} \in \mathcal{S} = \{F, M\}$

i	1	2	3	4	5	6	7	8	9	10	11	12
Výška $x^{(i)}$	115	125	130	140	150	155	165	170	175	180	185	190
Pohlaví $s^{(i)}$	F	F	F	F	F	F	F	M	M	M	M	M
Pohlaví $y^{(i)}$ (+1/-1)	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1



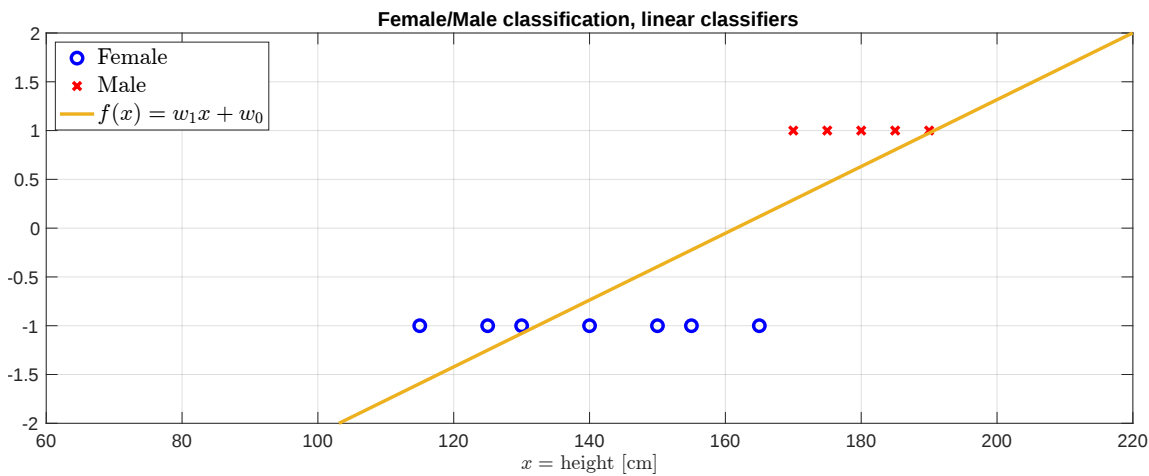
Nové pozorování ke klasifikaci: $x^Q = 163$

Do jaké třídy patří x^Q ? $\delta(x^Q) = ?$

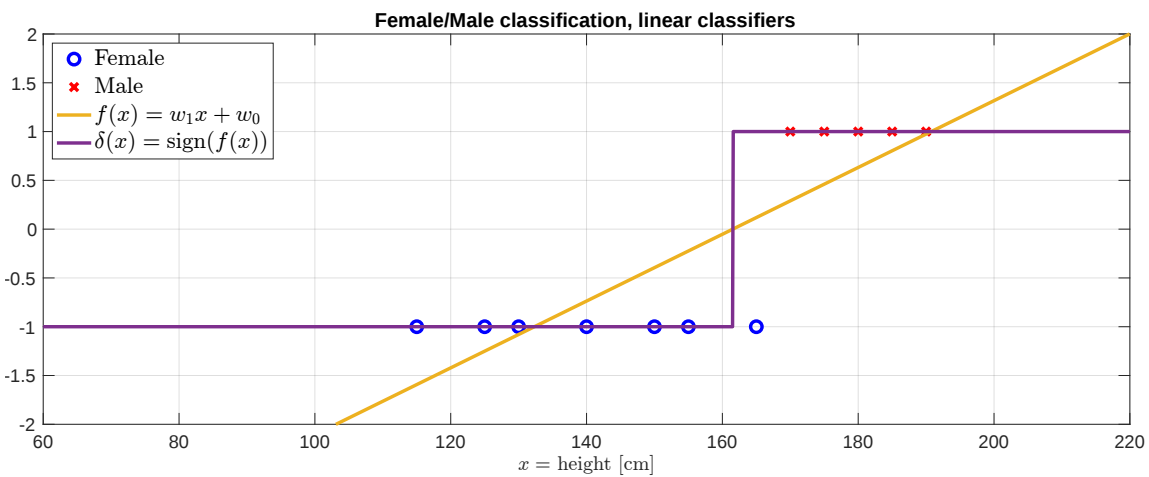
Notes

Run onedim_linclass_learning

Příklad: Lineární diskř. funkce nalezená metodou nejmenších čtverců

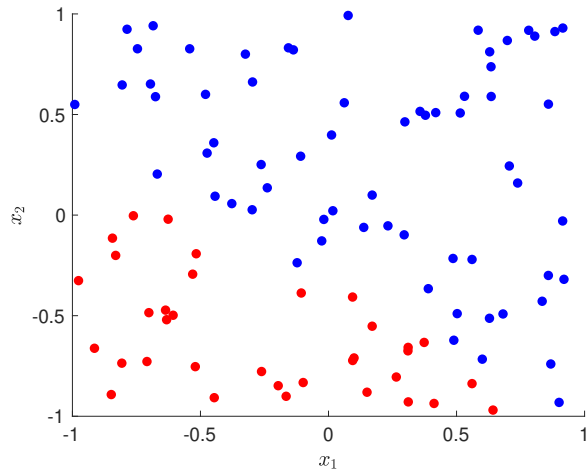


Příklad: Odpovídající rozhodovací strategie



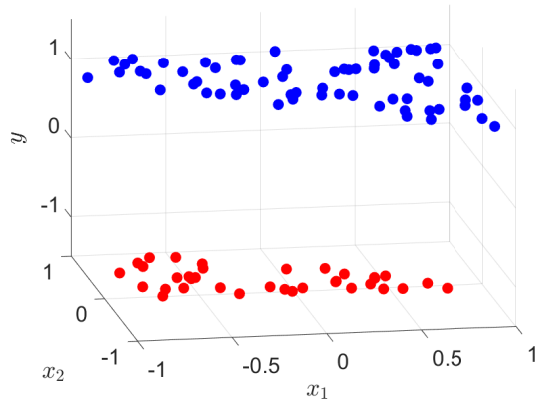
Notes

Učení lineárního klasifikátoru: naivní přístup



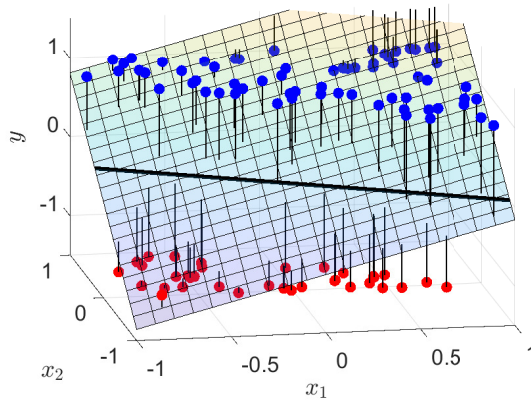
Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné třídy $s^{(i)}$.

Učení lineárního klasifikátoru: naivní přístup



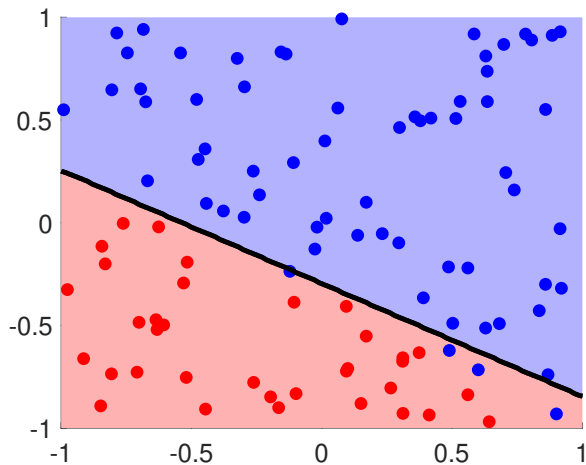
Vyjádřeme třídy hodnotami proměnné y , tj. $y^{(i)} = -1$ nebo $y^{(i)} = 1$.

Učení lineárního klasifikátoru: naivní přístup



Proložme těmito daty lineární diskř. funkci minimalizující MSE jako při regresi.
Vrstevnice $y = 0 \dots$

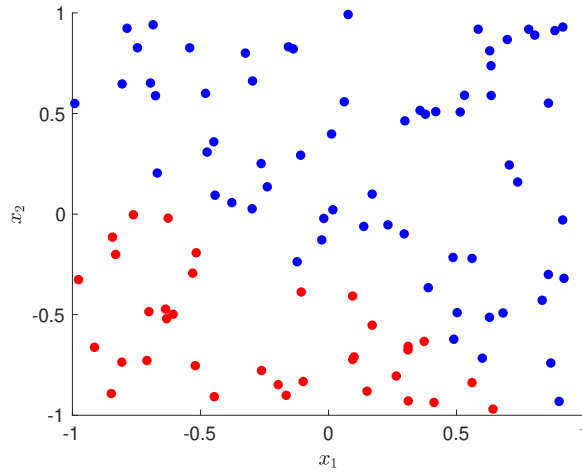
Učení lineárního klasifikátoru: naivní přístup



... pak tvoří lineární rozhodovací hranici v původním 2D prostoru.
Ale je takový klasifikátor obecně dobrý?

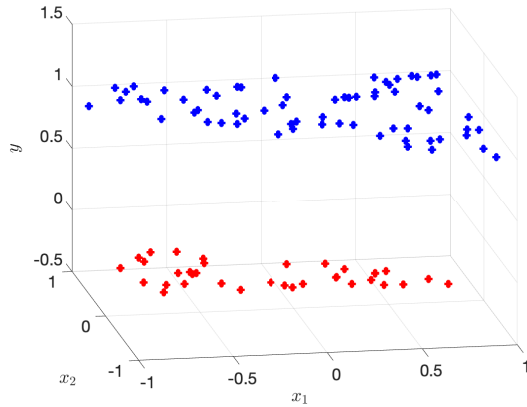
Lze udělat něco lepšího, než prokládat lineární funkcí?

Prokládání vhodnější funkce: Logistická regrese



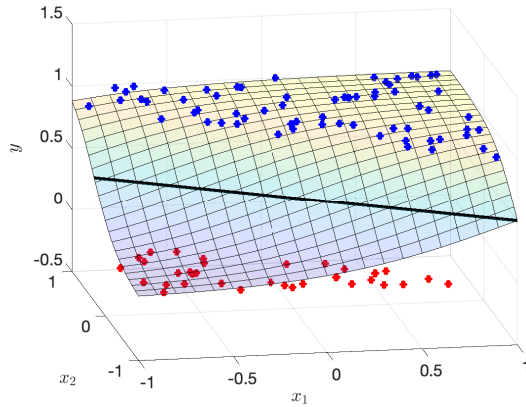
Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné třídy $s^{(i)}$.

Prokládání vhodnější funkce: Logistická regrese



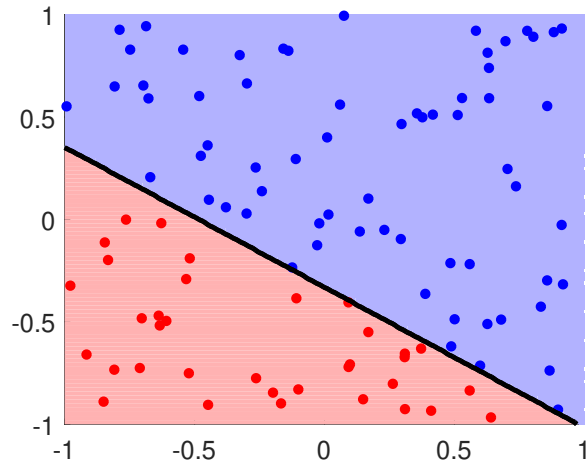
Vyjádříme třídy hodnotami proměnné y , tj. $y^{(i)} = 0$ nebo $y^{(i)} = 1$.

Prokládání vhodnější funkce: Logistická regrese



Proložme těmito daty **sigmoidální** diskř. funkci minimalizující MSE jako při regresi.
Vrstevnice $y = 0.5 \dots$

Prokládání vhodnější funkce: Logistická regrese



...pak tvoří lineární rozhodovací hranici v původním 2D prostoru.

Model logistické regrese

Logistická regrese používá diskriminační funkci, která je nelineární transformací hodnot lineární funkce

$$f_{\vec{w}}(\vec{x}) = g(\vec{w}^{\top} \vec{x}) = \frac{1}{1 + e^{-\vec{w}^{\top} \vec{x}}},$$

kde $g(z) = \frac{1}{1 + e^{-z}}$ je **sigmoidální** funkce (též sigmoida nebo **logistická** funkce).

Interpretace modelu:

- ▶ $f_{\vec{w}}(\vec{x})$ se často interpretuje jako odhad pravděpodobnosti, že $\vec{x}$ patří do třídy 1.
- ▶ Rozhodovací hranice je definována jako vrstevnice $\{\vec{x} : f_{\vec{w}}(\vec{x}) = 0.5\}$.
- ▶ Logistická regrese je *klasifikační* model!
- ▶ Diskriminační funkce $f_{\vec{w}}(\vec{x})$ samotná není lineární; ale *rozhodovací hranice je stále lineární!*
- ▶ Díky sigmoidální transformaci logistická regrese téměř není ovlivněna příklady, které leží daleko od rozhodovací hranice!

Notes

Try to draw the course of the function by hand.

Model logistické regrese

Logistická regrese používá diskriminační funkci, která je nelineární transformací hodnot lineární funkce

$$f_{\vec{w}}(\vec{x}) = g(\vec{w}^\top \vec{x}) = \frac{1}{1 + e^{-\vec{w}^\top \vec{x}}},$$

kde $g(z) = \frac{1}{1 + e^{-z}}$ je **sigmoidální** funkce (též sigmoida nebo **logistická** funkce).

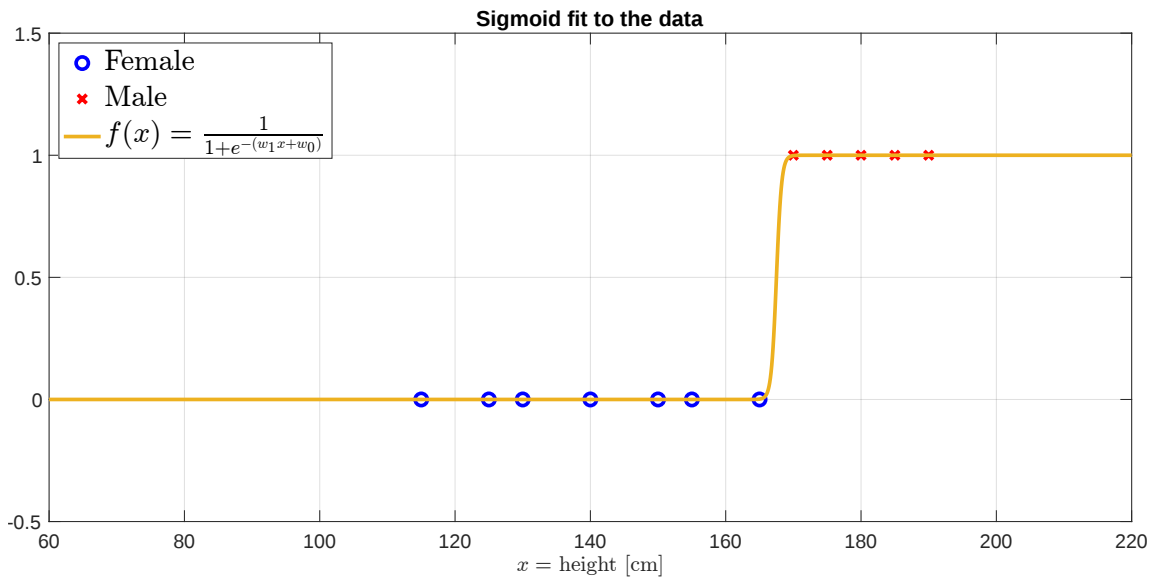
Interpretace modelu:

- ▶ $f_{\vec{w}}(\vec{x})$ se často interpretuje jako odhad pravděpodobnosti, že $\vec{x}$ patří do třídy 1.
- ▶ **Rozhodovací hranice** je definována jako vrstevnice $\{\vec{x} : f_{\vec{w}}(\vec{x}) = 0.5\}$.
- ▶ Logistická *regrese* je *klasifikační* model!
- ▶ Diskriminační funkce $f_{\vec{w}}(\vec{x})$ samotná není lineární; ale *rozhodovací hranice je stále lineární!*
- ▶ Díky sigmoidální transformaci logistická regrese téměř není ovlivněna příklady, které leží daleko od rozhodovací hranice!

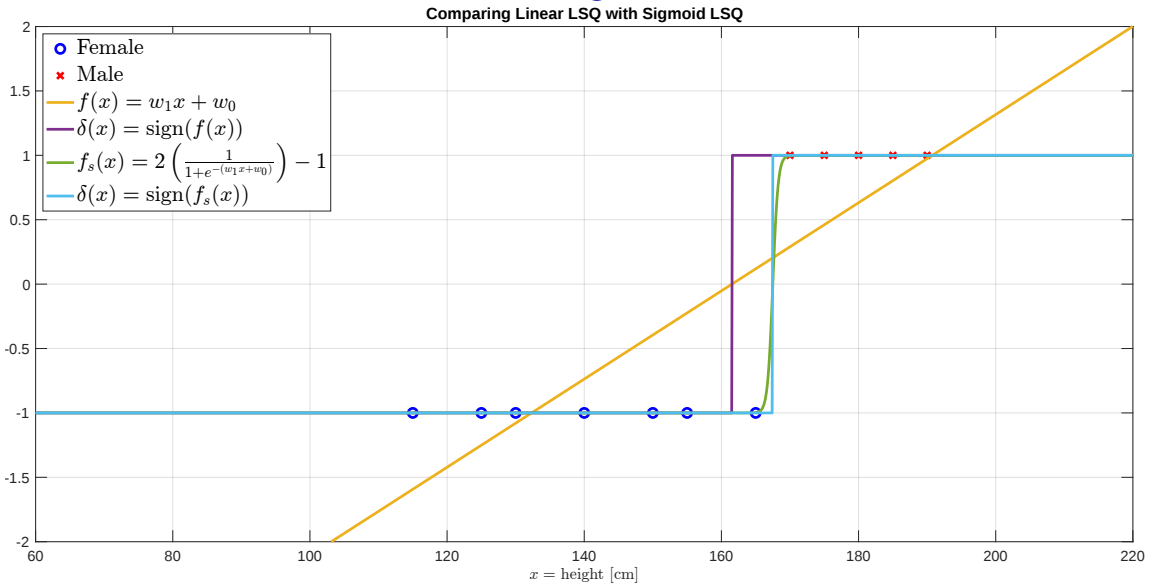
Notes

Try to draw the course of the function by hand.

Sigmoida proložená metodou nejmenších čtverců



Proložení lineární funkcí vs proložení sigmoidou



Jaká ztrátová funkce ℓ je vhodná?

Model logistické regrese lze natrénovat minimalizací střední kv. chyby J_{MSE} :

- nekonvexní, multimodální funkce, kterou není snadné optimalizovat.

Logistická regrese používá tzv. křížovou entropii (cross-entropy) :

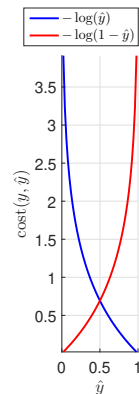
$$J(\vec{w}, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, f_{\vec{w}}(\vec{x}^{(i)})), \text{ kde}$$

$$\ell(y, \hat{y}) = \begin{cases} -\log(\hat{y}) & \text{if } y = 1 \\ -\log(1 - \hat{y}) & \text{if } y = 0 \end{cases}$$

což se dá přepsat do jediného výrazu jako

$$\ell(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}).$$

- Snáze se optimalizuje pomocí numerických solverů.



Jaká ztrátová funkce ℓ je vhodná?

Model logistické regrese lze natrénovat minimalizací střední kv. chyby J_{MSE} :

- nekonvexní, multimodální funkce, kterou není snadné optimalizovat.

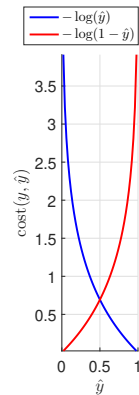
Logistická regrese používá tzv. **křížovou entropii (cross-entropy)** :

$$J(\vec{w}, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, f_{\vec{w}}(\vec{x}^{(i)})), \text{ kde}$$

$$\ell(y, \hat{y}) = \begin{cases} -\log(\hat{y}) & \text{if } y = 1 \\ -\log(1 - \hat{y}) & \text{if } y = 0 \end{cases},$$

což se dá přepsat do jediného výrazu jako

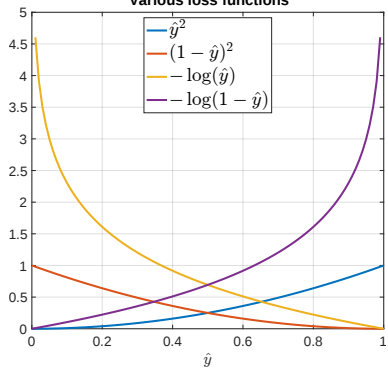
$$\ell(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}).$$



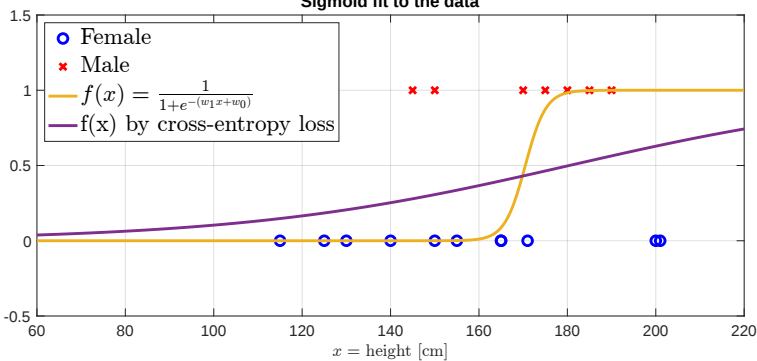
- Snáze se optimalizuje pomocí numerických solverů.

Střední kv. chyba vs. křížová entropie

Various loss functions



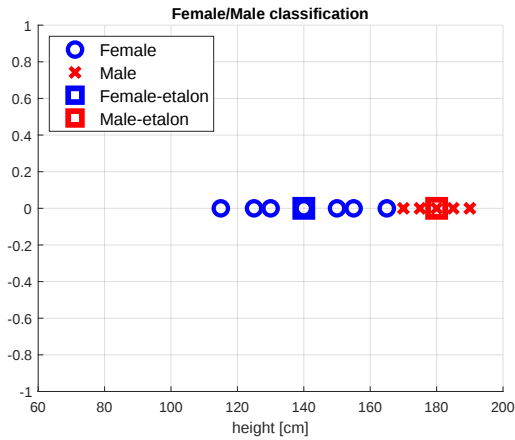
Sigmoid fit to the data



Sigmoidální $f(x)$ interpretujeme jako $P(s = \text{Male} \mid x)$: přímé učení **diskriminativního modelu**.
V porovnání s MSE křížová entropie silně penalizuje velké chyby!

Alternativní nápad: etalony

Reprezentujme každou třídu jediným příkladem, tzv. **etalonem** ! (Nebo několika etalony.)



$$e_F = \text{ave}(\{x^{(i)} : s^{(i)} = F\}) = 140$$
$$e_M = \text{ave}(\{x^{(i)} : s^{(i)} = M\}) = 180$$

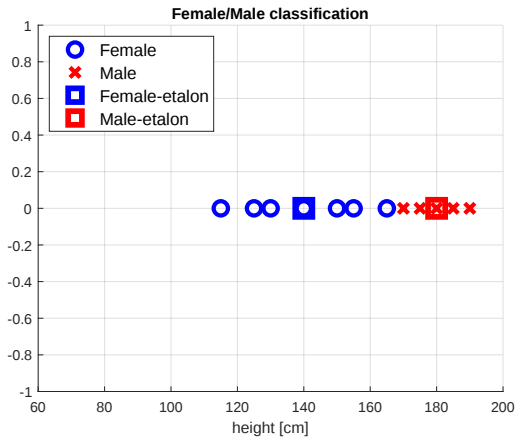
$x^Q = 163$
Na základě etalonů: $d^Q = \delta(x^Q) = ?$

Notes

Based on etalons: $d^Q = M$

Alternativní nápad: etalony

Reprezentujme každou třídu jediným příkladem, tzv. **etalonem** ! (Nebo několika etalony.)



$$e_F = \text{ave}(\{x^{(i)} : s^{(i)} = F\}) = 140$$
$$e_M = \text{ave}(\{x^{(i)} : s^{(i)} = M\}) = 180$$

$$x^Q = 163$$

Na základě etalonů: $d^Q = \delta(x^Q) = ?$

A $d^Q = F$

B $d^Q = M$

C Obě třídy stejně pravděpodobné

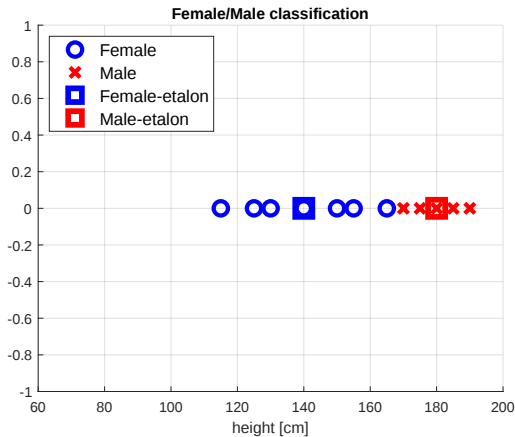
D Neumím rozhodnout

Notes

Based on etalons: $d^Q = M$

Alternativní nápad: etalony

Reprezentujme každou třídu jediným příkladem, tzv. **etalonem** ! (Nebo několika etalony.)



$$e_F = \text{ave}(\{x^{(i)} : s^{(i)} = F\}) = 140$$
$$e_M = \text{ave}(\{x^{(i)} : s^{(i)} = M\}) = 180$$

$$x^Q = 163$$

Na základě etalonů: $d^Q = \delta(x^Q) = ?$

Klasifikuj jako $d^Q = \operatorname{argmin}_{s \in \mathcal{S}} \operatorname{dist}(x^Q, e_s)$

Jakého typu je funkce $\operatorname{dist}(x^Q, e_s)$?

Notes

Based on etalons: $d^Q = M$

Etalonový klasifikátor je lineární klasifikátor!

Pokud $\text{dist}(x, e) = (x - e)^2$, pak

$$\begin{aligned}\operatorname{argmin}_{s \in S} \text{dist}(x, e_s) &= \operatorname{argmin}_{s \in S} (x - e_s)^2 = \operatorname{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) = \\ &= \operatorname{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2}e_s^2}_{\text{lineární v } x})\end{aligned}$$

Klasifikace do více tříd: každá třída s má lineární diskriminační funkci $f_s(x) = a_s x + b_s$ a

$$\delta(x) = \operatorname{argmax}_{s \in S} f_s(x)$$

Binární klasifikace: stačí jediná lineární diskriminační funkce $g(x)$ a

$$\delta(x) = \begin{cases} s_1 & \text{pokud } g(x) \geq 0, \\ s_2 & \text{pokud } g(x) < 0. \end{cases}$$

Etalonový klasifikátor je lineární klasifikátor!

Pokud $\text{dist}(x, e) = (x - e)^2$, pak

$$\begin{aligned}\operatorname{argmin}_{s \in S} \text{dist}(x, e_s) &= \operatorname{argmin}_{s \in S} (x - e_s)^2 = \operatorname{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) = \\ &= \operatorname{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2}e_s^2}_{\text{lineární v } x})\end{aligned}$$

Klasifikace do více tříd: každá třída s má lineární diskriminační funkci $f_s(x) = a_s x + b_s$ a

$$\delta(x) = \operatorname{argmax}_{s \in S} f_s(x)$$

Binární klasifikace: stačí jediná lineární diskriminační funkce $g(x)$ a

$$\delta(x) = \begin{cases} s_1 & \text{pokud } g(x) \geq 0, \\ s_2 & \text{pokud } g(x) < 0. \end{cases}$$

Etalonový klasifikátor je lineární klasifikátor!

Pokud $\text{dist}(x, e) = (x - e)^2$, pak

$$\begin{aligned}\operatorname{argmin}_{s \in S} \text{dist}(x, e_s) &= \operatorname{argmin}_{s \in S} (x - e_s)^2 = \operatorname{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) = \\ &= \operatorname{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2}e_s^2}_{\text{lineární v } x})\end{aligned}$$

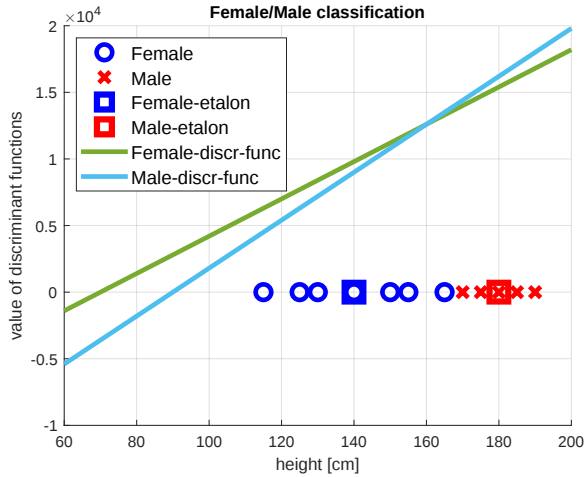
Klasifikace do více tříd: každá třída s má lineární diskriminační funkci $f_s(x) = a_s x + b_s$ a

$$\delta(x) = \operatorname{argmax}_{s \in S} f_s(x)$$

Binární klasifikace: stačí jediná lineární diskriminační funkce $g(x)$ a

$$\delta(x) = \begin{cases} s_1 & \text{pokud } g(x) \geq 0, \\ s_2 & \text{pokud } g(x) < 0. \end{cases}$$

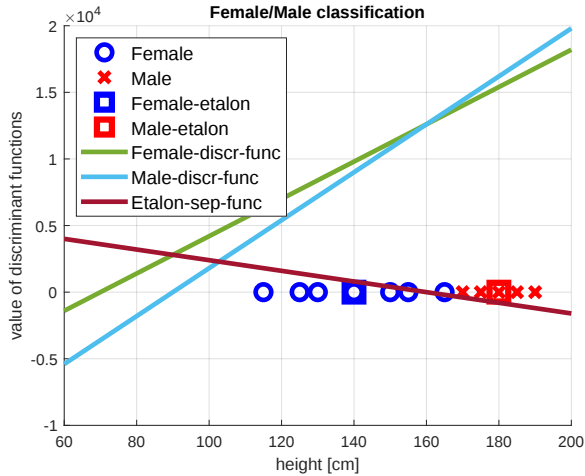
Příklad: F/M – Lineární diskriminační funkce založené na etalonech



Diskriminační funkce pro 2 třídy:

$$\begin{aligned}f_F(x) &= a_F x + b_F = \\&= e_F x - \frac{1}{2} e_F^2 = 140x - 9800 \\f_M(x) &= a_M x + b_M = \\&= e_M x - \frac{1}{2} e_M^2 = 180x - 16200\end{aligned}$$

Příklad: F/M – Lineární diskriminační funkce založené na etalonech



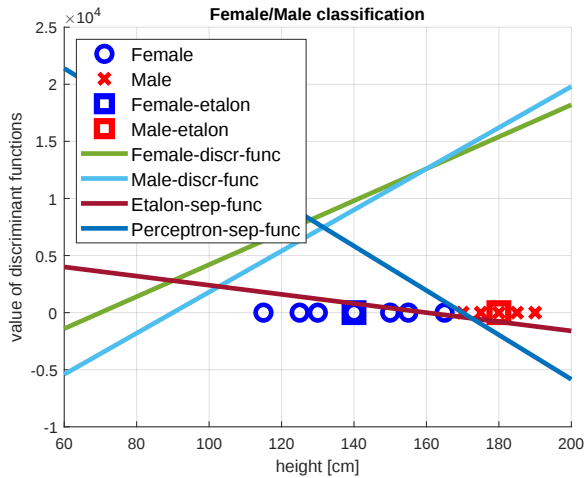
Diskriminační funkce pro 2 třídy:

$$\begin{aligned}f_F(x) &= a_F x + b_F = \\&= e_F x - \frac{1}{2} e_F^2 = 140x - 9800 \\f_M(x) &= a_M x + b_M = \\&= e_M x - \frac{1}{2} e_M^2 = 180x - 16200\end{aligned}$$

Jediná diskř. funkce oddělující obě třídy:

$$\begin{aligned}g(x) &= f_F(x) - f_M(x) = \\&= -40x + 6400\end{aligned}$$

Příklad: F/M – Umíme najít lepší etalony?

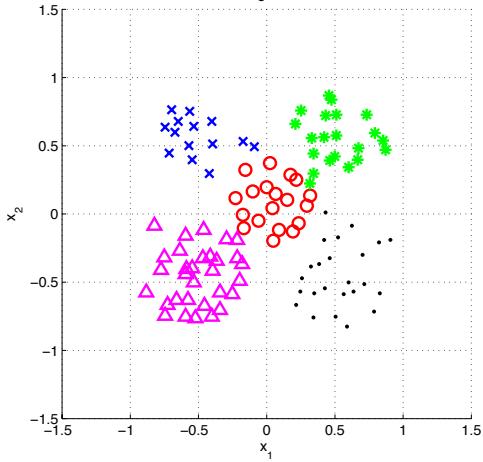


Lineární klasifikátory založené na průměrných etalonech dělají chyby.

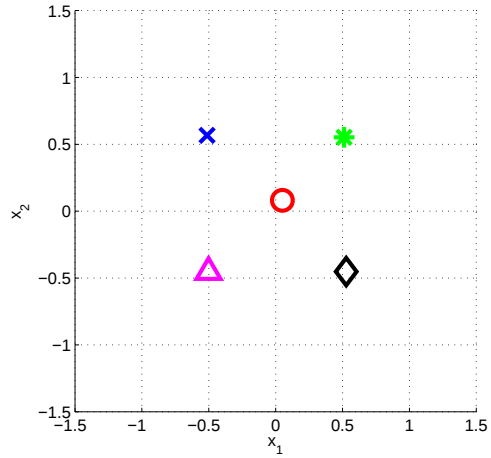
Perceptronový algoritmus (např.) umí najít bezchybný klasifikátor (pokud existuje).

Etalony ve vícerozměrných prostorech

Pentagon data



minimum distance from etalons



Z dat $\mathcal{T} = \{(\vec{x}^{(i)}, s^{(i)})\}$, extrahuj jeden etalon $\vec{e}_s$ pro každou třídu $s \in \mathcal{S}$.

Etalony ve vícerozměrných prostorech (pokr.)

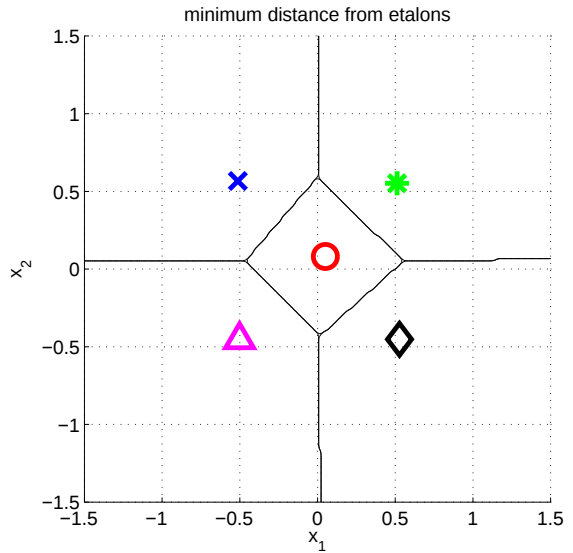
Extrahuj etalon pro každou třídu s :

$$\vec{e}_s = \text{ave}(\{\vec{x}^{(i)} : s^{(i)} = s\})$$

Rozhodovací strategie

$$\delta(\vec{x}) = \underset{s \in S}{\operatorname{argmin}} \|\vec{x} - \vec{e}_s\|^2$$

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



Etalony ve vícerozměrných prostorech (pokr.)

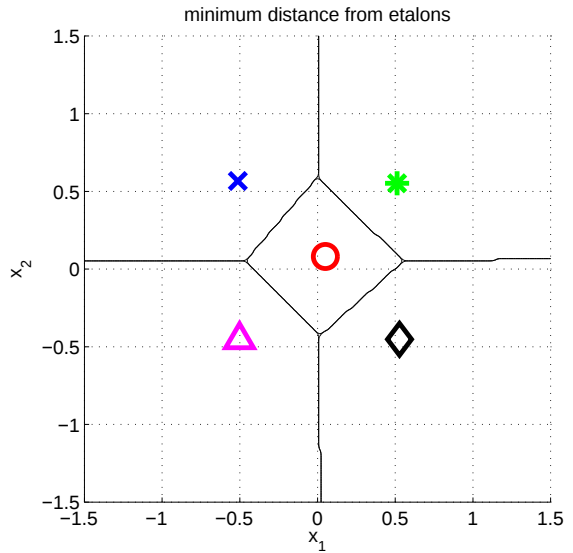
Extrahuj etalon pro každou třídu s :

$$\vec{e}_s = \text{ave}(\{\vec{x}^{(i)} : s^{(i)} = s\})$$

Rozhodovací strategie

$$\delta(\vec{x}) = \underset{s \in S}{\operatorname{argmin}} \|\vec{x} - \vec{e}_s\|^2$$

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



Etalony ve vícerozměrných prostorech (pokr.)

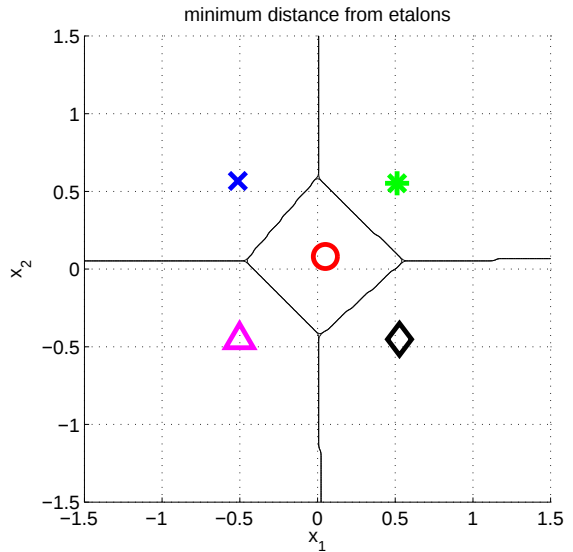
Extrahuj etalon pro každou třídu s :

$$\vec{e}_s = \text{ave}(\{\vec{x}^{(i)} : s^{(i)} = s\})$$

Rozhodovací strategie

$$\delta(\vec{x}) = \underset{s \in S}{\operatorname{argmin}} \|\vec{x} - \vec{e}_s\|^2$$

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



$$\begin{aligned}\delta(\vec{x}) &= \operatorname{argmin}_{s \in S} \|\vec{x} - \vec{e}_s\|^2 = \operatorname{argmin}_{s \in S} (\vec{x}^\top \vec{x} - 2 \vec{e}_s^\top \vec{x} + \vec{e}_s^\top \vec{e}_s) = \\ &= \operatorname{argmin}_{s \in S} \left(\vec{x}^\top \vec{x} - 2 (\vec{e}_s^\top \vec{x} - \frac{1}{2} (\vec{e}_s^\top \vec{e}_s)) \right) = \\ &= \operatorname{argmax}_{s \in S} \left(\vec{e}_s^\top \vec{x} - \frac{1}{2} (\vec{e}_s^\top \vec{e}_s) \right) = \\ &= \operatorname{argmax}_{s \in S} (\vec{w}_s^\top \vec{x} + w_{s0}) = \operatorname{argmax}_{s \in S} g_s(\vec{x}).\end{aligned}$$

Lineární funkce (plus abs. člen)

$$g_s(\vec{x}) = \vec{w}_s^\top \vec{x} + w_{s0}, \quad \text{kde} \quad \vec{w}_s = \vec{e}_s \quad \text{a} \quad w_{s0} = -\frac{1}{2} \vec{e}_s^\top \vec{e}_s.$$

Notes

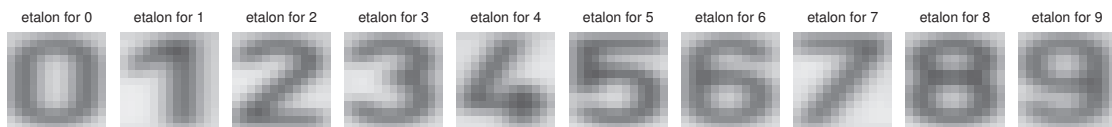
The result is a *linear discriminant function* – hence etalon classifier is a linear classifier.

We classify into the class with highest value of the discriminant function.

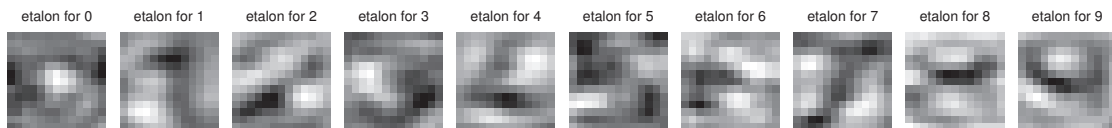
$\vec{w}_s$ is a generalized etalon. How do we find it? Such that it is better than just the mean of the class members in the training set.

Etalony v rozpoznávání číslic

Etalony založené na průměru:



Etalony odvozené z diskř. funkcí nalezených perceptronem:



Obrázky z [7].

Notes

Pamatujte, použití průměru pro konstrukci etalonu je jen heuristika. V obecném případě takový model neminimalizuje žádnou ztrátovou funkci.

Bayesovská klasifikace vs diskriminační funkce

Rozhodnutí založené na diskriminační funkci:

$$\delta(\vec{x}) = \operatorname{argmax}_{s \in \mathcal{S}} f(\vec{x}, s)$$

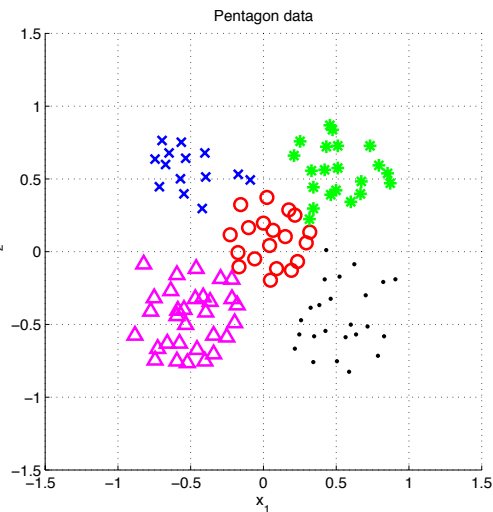
Rozhodnutí založené na posteriorní psti:

$$\delta(\vec{x}) = \operatorname{argmax}_{s \in \mathcal{S}} P(s|\vec{x}) = \operatorname{argmax}_{s \in \mathcal{S}} \frac{P(\vec{x} | s)P(s)}{P(\vec{x})}$$

Pokud zvolíme

$$f(\vec{x}, s) = P(\vec{x} | s)P(s),$$

obě metody splývají.



40 / 45

Notes

Normal distribution for general dimensionality D:

$$\mathcal{N}(\vec{x}|\vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu})^\top \Sigma^{-1}(\vec{x} - \vec{\mu})\right\}$$

Discriminant function:

$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} f(\vec{x}, s) = \operatorname{argmax}_{s \in \mathcal{S}} P(s) \mathcal{N}(\vec{x}|\vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu})^\top \Sigma^{-1}(\vec{x} - \vec{\mu})\right\}$$

How about learning $f(\vec{x}, s)$ directly without explicit modeling of underlying probabilities?

What about $f(\vec{x}, s) = \vec{w}_s^\top \vec{x} + w_{s0}$

Učení a rozhodování

Fáze **učení** : určení modulu/funkce/parametrů na základě dat.

Fáze **rozhodování** : rozhodnutí o neznámém pozorování $\vec{x}$.

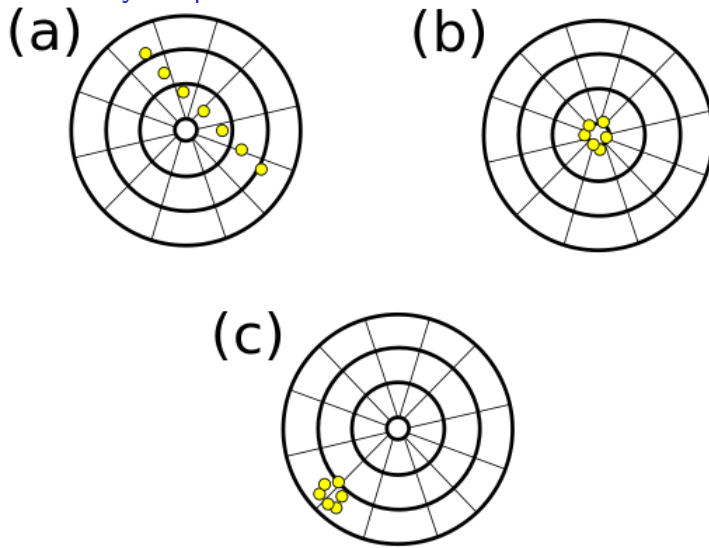
Co se učit?

- ▶ **Generativní model** : Naučit se $P(\vec{x}, s)$. Rozhodovat podle $\operatorname{argmax}_s P(s|\vec{x})$.
- ▶ **Discriminativní model** : Naučit se přímo $P(s|\vec{x})$ a rozhodovat dle něj.
- ▶ **Diskriminační funkce** : Naučit se $f_s(\vec{x})$ a rozhodovat podle $\operatorname{argmax}_s f_s(\vec{x})$.

Notes

Generative models because by sampling from them it is possible to generate synthetic data points $\vec{x}$.

Accuracy vs precision



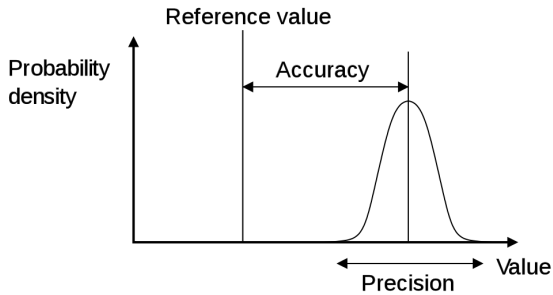
https://commons.wikimedia.org/wiki/File:Precision_versus_accuracy.svg

42 / 45

Notes

Accuracy: how close (is your model) to the truth. Precision: how consistent/stable your model is.

Přesnost, pravdivost, preciznost



- ▶ **Pravdivost (dříve správnost)** : blízkost průměru ke správné hodnotě (systematická chyba, zkreslení)
- ▶ **Preciznost (dříve shodnost)** : těsnost shody mezi výsledky měření (rozptyl, opakovatelnost, reprodukovatelnost)
- ▶ **Přesnost** : zahrnuje pravdivost i preciznost

https://cs.wikipedia.org/wiki/P%C5%99esnost_a_preciznost

<https://www.technicke-normy-csn-cz/csn-iso-5725-1-010251-158147.html>

43 / 45

Notes

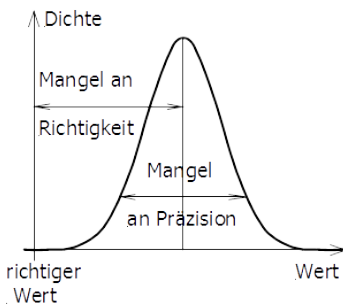
In German:

- Accuracy: Richtigkeit
- Precision: Präzision
- Both together: Genauigkeit

In Czech:

- Accuracy: Pravdivost (dříve také správnost).
- Precision: Preciznost (dříve také shodnost).
- Both together: Přesnost.

Think about terms *bias* and *error*. I



Další čtení: Kapitola 18 z [6], nebo kapitola 4 z [1], nebo kapitola 5 z [2]. Mnoho obrázků vytvořeno pomocí [3]. Můžete si také spustit demo funkce z [7]. Jak lidé rozhodují a předpovídají za nejistoty, [4] (in Czech [5])

[1] Christopher M. Bishop.

Pattern Recognition and Machine Learning.

Springer Science+Business Media, New York, NY, 2006.

[https://www.microsoft.com/en-us/research/uploads/prod/2006/01/](https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf)

[Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf](https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf).

[2] Richard O. Duda, Peter E. Hart, and David G. Stork.

Pattern Classification.

John Wiley & Sons, 2nd edition, 2001.

[3] Vojtěch Franc and Václav Hlaváč.

Statistical pattern recognition toolbox.

<http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html>.

- [4] D. Kahneman, O. Sibony, and C.R. Sunstein.
Noise: A Flaw in Human Judgment.
Little Brown Spark, 2021.
- [5] D. Kahneman, O. Sibony, and C.R. Sunstein.
Šum, O chybách v lidském úsudku.
Jan Melvil Publishing, 2021.
- [6] Stuart Russell and Peter Norvig.
Artificial Intelligence: A Modern Approach.
Prentice Hall, 3rd edition, 2010.
<http://aima.cs.berkeley.edu/>.
- [7] Tomáš Svoboda, Jan Kybic, and Hlaváč Václav.
Image Processing, Analysis and Machine Vision — A MATLAB Companion.
Thomson, Toronto, Canada, 1st edition, September 2007.
<http://visionbook.felk.cvut.cz/>.