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

18. května 2024

Obsah

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.

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.

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.

Učení: minimalizace empirického rizika

- Mějme množinu parametrizovaných strategií $\delta \colon \mathcal{X} \to \mathcal{D}$ a ztrátovou funkci $\ell \colon \mathcal{S} \times \mathcal{D} \to \mathbb{R}$. 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 $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^{N}$, $x \in \mathcal{X}$, $s \in \mathcal{S}$:

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

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

Učení: minimalizace empirického rizika

- Mějme množinu parametrizovaných strategií $\delta\colon \mathcal{X} \to \mathcal{D}$ a ztrátovou funkci $\ell\colon \mathcal{S} \times \mathcal{D} \to \mathbb{R}$. 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 $\mathcal{T} = \{(x^{(i)}, s^{(i)})\}_{i=1}^{N}$, $x \in \mathcal{X}$, $s \in \mathcal{S}$:

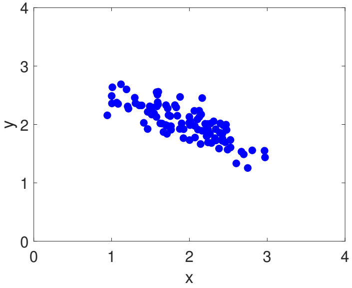
$$R_{\mathrm{emp}}(\delta) = \frac{1}{N} \sum_{(x^{(i)}, s^{(i)}) \in \mathcal{T}} \ell(s^{(i)}, \delta(x^{(i)})).$$

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

Obsah

Kvíz: Proložení bodů přímkou

Chceme přímkou ve tvaru $\widehat{y} = w_0 + w_1 x$ proložit následující data:



Nejlépe bude datům odpovídat přímka s parametry

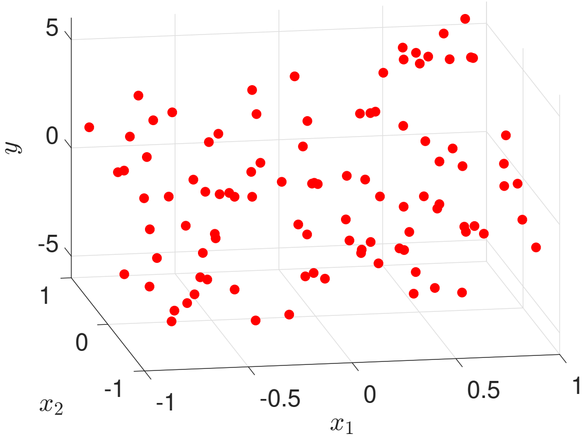
A $w_0 = -1,\ w_1 = -2$

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

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

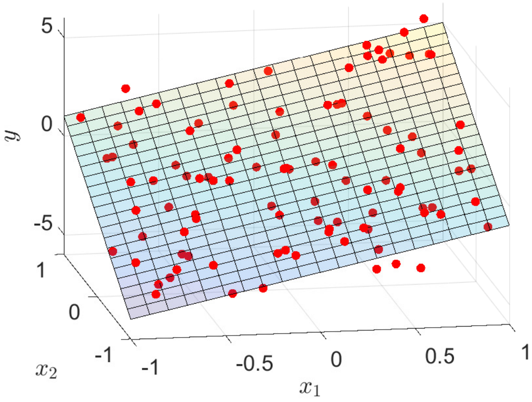
D $w_0 = 2,\ w_1 = \frac{1}{3}$

Lineární regrese: ilustrace



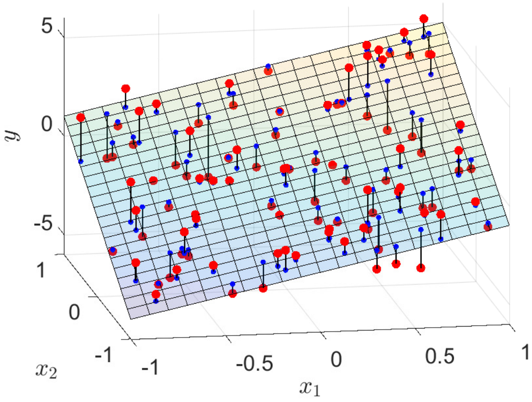
Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné hodnoty výstupní proměnné $y^{(i)}$.

Lineární regrese: ilustrace



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

Lineární regrese: ilustrace



. . . který minimalizuje odchylky odhadů od správných hodnot.

Regrese

Lineární algebra přeformulovaná do jazyka strojového učení.

Regresní úloha je úloha učení s učitelem, tj.

- máme trénovací (multi-)množinu $\mathcal{T} = \{(\vec{x}^{(1)}, y^{(1)}), \ldots, (\vec{x}^{(N)}, y^{(N)})\}$, 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) $\vec{x} = (x_1, \ldots, x_D)$ a závislou proměnnou (výstupem) y.

Lineární regrese

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

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

kde

- $\widehat{y}$ je *predikce* modelu (*odhad* skutečné hodnoty y),
- $\delta(\vec{x})$ je rozhodovací strategie (v tomto případě lineární model),
- $w_0, \ldots, w_D$ 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 $\vec{x} = (1, x_1, \ldots, x_D)$, a
- pokud zahrneme absolutní člen w_0 do vektoru $\vec{w}$ tak, že $\vec{w} = (w_0, w_1, \ldots, w_D)$, pak

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

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

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

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

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

Poznámky k notaci

Homogenní souřadnice :

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

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

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

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

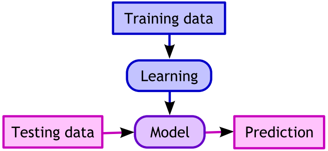
pak můžeme zapsat dávkový výpočet predikcí pro všechna pozorování v X jako

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

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

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

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

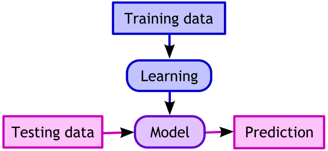
$$\vec{w}^* = \operatorname*{argmin}_{\vec{w}} R_{\mathrm{emp}}(\delta_{\vec{w}}) = \operatorname*{argmin}_{\vec{w}} J(\vec{w}, \mathcal{T}),$$

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

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:

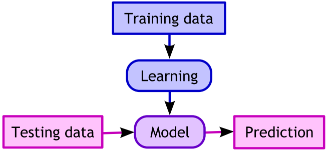
$$\vec{w}^* = \underset{\vec{w}}{\operatorname{argmin}}\, R_{\text{emp}}(\delta_{\vec{w}}) = \underset{\vec{w}}{\operatorname{argmin}}\, J(\vec{w}, \mathcal{T}),$$

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?

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:

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

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

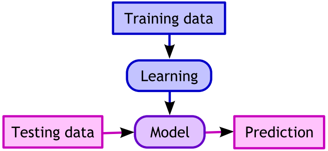
$$\vec{w}^* = \operatorname*{argmin}_{\vec{w}} R_{\mathrm{emp}}(\delta_{\vec{w}}) = \operatorname*{argmin}_{\vec{w}} J(\vec{w}, \mathcal{T}),$$

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

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:

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

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

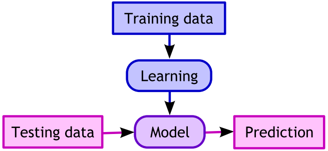
$$\vec{w}^* = \operatorname*{argmin}_{\vec{w}} R_{\mathrm{emp}}(\delta_{\vec{w}}) = \operatorname*{argmin}_{\vec{w}} J(\vec{w}, \mathcal{T}),$$

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?

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:

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

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?

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 z 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 přímku, která minimizuje velikost chyb $y - \hat{y}$:

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

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 z 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 přímku, která minimizuje velikost chyb $y - \widehat{y}$:

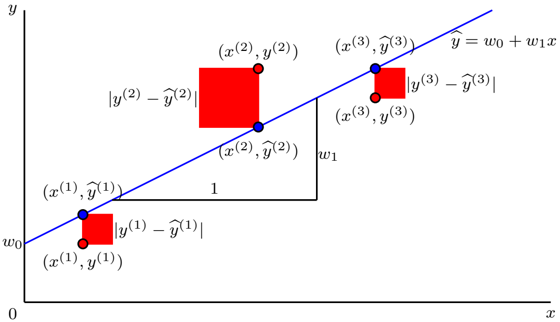
$$\vec{w}^* = (w_0^*, w_1^*) = \operatorname*{argmin}_{w_0, w_1} R_{\text{emp}}(w_0, w_1) = \operatorname*{argmin}_{w_0, w_1} J(w_0, w_1, \mathcal{T}).$$

Metoda nejmenších čtverců

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

$$J_{MSE}(\vec{w}) = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \widehat{y}^{(i)}\right)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \delta_{\vec{w}}(\vec{x}^{(i)})\right)^2 .$$



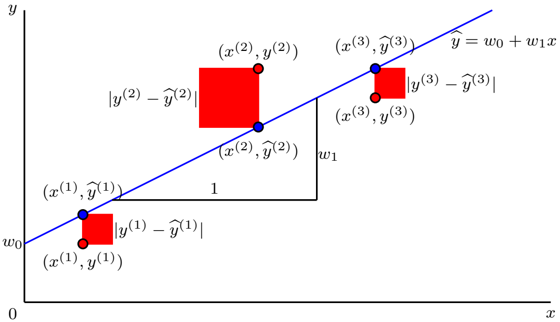
Existuje analytické řešení? Explicitní řešení:

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

Metoda nejmenších čtverců

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

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

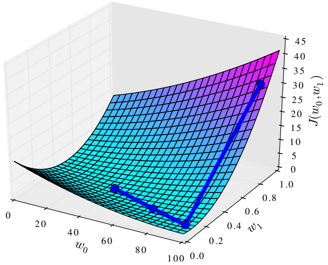


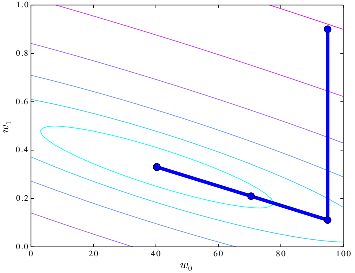
Existuje analytické řešení? Explicitní řešení:

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

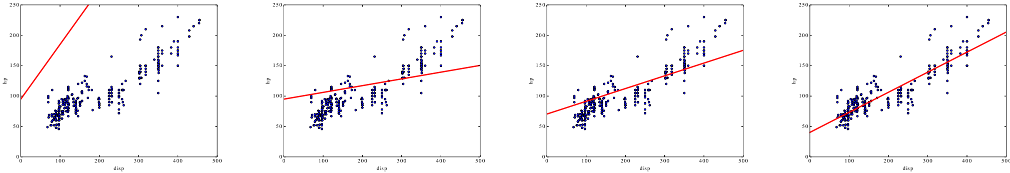
Univerzální metoda: minimalizace kriteriální funkce J

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





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



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).

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

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

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

Mnoharozměrná lineární regrese

- $\vec{x}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})^\top$, tj. příklady jsou popsány více než jednou vstupní proměnnou (jsou D-rozměrné).
- Najděte parametry $\vec{w} = (w_0, \ldots, w_D)^\top$ 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: $\mathrm{Y} = \vec{w}^\top \mathrm{X}$

Jaké jsou rozměry X?

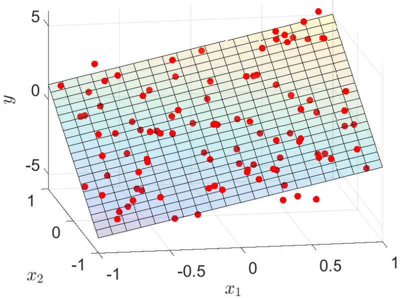
A $(D+1) \times (D+1)$

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

- $\vec{x}^{(i)} = (x_1^{(i)}, \ldots, x_D^{(i)})^\top$, tj. příklady jsou popsány více než jednou vstupní proměnnou (jsou D-rozměrné).
- Najděte parametry $\vec{w} = (w_0, \ldots, w_D)^\top$ lineárního modelu $\widehat{y} = \vec{w}^\top \vec{x}$ 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: $\mathrm{Y} = \vec{w}^\top \mathrm{X}$

Jaké jsou rozměry X?

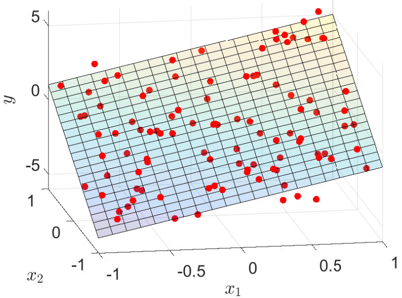
A $(D+1) \times (D+1)$

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):

$$\vec{w}^* = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{Y}^\top$$

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

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):

$$\vec{w}^* = (\mathrm{X}\mathrm{X}^\top)^{-1}\mathrm{X}\mathrm{Y}^\top$$

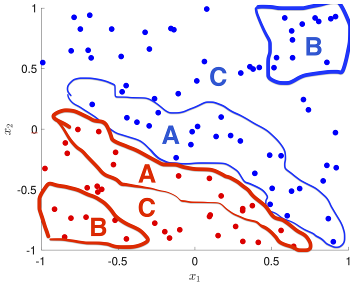
   - Metoda pro nalezení optimálních $\vec{w}^*$ analyticky!
   - Není třeba volit parametry optimalizačního algoritmu. Žádné iterace.
   - Je třeba spočítat $(\mathrm{X}\mathrm{X}^\top)^{-1}$; časová složitost $O((D+1)^3)$. Pro velké D nepraktické.

Obsah

Klasifikace

- 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 nejblíž 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 $\mathcal{T} = \{(\vec{x}^{(1)}, y^{(1)}), \ldots, (\vec{x}^{(N)}, y^{(N)})\}$:

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

Cíl:

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

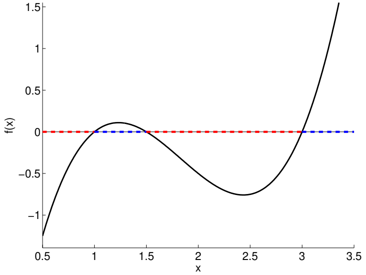
Diskriminační funkce

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):

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

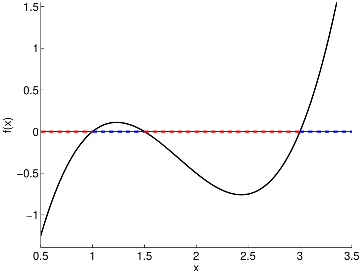
tj. $\hat{y} = \delta(\vec{x}) = \operatorname{sign}(f(\vec{x}))$.



- 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).

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):

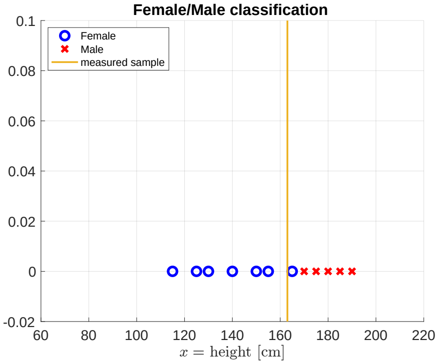
$$\hat{y} = \delta(\vec{x}) = \begin{cases} +1 & \text{iff} \quad f(\vec{x}) > 0, \text{ a} \\ -1 & \text{iff} \quad f(\vec{x}) < 0, \end{cases}$$

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

Příklad: Klasifikace Muž/Žena na základě výšky

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

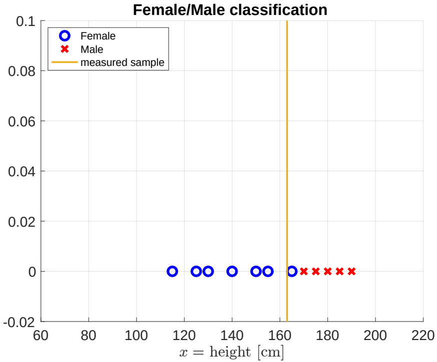
i	1	2	3	4	5	6	7	8	9	10	11	12
Výška $x^{(i)}$	115	125	130	140	150	155	165	170	175	180	185	190
Pohlaví $s^{(i)}$	F	F	F	F	F	F	F	M	M	M	M	M
Pohlaví $y^{(i)}$ (+1/-1)	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1



Příklad: Klasifikace Muž/Žena na základě výšky

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

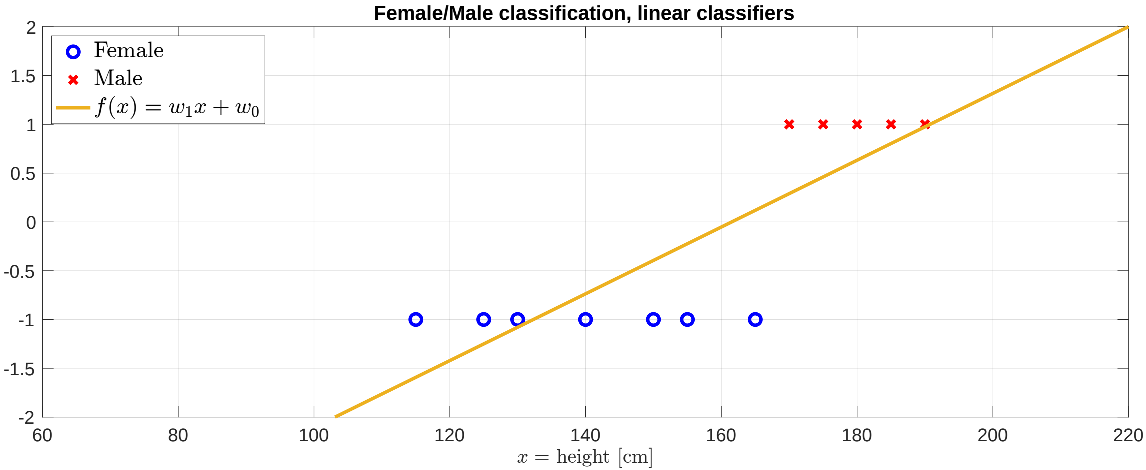
i	1	2	3	4	5	6	7	8	9	10	11	12
Výška $x^{(i)}$	115	125	130	140	150	155	165	170	175	180	185	190
Pohlaví $s^{(i)}$	F	F	F	F	F	F	F	M	M	M	M	M
Pohlaví $y^{(i)}$ (+1/-1)	-1	-1	-1	-1	-1	-1	-1	+1	+1	+1	+1	+1



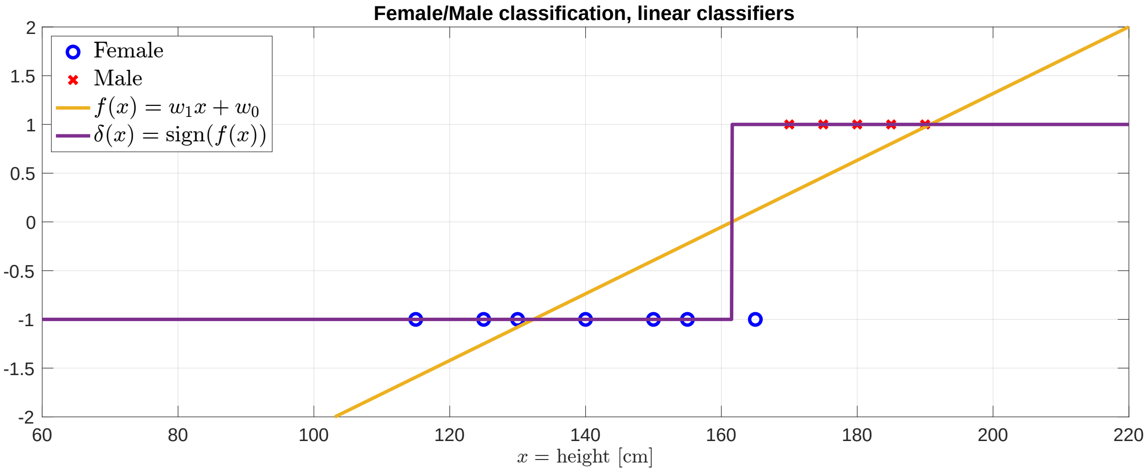
Nové pozorování ke klasifikaci: $x^Q = 163$

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

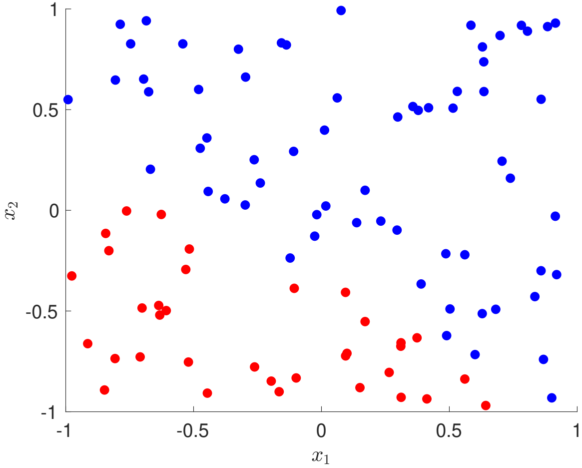
Příklad: Lineární diskr. funkce nalezená metodou nejmenších čtverců



Příklad: Odpovídající rozhodovací strategie

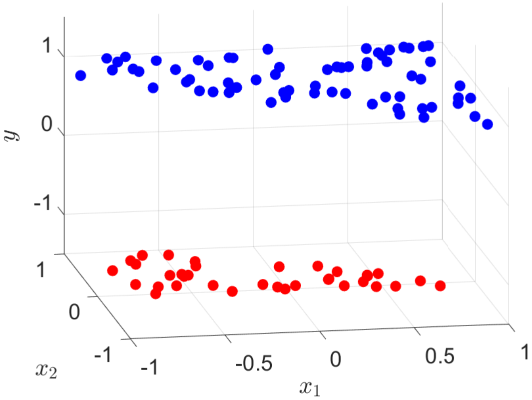


Učení lineárního klasifikátoru: naivní přístup



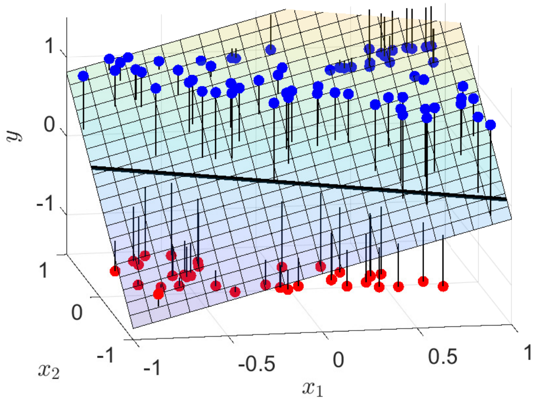
Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné třídy $s^{(i)}$.

Učení lineárního klasifikátoru: naivní přístup



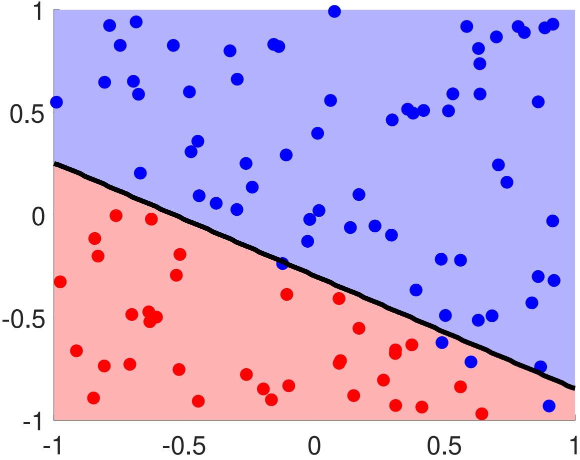
Vyjádřeme třídy hodnotami proměnné y, tj. $y^{(i)} = -1$ nebo $y^{(i)} = 1$.

Učení lineárního klasifikátoru: naivní přístup



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

Učení lineárního klasifikátoru: naivní přístup

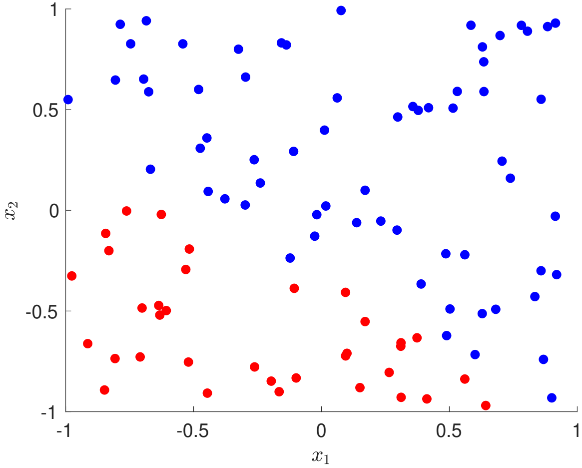


…pak tvoří lineární rozhodovací hranici v původním 2D prostoru.

Ale je takový klasifikátor obecně dobrý?

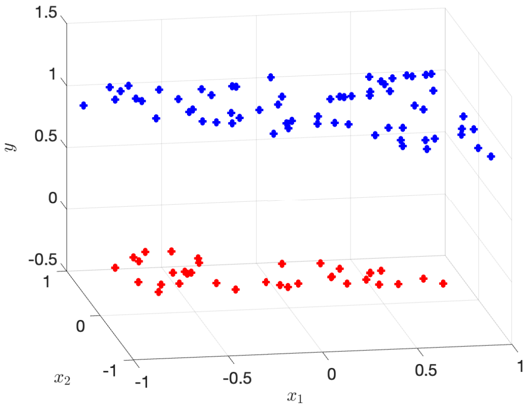
Lze udělat něco lepšího, než prokládat lineární funkci?

Prokládání vhodnější funkce: Logistická regrese



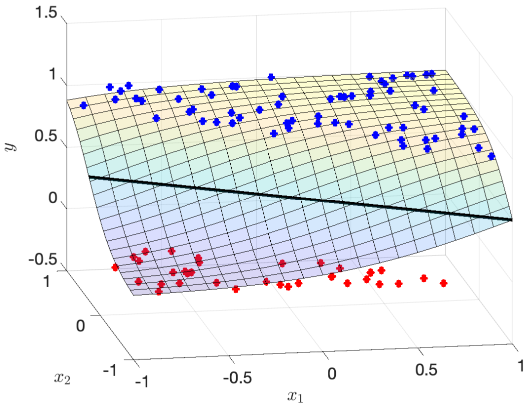
Mějme datovou sadu vstupních vektorů $\vec{x}^{(i)}$ a příslušné třídy $s^{(i)}$.

Prokládání vhodnější funkce: Logistická regrese



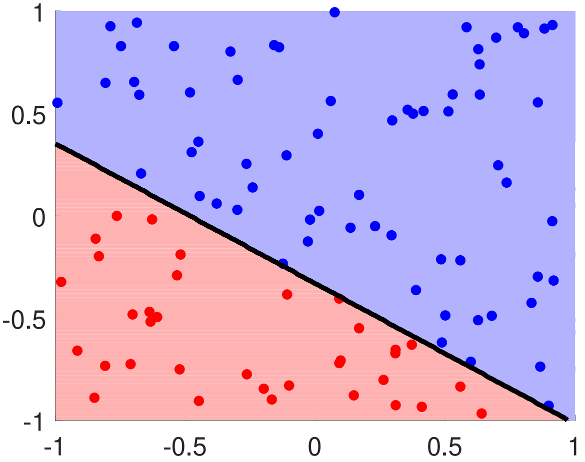
Vyjádřeme třídy hodnotami proměnné y, tj. $y^{(i)} = 0$ nebo $y^{(i)} = 1$.

Prokládání vhodnější funkce: Logistická regrese



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

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 $g(z) = \dfrac{1}{1 + e^{-z}}$ 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!

Model logistické regrese

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

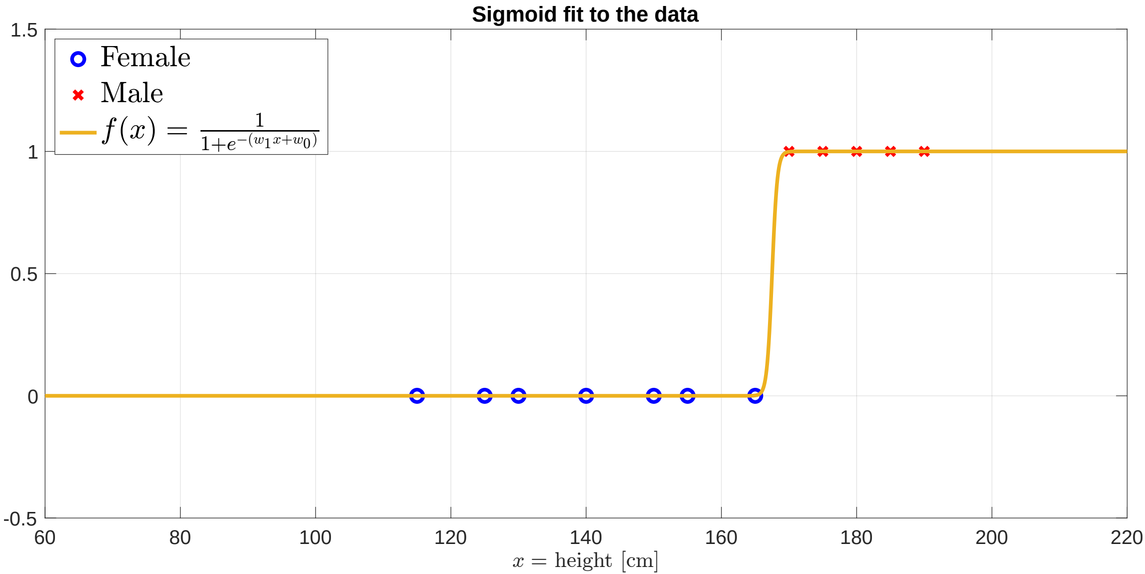
$$f_{\vec{w}}(\vec{x}) = g(\vec{w}^\top \vec{x}) = \frac{1}{1 + e^{-\vec{w}^\top \vec{x}}},$$

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

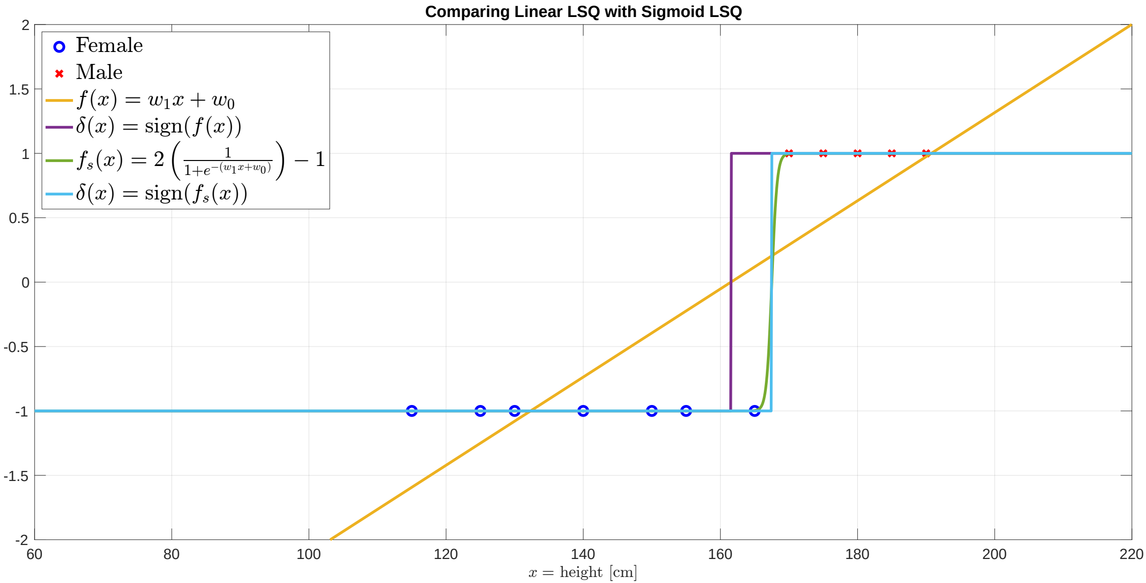
Interpretace modelu:

- $f_{\vec{w}}(\vec{x})$ 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!

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) :

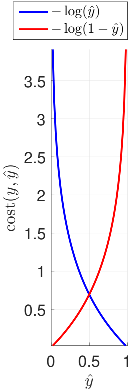
$$J(\vec{w}, \mathcal{T}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y^{(i)}, f_{\vec{w}}(\vec{x}^{(i)})), \text{ kde}$$

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

což se dá přepsat do jediného výrazu jako

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

- 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) :

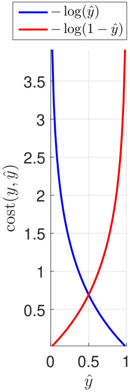
$$J(\vec{w}, \mathcal{T}) = \frac{1}{N} \sum_{i=1}^{N} \ell(y^{(i)}, f_{\vec{w}}(\vec{x}^{(i)})), \text{ kde}$$

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

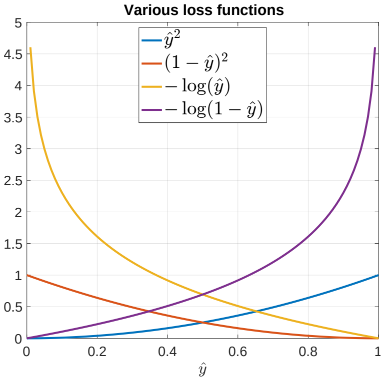
což se dá přepsat do jediného výrazu jako

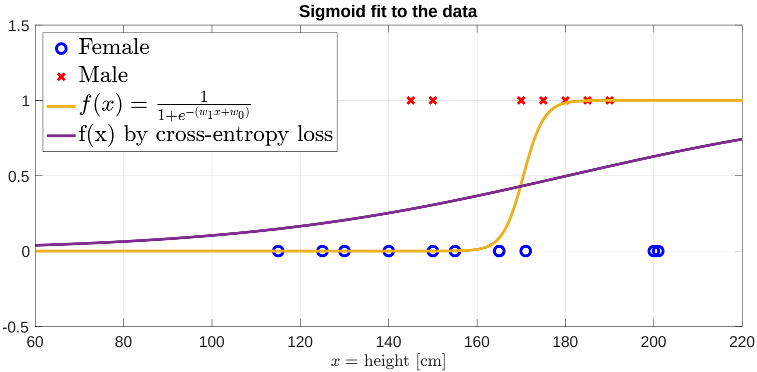
$$\ell(y, \hat{y}) = -y \cdot \log(\hat{y}) - (1-y) \cdot \log(1-\hat{y}).$$

- Snáze se optimalizuje pomocí numerických solverů.



Střední kv. chyba vs. křížová entropie

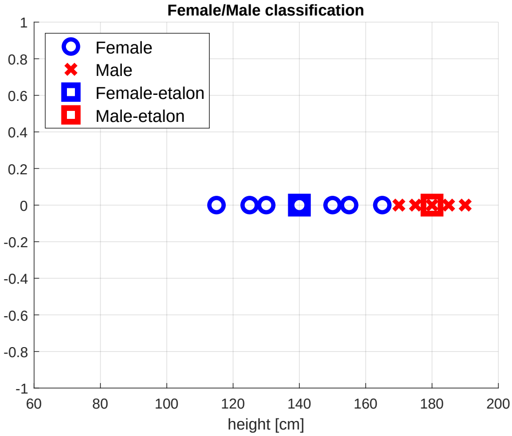




Sigmoidální $f(x)$ interpretujeme jako $P(s = \text{Male} \mid x)$: přímé učení diskriminativního modelu .
V porovnání s MSE křížová entropie silně penalizuje velké chyby!

Alternativní nápad: etalony

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



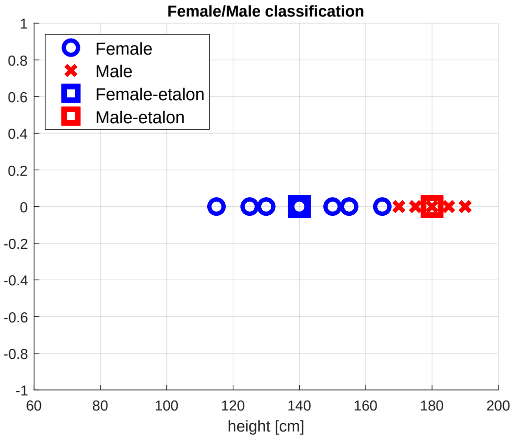
$$e_F = \text{ave}(\{x^{(i)} : s^{(i)} = F\}) = 140$$
$$e_M = \text{ave}(\{x^{(i)} : s^{(i)} = M\}) = 180$$

$x^Q = 163$

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

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 $d^Q = F$

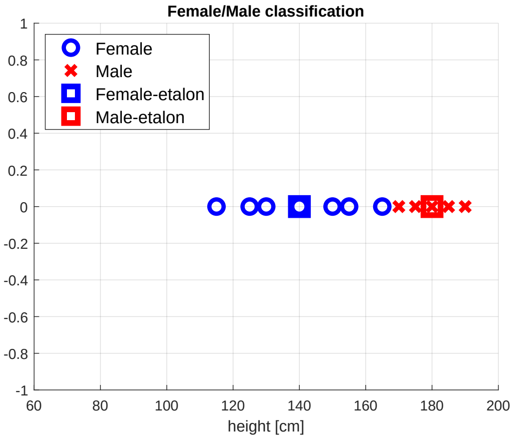
B $d^Q = M$

C Obě třídy stejně pravděpodobné

D Neumím rozhodnout

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

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

Etalonový klasifikátor je lineární klasifikátor!

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

$$\operatorname*{argmin}_{s \in S} \mathrm{dist}(x, e_s) = \operatorname*{argmin}_{s \in S} (x - e_s)^2 = \operatorname*{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) =$$

$$= \operatorname*{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname*{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2} e_s^2}_{\text{lineární v } x})$$

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

$$\operatorname*{argmin}_{s \in S} \operatorname{dist}(x, e_s) = \operatorname*{argmin}_{s \in S} (x - e_s)^2 = \operatorname*{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) =$$

$$= \operatorname*{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname*{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2} e_s^2}_{\text{lineární v } x})$$

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

$$\operatorname*{argmin}_{s \in S} \mathrm{dist}(x, e_s) = \operatorname*{argmin}_{s \in S} (x - e_s)^2 = \operatorname*{argmin}_{s \in S} (\underbrace{x^2}_{\text{konst.}} - 2e_s x + e_s^2) =$$

$$= \operatorname*{argmin}_{s \in S} (-2e_s x + e_s^2) = \operatorname*{argmax}_{s \in S} (\underbrace{e_s x - \frac{1}{2} e_s^2}_{\text{lineární v } x})$$

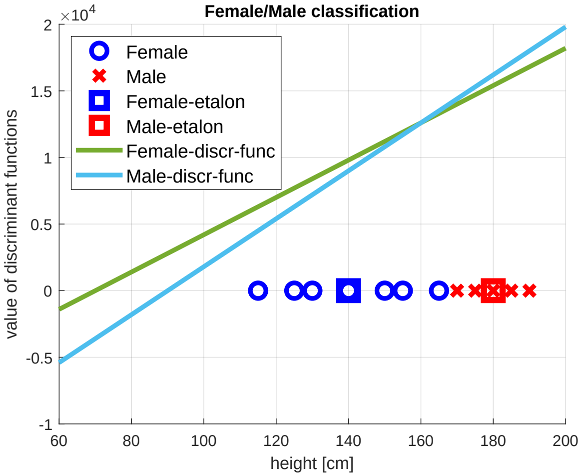
Klasifikace do více tříd: každá třída s má lineární diskriminační funkci $f_s(x) = a_s x + b_s$ a

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

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

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

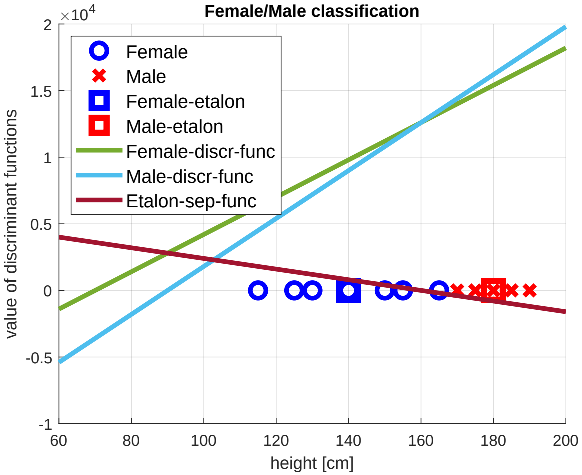
Příklad: F/M – Lineární diskriminační funkce založené na etalonech



Diskriminační funkce pro 2 třídy:

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

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



Diskriminační funkce pro 2 třídy:

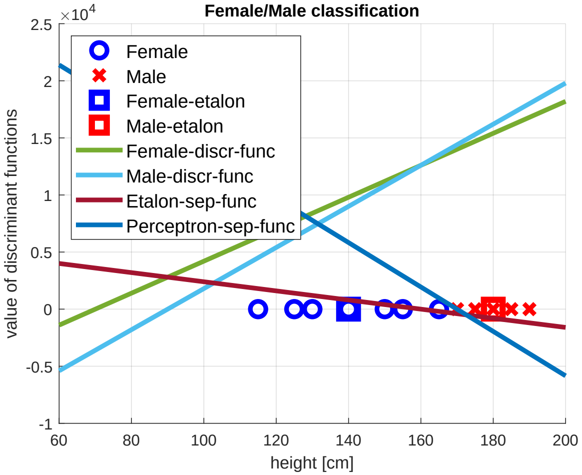
$$f_F(x) = a_F x + b_F = \\ = e_F x - \frac{1}{2}e_F^2 = 140x - 9800$$

$$f_M(x) = a_M x + b_M = \\ = e_M x - \frac{1}{2}e_M^2 = 180x - 16200$$

Jediná diskr. funkce oddělující obě třídy:

$$g(x) = f_F(x) - f_M(x) = \\ = -40x + 6400$$

Příklad: F/M – Umíme najít lepší etalony?

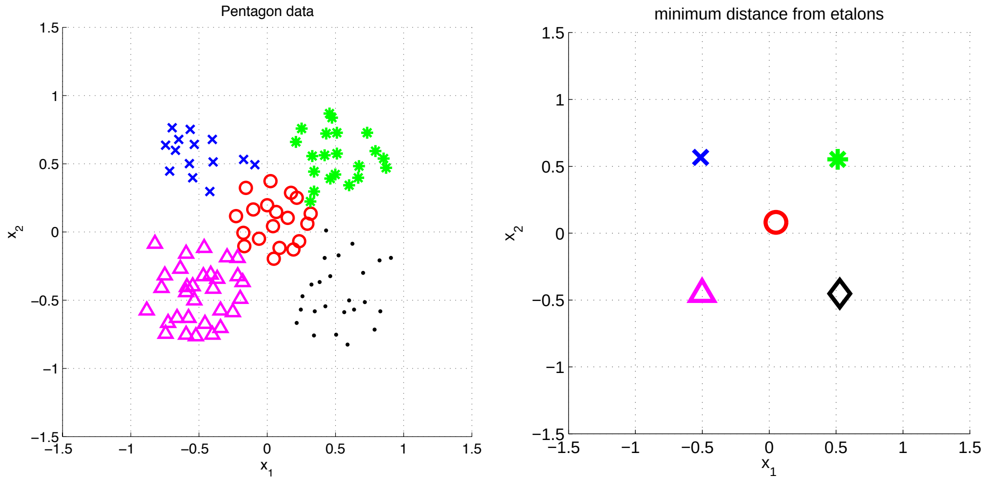


Lineární klasifikátory založené na průměrných etalonech dělají chyby.

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

Obsah

Etalony ve vícerozměrných prostorech



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.)

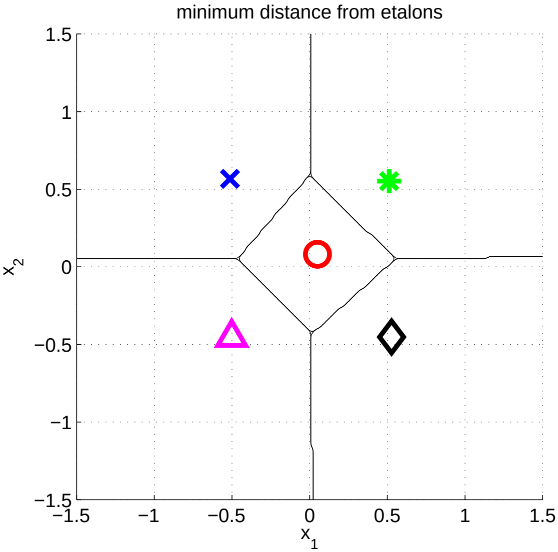
Extrahuj etalon pro každou třídu s:

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

Rozhodovací strategie

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

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



Etalony ve vícerozměrných prostorech (pokr.)

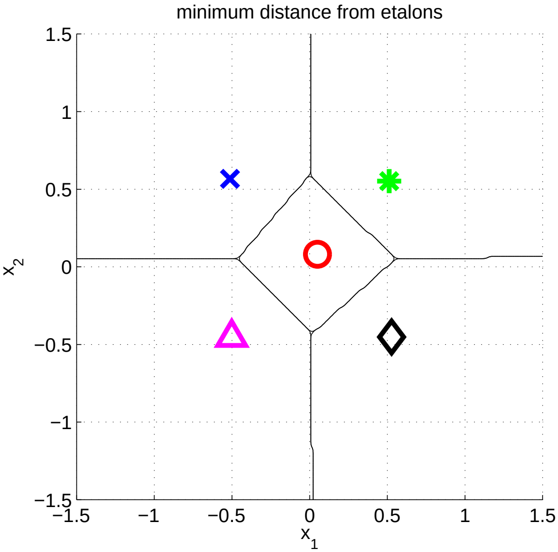
Extrahuj etalon pro každou třídu s:

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

Rozhodovací strategie

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

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



Etalony ve vícerozměrných prostorech (pokr.)

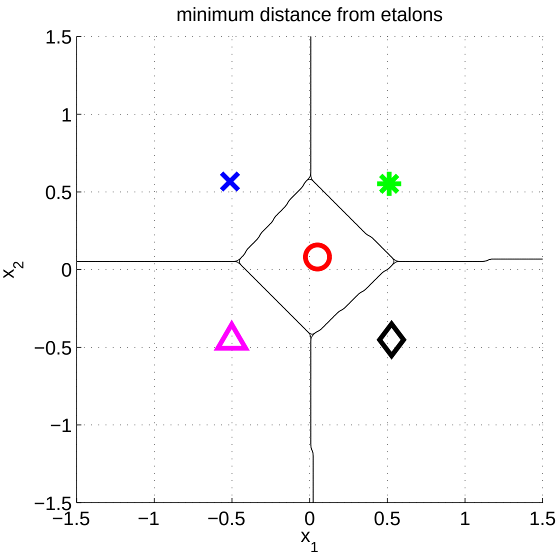
Extrahuj etalon pro každou třídu s:

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

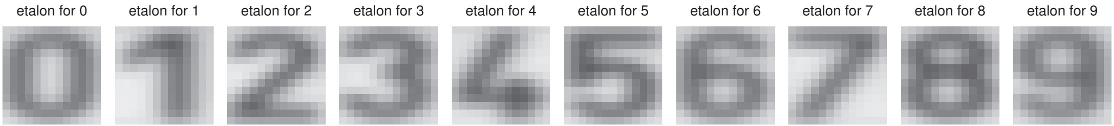
Rozhodovací strategie

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

Odpovídající rozhodovací hranice půlí vzdálenosti mezi dvojicemi etalonů.



Rozpoznávání číslic – etalony založené na průměru



Obrázky z [7].

Obsah

Bayesovská klasifikace vs diskriminační funkce

Rozhodnutí založené na diskriminační funkci:

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

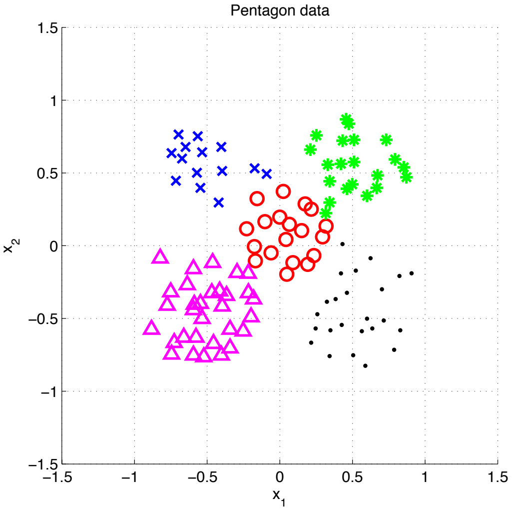
Rozhodnutí založené na posteriorní psti:

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

Pokud zvolíme

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

obě metody splývají.



Etalonový klasifikátor: generalizace do vyšších dimenzí

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

Lineární funkce (plus abs. člen)

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

Učení a rozhodování

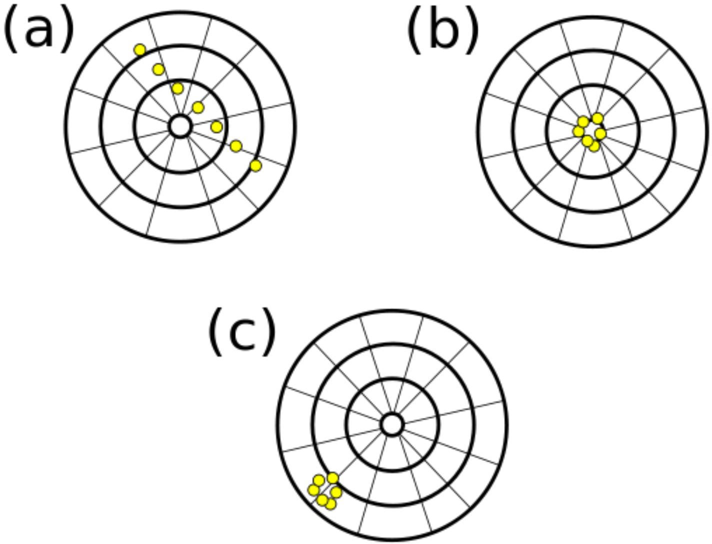
Fáze učení : určení modulu/funkce/parametrů na základě dat.
Fáze rozhodování : rozhodnutí o neznámém pozorování $\vec{x}$.

Co se učit?

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

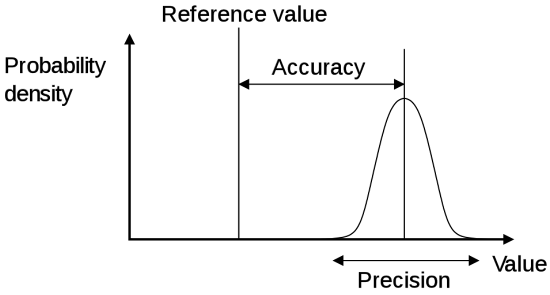
Obsah

Accuracy vs precision



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

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

Obsah

Literatura 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+Bussiness Media, New York, NY, 2006.
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.

Literatura II

[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/.