Neuronové sítě I. Perceptron. Perceptronový algoritmus a jeho konvergence

Václav Krauz, Tomáš Henych

15. května 2002

Obsah

1 Lineární diskriminační funkce (LDF) 2
1.1 Pojem LDF . . . 2
1.2 Význam LDF . . . 2
1.3 Rozdělující funkce . . . 5
1.4 LDF jako jednoduchý neuron . . . 6

2 Perceptron 7

3 Perceptronový algoritmus 8
3.1 Adaptační fáze (učení) . . . 8
3.2 Algoritmus učení perceptronu: . . . 9
3.3 Pracovní fáze . . . 9
3.4 Novikoffova věta . . . 9
3.5 Problém neseparabilních dat . . . 11

4 Shrnutí 12

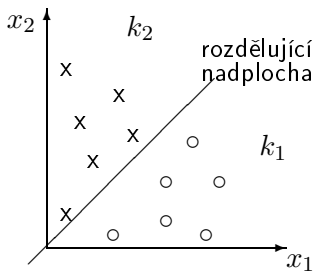
1 Lineární diskriminační funkce (LDF)

1.1 Pojem LDF

Uvažujme příznakový prostor $X = \{x_1, x_2, \ldots, x_n\}$ velikosti n (data naměřená na daném objektu), jenž je třeba klasifikovat do množiny tříd reprezentovaných vektorem $K = \{k_1, k_2, \ldots, k_m\}$ velikosti m. V doposud probírané Bayesovské strategii,[1] která je založena na minimalizaci rizika $\mathbf{R}(\mathrm{d})$, se setkáváme s problémem určení odhadu podmíněné pravděpodobnosti $P(X|k)$. Odhad $\hat{P}(X|k)$ je určován z trénovací množiny $T = ((x_1, k_1), (x_2, k_2), \ldots, (x_n, k_m))$, jejíž mohutnost je omezená. Mohou zde rovněž existovat stejné příznaky pro jevy různých tříd. Tento tedy mnohdy nepřesný odhad je považován za správný a chyba odhadu je v Bayesovském algoritmu zanedbána, což se může projevit chybným nastavením klasifikátoru. Na rozdíl od předchozího řešení lze data klasifikovat přímým nalezením tzv. diskriminační funkce $d(x)$ bez dalšího pochopení dat či jiných empirických znalostí.

Nejjednodušším případem je *diskriminační funkce lineární* (dále jen LDF), která rozděluje množinu vstupních dat do dvou tříd (viz obr. 1). Hodnota m je tedy 2. LDF s konkrétním prvkem příznakového prostoru tak určuje příslušnost k dané třídě:

$$d(x) = \vec{w}\ \vec{x} + w_0 \tag{1}$$



Obrázek 1: Příklad LDF na trénovací množině

1.2 Význam LDF

LDF zasluhují zvláštní pozornost z několika důvodů:

1. Pro některé statistické metody se optimální strataegie (Bayesovská i nebayesovská) realizuje pomocí LDF.

   ⇒ Normální rozdělení $P(X|\omega)$ se stejnou kovarianční maticí - pro 2D Gaussovo rozložení je ideálním odhadem křivka 2.řádu (typicky elipsa nebo parabola). Při malém počtu vzorků (příznakových

vektorů) je však rozdělující nadplocha určená pomocí LDF "přesnější" (klasifikuje s menší chybou) než nadplocha kvadratická.

Příklad 1: 2D Gaussovo rozložení

$$p(\vec{x}|k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\mathbb{C}|^{\frac{1}{2}}} e^{-\frac{1}{2}\left((\vec{x}-\vec{\mu}_k)^T \mathbb{C}^{-1}(\vec{x}-\vec{\mu}_k)\right)}$$

$p(\vec{x}\|k)$	...	podmíněná pravděpodobnost jevu (objekt je ve skrytém stavu k a pozorujeme příznak x)
$\mathbb{C}$	...	kovarianční matice, pro 1D Gaussovo rozložení je shodná s rozptylem σ^2
d	...	dimenze příznakového prostoru

Výraz upravíme logaritmováním:

$$\ln p(\vec{x}|k) = \Big(-\frac{1}{2}(\vec{x}-\vec{\mu}_k)^T \mathbb{C}^{-1}(\vec{x}-\vec{\mu}_k)\Big) + \ln \mathbb{C}_1 = \vec{w}\ \vec{x} + w_0$$

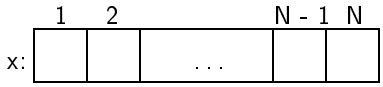
Čímž dostáváme již linearizovanou funkci.

⇒ Binární nezávislé příznaky

Příklad 2:

$$P(x_i|k) = P_{ki}^{x_i}(1-P_{ki})^{1-x_i} = \begin{cases} P_k^{x_i} & \text{pro } x_i = 1 \\ 1 - P_k^{x_i} & \text{pro } x_i = 0 \end{cases}$$

kde x_i jsou binární na sobě nezávislé příznaky $(i = 1, 2, \ldots, N)$



$\left.\begin{array}{l} P(x_i = 0|k) \\ P(x_i = 1|k) \end{array}\right\}$ potřebujeme znát pro všechna k

0	0.9	0.6	...	0.3
1	0.1	0.4	...	0.7

$p(x_i|k_i)$ – jednotlivé příznaky s určitou pravděpodobností

$$p(\vec{x}|k) = \prod_{i=1}^{N} P_{ki}^{x_i}(1-P_{ki})^{1-x_i}$$

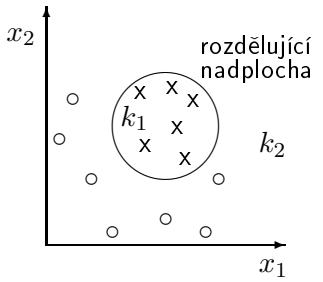
např: $\vec{x} = (0, 0, \ldots, 1)$

$P((0, 0, \ldots, 1)|k_1) = \prod p(x_i|k) = 0.9 * 0.6 * \ldots * 0.7$

2. Některé nelineární diskriminační funkce lze vyjádřit jako lineární po tzv. "vyrovnání příznakového prostoru"

$$\Phi_i : \mathbb{R}^d \to \mathbb{R}, \qquad g_k(x) = \sum_i w_{ki}\Phi_i(x) \tag{2}$$

Příklad 3:
Klasifikace do dvou tříd, jejichž rozdělující nadplocha je kvadratická (kružnice).
i-tý pozorovaný vektotr příznaků: $\vec{x}_i = (x_{i1}, x_{i2})$



Pro vnitřní část kruhu platí:

$$||(\vec{x} - \vec{c})||^2 < r^2$$

$$(x_1 - c_1)^2 + (x_2 - c_2)^2 < r^2$$

$$x_1^2 - 2x_1c_1 + c_1^2 + x_2^2 - 2x_2c_2 + c_2^2 - r^2 < 0$$

Čímž dostáváme již linearizovanou podobu:

$$y_1 + a_1y_2 + y_3 + a_2y_4 + y_5 < 0$$

"Vyrovnání příznakového prostoru" tedy provedeme transformací souřadnic vektorů $\Psi : \mathbb{R}^2 \to \mathbb{R}^5$. Pro jednotlivé vektory příznaků x_i potom:

$$(x_{i1},\ x_{i2}) \to (x_{i1}^2,\ x_{i1},\ x_{i2}^2,\ x_{i2},\ 1) \ \approx\ (y_{i1},\ y_{i2},\ y_{i3},\ y_{i4},\ y_{i5})$$

Poté již lze na takto linearizovaný příznakový prostor snadno aplikovat LDF.

3. Existuje univerzální způsob zobrazení příznakového prostoru do prostoru pravděpodobností, kde se řešení každé statistické úlohy realizuje lineární funkcí.

Příklad 4:
Případ binárních příznaků. Součin z příkladu č.2

$$p(\vec{x}|k) = \prod_{i=1}^{N} P_{ki}^{x_i}(1 - P_{ki})^{1-x_i}$$

upravíme logaritmováním na součet:

$$\sum_{i=1}^{N} \Big(x_i \ln P_{ki} + (1 - x_i) \ln(1 - P_{ki}) \Big)$$

$$\sum_{i=1}^{N} \Big(x_i (\ln P_{ki} - \ln(1 - P_{ki})) + \ln(1 - P_{ki}) \Big)$$

$$\sum_{i=1}^{N} (x_i w_{ki} + w_{k0}) \quad \ldots \text{rozepsaný skalární součin}$$

čímž jsme dostali lineární zobrazení: $\vec{x}\ \vec{w} + w_0$

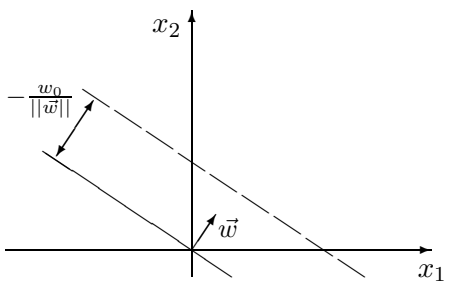
4. Kapacita (VC dimenze) lineárních strategií v n-rozměrném prostoru je $n + 2$ a učení je korektní.

5. Existují postupy pro nalezení linearních diskriminačních funkcí

   - ◇ Perceptron
   - ◇ Kozineův algoritmus
   - ◇ Support vector machine

1.3 Rozdělující funkce

Lineární rozdělující funkci hledáme ve tvaru: $d(x) = \vec{w}\ \vec{x} + w_0$, kde w je vektor vah (normálový vektor rozdělující nadroviny), x je vektor příznaků, w_0 konstanta posunutí rozdělující nadroviny (viz obr. 2).



Obrázek 2: Rozdělující nadrovina, normálový vektor a posunutí

Pro každou třídu z množiny $K = \{1, 2, 3, \ldots, M\}$ je diskriminační funkce dána vztahem: $g_k(\vec{x}) = \vec{w}_k \vec{x} + w_{k0}$

kde $\vec{x}$ je příznakový vektor
$\vec{w}$ je normálový vektor rozdělující nadroviny (viz obr. 2)
w_{k0} je posunutí rozdělující nadroviny

Hledaná rozdělující funkce $d(x)$ potom bude:

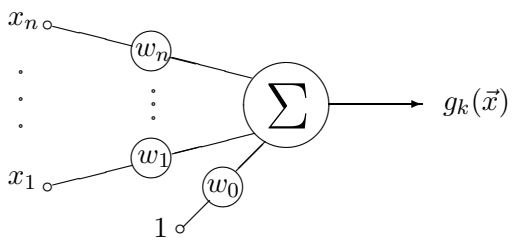
$$d(x) = \arg \max_{k \in K} g_k(\vec{x}) \tag{3}$$

nebo též jinou formou zapsáno $d(x) = k^* \iff g_k^*(\vec{x}) \geq g_k(\vec{x}), k \in K$

kde $g_k(\vec{x})$ je věrohodnost třídy k pro vektor $\vec{x}$. Výsledkem tedy bude pro příslušný příznakový vektor třída, která má maximální věrohodnost.

1.4 LDF jako jednoduchý neuron

Lineární klasifikátor může být představován jako jednoduchý neuron s váženými vstupy a jedním výstupem. Konstantu posunutí ω_0 lze obsáhnout na začátku nomálového vektoru rozdělující nadroviny, pokud příznakové vektory rovněž na začátku rozšíříme o 1 (viz obr.3).



Obrázek 3: Jednoduchý neuron

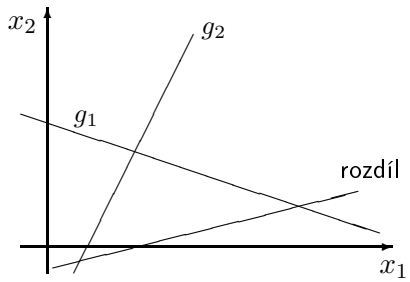
Pro dvě třídy jsou diskriminační funkce dány vztahy:

$$g_1(\vec{x}) = \vec{w}_1\vec{x} + w_{10}$$
$$g_2(\vec{x}) = \vec{w}_2\vec{x} + w_{20}$$

Pro porovnání rychlosti růstu obou funkcí pro dané x poslouží nalezení jejich rozdílu:

$$g_1 - g_2 = (\vec{w}_1 - \vec{w}_2)\vec{x} + w_{10} - w_{20}$$

Ke změně rozhodování pak dochazí, když rozdíl obou diskriminačních funkcí prochazí nulovou hodnotou (viz obr.4).



Obrázek 4: Diskriminační funkce pro dvě třídy jako funkce rozdílu

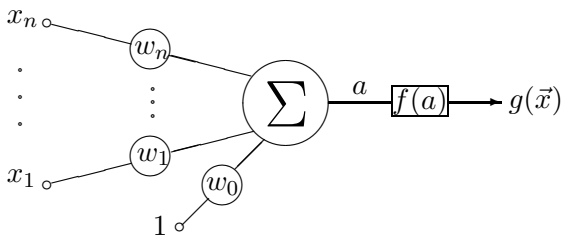
2 Perceptron

Jedním ze základních algoritmů přímého hledání diskriminační funkce je Perceptronový algoritmus, Perceptron. Jak bude později ukázáno, i zde se setkáváme s problémy např. v podobě nedostatečného počtu vzorků (příznakových vektorů) či velkého množství parametrů, jež jsou výpočetně prakticky nezpracovatelné. Nicméně tyto problémy mají svá řešení v určitých modifikacích základního algoritmu.
Diskriminační funkce může být zobecněna nahrazením lineární funkce

$$g_k(x) = \vec{w}_k\vec{x} + w_{k0} \tag{4}$$

monotónní funkcí aplikovanou na lineární kombinaci $\vec{w}$ a $\vec{x}$

$$g_k(x) = f(\vec{w}_k\vec{x} + w_{k0}). \tag{5}$$



Obrázek 5: Perceptron (Model McCulloch-Pittsova neuronu)

Funkce f se nazývá aktivační funkce.

Perceptron je tedy jednovrstvá neuronová síť (viz obr.5) s prahovou aktivační funkcí, která se většinou volí jako lichá ve tvaru:

$$f(a) = \begin{cases} 1 & \text{pro } a \geq 0 \\ -1 & \text{pro } a < 0 \end{cases}$$

což odpovídá funkci *sign*. Pro shrnutí tedy:

Diskriminační funkce (DF):	$g(\vec{x}) = \vec{w}\vec{x} = \sum_i w_i x_i$
Rozhodnutí:	$d(\vec{x}) = sign(g(\vec{x}))$

3 Perceptronový algoritmus

3.1 Adaptační fáze (učení)

Uvažujme 2 třídy, označme je 1,−1 t.j. $K = \Omega = \{-1, 1\}$. Dále známe modifikovanou trénovací multi-množinu $T = ((\vec{x}_1, k_1), (\vec{x}_2, k_2), \ldots, (\vec{x}_n, k_n))$, jež je vstupem perceptronového algoritmu. Přičemž $\vec{x}_i$ jsou vektory příznaků na konci rozšířené o hodnotu 1 a $\vec{w}$ je vektor vah na identické souřadnici rozšířený o posun rozdělující nadroviny w_0. Formálně zapsáno:

$$\vec{x} \in \mathbb{R}^n \rightarrow \mathbb{R}^{n+1}$$
$$\vec{x} = (x'_1, x'_2, \ldots, x'_n, 1)$$

$$\vec{w} \in \mathbb{R}^n \rightarrow \mathbb{R}^{n+1}$$
$$\vec{w} = (w'_1, w'_2, \ldots, w'_n, w'_0).$$

Tato jednoduchá úprava nám umožňuje realizovat výpočet:

$$\vec{w}'\vec{x}' + w'_0$$

pouhým skalárním součinem $\vec{w}\vec{x}$.

Pak na počátku (v čase $t = 0$) uvažujeme $\vec{w}^0 = (0, 0, 0, \ldots, 0)$. V čase $t+1$ klasifikujeme vstupní vektor podle $k_t(\vec{w}^t\vec{x}_t)$. Pokud je výsledek klasifikace kladné číslo (>0), byla klasifikace správná a nedojde k žádné změně vektoru vah:

$$\vec{w}^{t+1} = \vec{w}^t.$$

V případě, že je však výsledkem nekladná hodnota (≤ 0), nebyla klasifikace vektoru $\vec{x}^t$ správná a vektor vah musíme upravit podle vztahu:

$$\vec{w}^{t+1} = \vec{w}^t + k_t x_t.$$

Poté však musí být celá trénovací multi-množina opětovně zkontrolována, neboť by mohlo dojít vlivem posunutí rozdělující nadroviny k chybné klasifikaci jiného prvku.

3.2 Algoritmus učení perceptronu:

Vstupem perceptronového algoritmu je modifikovaná trénovací multi-množina (viz výše). Dále je využit vektor synaptických vah rozšířený o práh neuronu. Postup je následující:

1. Inicializuj vektor vah

$$\vec{w}^0 = (0, 0, \ldots, 0)^T, \tag{6}$$

   přičemž horní index 0 označuje časový okamžik iterace.

2. Nalezni v trénovací množině špatně klasifikovaný vektor (označme ho x_t a jeho třídu k_t), pro který platí:

$$k_t(\vec{w}^t \vec{x}_t) \leq 0. \tag{7}$$

3. Pokud nebyl nalezen žádný špatně klasifikovaný vektor, jsou váhy adaptovány a algoritmus se ukončí. Byl-li takový vektor nalezen, uprav vektor vah podle pravidla:

$$\vec{w}^{t+1} = \vec{w}^t + k_t x_t. \tag{8}$$

4. Přejdi na krok 2.

Jestliže vstupní množinou jsou lineárně separabilní data, algoritmus je ukončen v konečném počtu kroků (důkaz viz Novikoffova věta), v opačném případě se algoritmus nezastaví. Z nalezeného vektoru w pak zpětnou tranformací opět získáme vektor vah a práh neuronu, který odpovídá posunutí rozdělující nadroviny.

3.3 Pracovní fáze

Za předpokladu, že máme naučenou neuronovou síť, resp. perceptron, tj. známe vektor vah $\vec{w}$ a práh w_0, klasifikujeme vstupní vektor $\vec{x_t}$ dosazením do vztahu:

$$k_t(\vec{w}\vec{x}_t + w_0). \tag{9}$$

Pokud je vnitřní část závorek (tzv. potenciál neuronu) větší nebo roven nule, je vektor $\vec{x}$ klasifikován do třídy $k = 1$ a v opačném případě je klasifikován do třídy $k = -1$.

3.4 Novikoffova věta

Je-li T (trénovací multi-množina) lineárně separabilní, tj. existuje

$$w' : k_i(\vec{w}'\vec{x}_i) > 0, \text{ pro } i = 1, \ldots, n,$$

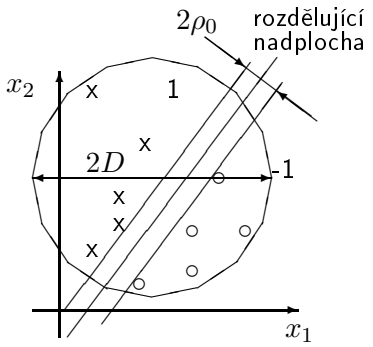
pak perceptronové učení konverguje, tzn. po čase t_s (pro $t > t_s$) $\vec{w}^t = \vec{w}^{t_s}$

$$s < \frac{D^2}{\rho_0^2} \tag{10}$$

kde D je poloměr nejmenší opsané koule datům $\vec{x}_i$, tj. $|\vec{x}_i| < D$

ρ_0 je minimální vzdálenost prvku od separační nadplochy $\rho_0 \leq \min_i k_i \frac{\vec{w}_0 \vec{x}_i}{|\vec{w}_0|}$

w_0 určuje optimální rozdělující nadplochu



Obrázek 6: Separabilní data v konvexním obalu

1. $\vec{\omega}^0 = \vec{o}$
2. Hledáme horní odhad velikosti $|\vec{w}^t|$.
   Velikost $\vec{w}^t$ roste pouze při korekci; maximálně o D^2

$$|\vec{w}^t|^2 = |\vec{w}^{t-1}|^2 + \underbrace{2k_t(\vec{x}_t \vec{w}^{t-1})}_{\leq 0} + |x_t|^2$$

$$|x_t|^2 \leq D^2 \text{ (z definice } D)$$

$$\Rightarrow |\vec{w}^t|^2 \leq |\vec{w}^{t-1}|^2 + D^2$$

do času t došlo maximálně k t korekcím a tedy

$$\boxed{|\vec{w}^t| \leq tD^2} \tag{11}$$

neboť $|\vec{w}^0| = 0$

3. $\exists$ vektor $\vec{w}^0$: $\dfrac{k_i(\vec{w}^0\vec{x}_i)}{|\vec{w}^0|} \geq \rho_0$ (předpoklad věty)

$$\frac{\vec{w}^t\vec{w}^0}{|\vec{w}^0|} = \frac{\vec{w}^{t-1}\vec{w}^0}{|\vec{w}^0|} + \frac{k_t(\vec{x}_t\vec{w}^0)}{|\vec{w}^0|} > \frac{\vec{w}^{t-1}\vec{w}^0}{|\vec{w}^0|} + \rho_0 \geq t\rho_0$$

Swarz - Cauchyova nerovnost

$$|\vec{w}^t||\vec{w}^0| \geq \vec{w}^0\vec{w}^t$$

$$\boxed{|\vec{w}^t| \geq \frac{\vec{w}^t\vec{w}^0}{|\vec{w}^0|} > t\rho_0} \tag{12}$$

$$t^2\rho_0^2 \leq |\vec{w}^t| \leq tD^2$$

$$\boxed{t \leq \frac{D^2}{\rho_0^2}} \tag{13}$$

3.5 Problém neseparabilních dat

V případě nesparabilních dat je složitá analýza a problém minimalizace chyby (počet chybně zařazených vzorů) je NP úplná úloha. Zde je tedy třeba se spokojit s pravděpodobnostním algoritmem.
Řešíme problém: Jak ukončit učení perceptronu, aby s pravděpodobností $1-\eta$ byla chyba maximálně ε?

$\Rightarrow$ Zavedeme zastavovací pravidlo typu "ukonči učení po k krocích, jestliže $m(k)$ vzorků za sebou je klasifikováno správně!"
Pravděpodobnost, že učení bude zastaveno, i když pravděpodobnost chyby $P^{\varepsilon}(\vec{w}_k) > \varepsilon$ po $m(k)$ úspěšných klasifikacích, je

$$P_k = \Big(1 - P(w_k)\Big)^{m(k)} < (1-\varepsilon)^{m(k)}. \tag{14}$$

K této chybě může dojít v kterémkoli kroku. Celková pravděpodobnost zastavení i při chybě větší než ε je

$$P < \sum_{k=1}^{\infty} P_k < \sum_{k=1}^{\infty}(1-\varepsilon)^{m(k)} \leftarrow \text{můžeme volit} \tag{15}$$

Počet správně klasifikovaných vzorků tedy musí být

$$m(k) = \Big[\frac{2\ln K - \ln m + \ln\frac{\pi^2}{6}}{-\ln(1-\varepsilon)}\Big] + 1. \tag{16}$$

4 Shrnutí

Perceptron

Diskriminační funkce (DF):	$g(\vec{x}) = \vec{w}\vec{x} = \sum_i w_i x_i$
Rozhodnutí:	$d(\vec{x}) = sign(g(\vec{x}))$
Učení:	Perceptronový algoritmus (PA)
Výsledek:	Rozdělující nadrovina